Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.057294

ARTICLE

GL-YOLOv5: An Improved Lightweight Non-Dimensional Attention Algorithm Based on YOLOv5

Yuefan Liu, Ducheng Zhang and Chen Guo*

School of Software, Dalian Jiaotong University, Dalian, 116000, China

*Corresponding Author: Chen Guo. Email: gc0094@126.com

Received: 14 August 2024 Accepted: 27 September 2024 Published: 18 November 2024

ABSTRACT

Craniocerebral injuries represent the primary cause of fatalities among riders involved in two-wheeler accidents; nevertheless, the prevalence of helmet usage among these riders remains alarmingly low. Consequently, the accurate identification of riders who are wearing safety helmets is of paramount importance. Current detection algorithms exhibit several limitations, including inadequate accuracy, substantial model size, and suboptimal performance in complex environments with small targets. To address these challenges, we propose a novel lightweight detection algorithm, termed GL-YOLOv5, which is an enhancement of the You Only Look Once version 5 (YOLOv5) framework. This model incorporates a Global DualPooling NoReduction Blend Attention (GDPB) module, which optimizes the MobileNetV3 architecture by reducing the number of channels by half and implementing a parallelized channel and spatial attention mechanism without dimensionality reduction. Additionally, it replaces the conventional convolutional layer with a channel shuffle approach to overcome the constraints associated with the Squeeze-and-Excitation (SE) attention module, thereby significantly improving both the efficiency and accuracy of feature extraction and decreasing computational complexity. Furthermore, we have optimized the Variable Normalization and Attention Channel Spatial Partitioning (VNACSP) within the C3 module of YOLOv5, which enhances sensitivity to small targets through the application of a lightweight channel attention mechanism, substituting it for the standard convolution in the necking network. The Parameter-Free Spatial Adaptive Feature Fusion (PSAFF) module is designed to adaptively modify the weights of each spatial position through spatial pooling and activation functions, thereby effectively enhancing the model's ability to perceive contextual information over distances. Ultimately, GL-YOLOv5 performs remarkably in the custom dataset, achieving a model parameter count of 922,895 M, a computational load of 2.9 GFLOPS, and a mean average precision (mAP) of 92.1%. These advancements significantly improve the model's detection capabilities and underscore its potential for practical applications.

KEYWORDS

Lightweight; traffic safety helmet detection; YOLOv5; GDPB; PSAFF

1 Introduction

In the domain of computer vision, target detection represents a fundamental task with extensive potential applications in areas such as security surveillance [1], autonomous driving [2], and smart

Copyright © 2024 The Authors. Published by Tech Science Press.
This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

home technology [3]. However, traditional target detection algorithms often fail to meet the real-time requirements of embedded devices and mobile platforms due to high model complexity and large computation. In response to this challenge, researchers have increasingly focused on lightweight techniques to enhance target detection efficiency by minimizing the number of model parameters and reducing computational complexity. Lightweight methodologies, including model compression, pruning, and quantization, address the limitations of current target detection algorithms regarding model size and computational demands. These advancements enhance target detection on embedded systems and mobile devices while fostering new opportunities in computer vision and driving industry innovation.

In this context, safety helmet detection in transportation becomes a specific application scenario. In recent years, e-bikes have become the preferred choice for public transportation due to their convenient, low-cost, and environmentally friendly features. However, the traffic safety hazards they bring cannot be ignored. As indicated in the most recent report by the World Health Organization [4], approximately 1.3 million individuals lose their lives annually due to road traffic accidents, with another 20 to 50 million suffering non-fatal injuries, over fifty percent of which involve vulnerable road users. Helmet wearing can significantly reduce the risk of traffic accidents [5], but the wearing rate is generally low, especially in developing countries.

Consequently, traffic enforcement agencies must verify riders' compliance with helmet regulations. However, the conventional manual inspection method is inefficient and typically relies on random sampling to ensure adherence to helmet usage. This methodology is labor-intensive and costly, with a limited scope of coverage, and it also increases the likelihood of individuals taking risks or attempting to evade detection when observed from a distance. In light of advancements in deep learning, there is a pressing need to investigate more efficient and effective supervisory techniques. Currently, deep learning-based target detection models can be categorized into two main types: the first category comprises larger models that offer higher accuracy but are hindered by slower detection speeds and greater hardware requirements, which limits their widespread applicability; the second category includes lightweight models that provide rapid detection and lower hardware demands but suffer from reduced accuracy, leading to potential omissions and misdetections. To address these challenges, this study proposes a lightweight safety helmet detection algorithm called GL-YOLOv5, which integrates the You Only Look Once (YOLO) framework with a lightweight model. The objective of this approach is to minimize the model's size and its reliance on hardware performance while simultaneously enhancing accuracy. This innovation aims to rectify the limitations of existing lightweight models, facilitating more efficient and precise detection, which could serve as a viable alternative to human visual assessment.

GL-YOLOv5 has been specifically developed to enhance the performance of helmet detection tasks. To address the challenges of small targets and complex backgrounds, we introduce advanced modules: Global DualPooling NoReduction Blend Attention (GDPB), Variable Normalization and Attention Channel Spatial Partitioning (VNACSP), and Parameter-Free Spatial Adaptive Feature Fusion (PSAFF) to address the limitations of traditional models when dealing with small targets and complex scenarios. By facilitating more efficient feature extraction and channel mixing processes, these modules significantly improve the model's accuracy in detecting small targets while simultaneously reducing computational costs, aligning more closely with the practical requirements of helmet detection. Through innovative improvements to existing models, our GL-YOLOv5 model dramatically increases the accuracy and efficiency of helmet detection tasks. Notably, the GL-YOLOv5 consists of only 922,895 parameters, representing an 87% reduction compared to the original YOLOv5s. Additionally, its computational demand has been minimized to 2.9 GFLOPS. In terms of detection

performance, the GL-YOLOv5 achieves a mean Average Precision at 50% IoU (mAP50) of 92.1%, surpassing several current lightweight, high-precision state-of-the-art (SOTA) models, including 89.9% for YOLOv8n, 91.4% for YOLOv9t, and 91.2% for YOLOv10n. The following section on comparative experiments will present detailed results comparing these models. These findings indicate that the GL-YOLOv5 significantly reduces model size and hardware requirements and exhibits enhanced detection accuracy and stability in practical application contexts.

The main contributions of this paper include the following:

- Proposing DualPooling NoReduction Attention (DPA): This module can effectively capture the channel feature information with lower computational cost and without dimensionality reduction.
- Proposed Global DualPooling NoReduction Blend Attention (GDPB): This module combines DPA, focuses on global feature information, and enhances the inter-channel feature information exchange by reducing the number of channels and lowering the computational effort while using channel blending instead of convolution. The GDPB module addresses the limitations of the Squeeze-and-Excitation (SE) block in MobileNetV3 and serves as a foundational component for the GL-YOLOv5 algorithm's feature extraction network.
- An analysis was conducted on the defects of the C3 module within the YOLOv5 detection framework: a lightweight channel attention module has been incorporated into the cross-stage partial network, leading to the development of the Variable Normalization and Attention Channel Spatial Partitioning (VNACSP) module.
- Propose Parameter-Free Spatial Adaptive Feature Fusion (PSAFF) module: This module enhances the model's capacity to discern critical information adaptively, improving accuracy without introducing supplementary parameters.
- Creation of the Ride Together Helmet Recognition Dataset (RTHR): Due to the lack of a public dataset of two-wheelers and helmets, we created the RTHR dataset. The dataset consists of two categories: riders wearing safety helmets on e-bikes and riders not wearing safety helmets on e-bikes.

The remainder of this paper is organized in the following manner: Section 2 presents the related literature; Section 3 describes the model selection and improvement; Section 4 describes the experimental preparation and setup and analyzes the experimental results; and Section 5 summarizes the main conclusions.

2 Related Work

Target detection used to be dominated by two-stage algorithms such as Regions with Convolutional Neural Networks (R-CNN) [6], Fast R-CNN [7], and Faster R-CNN [8]. These algorithms first generate multiple target candidate frames on the image through preprocessing methods and then utilize convolutional neural networks to classify the samples to obtain high training and detection accuracy. However, the computational cost of these algorithms is high, which impacts the training and detection efficiency of the network.

Subsequently, one-stage algorithms such as YOLO [9–11] and Single Shot MultiBox Detector (SSD) [12] gradually came to the fore. These methods transform the target border localization problem into a regression problem, realizing end-to-end detection without needing candidate frames. Compared to the two-stage algorithms, the one-stage algorithms significantly improve training and detection efficiency, although they may be less accurate. This series of algorithms greatly facilitates the

speed of information collection, processing capability, and delivery, providing technical and theoretical support for intelligent detection related to road traffic safety.

In recent years, the increasing demand for embedded devices and mobile applications has highlighted the limitations of one-stage models, particularly their size, which often fails to satisfy the anticipated requirements. Consequently, there has been a growing emphasis on the development of lightweight models, such as MobileNet [13–15], PP-LCNet [16], EspNetv2 [17], and ShuffleNet [18,19], which have become central to current research efforts. Literature has been presented in [20] to improve the design of a garbage classification system based on ShuffleNetv2, with a single inference time of about 105 ms on a Raspberry Pi 4B, and it takes only 0.88 s to classify and collect a single object. Li et al. [21] designed a lightweight detector deployed offline on non-cloud devices based on PP-LCNet. Nguyen et al. implemented a faster and simpler traffic sign detection method using the EspNetv2 algorithm. Bi et al. implemented a low-cost and effective apple disease detection model with the MobileNet model as the core.

Despite their advantages, lightweight models exhibit several limitations in practical applications. For instance, the Squeeze-and-Excitation (SE) module in MobileNet experiences a loss in dimensionality reduction within the channel attention mechanism, which hampers its ability to capture feature information fully. Additionally, PP-LCNet demonstrates inadequate feature extraction capabilities when processing complex features and small targets. While EspNetv2 excels in lightweight design, it faces computational complexity and feature processing constraints when confronted with large-scale data and intricate scenes. ShuffleNet primarily utilizes channel shuffling for feature fusion; however, this approach may lead to insufficient feature utilization in the context of varying scales and complex scenarios, thereby restricting computational efficiency. These common deficiencies across the models suggest that current lightweight networks possess inherent limitations in addressing complex features, multi-scale targets, and extensive datasets, necessitating further optimization and enhancement to achieve improved target detection performance.

This paper introduces an innovative architectural design integrating a one-stage detection algorithm with a lightweight model to mitigate these challenges. This approach aims to minimize model size and reduce reliance on hardware performance while enhancing model accuracy and efficiency, thereby addressing the shortcomings of existing one-stage and lightweight models.

3 GL-YOLOv5 Model and Its Improvements

3.1 Model Selection and Model Deficiencies

YOLO is an end-to-end, single-stage object detection model developed and optimized over many years through several iterations. Among these, YOLOv5 stands out due to its minimal number of parameters, rapid detection speed, and strong generalization capabilities, making it particularly well-suited for security helmet detection on embedded devices. Consequently, YOLOv5 has been selected as the benchmark model for this paper. Nevertheless, it still has some problems. For example, the high number of convolution and pooling layers makes applying computationally intensive algorithms to lightweight edge detection devices difficult. Meanwhile, its feature capture and fusion capabilities are relatively insufficient, limiting the further improvement of its detection accuracy.

Consequently, we primarily enhance the backbone and neck components of the YOLOv5 architecture to develop a lightweight detection model with superior performance. The structure of GL-YOLOv5 is illustrated in Fig. 1. A more detailed account of our enhancements will be provided later in this section.

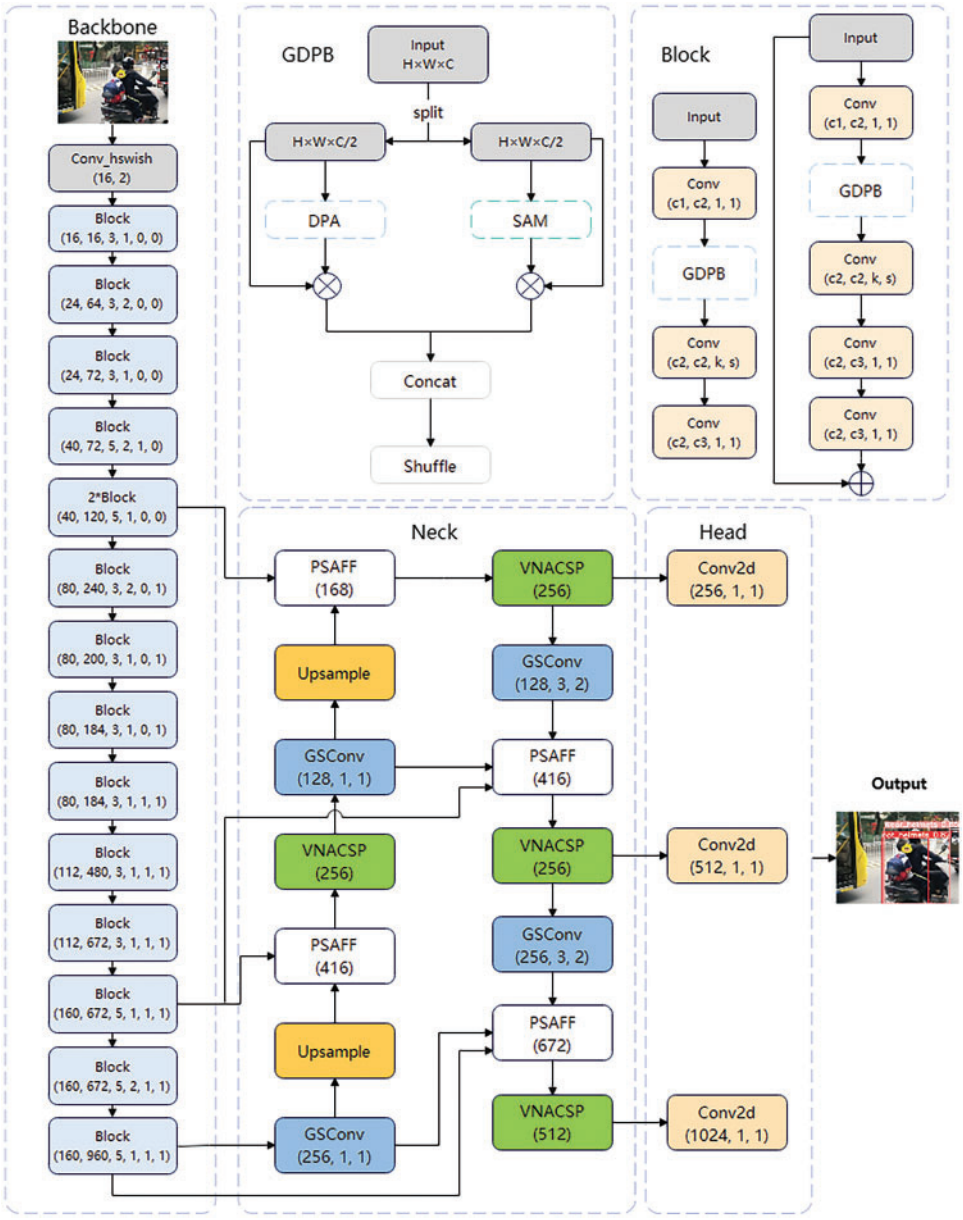

Figure 1: GL-YOLOv5 Network Architecture

3.2 Improving the MobileNetV3 Network with Global DualPooling NoReduction Blend Attention (GDPB)

The MobileNetV3 architecture [15] incorporates the depthwise separable convolution from the V1 model [13] and the inverse residual structure featuring a linear bottleneck from the V2 model [14]. The parameters for MobileNetV3 are derived through Network Architecture Search (NAS), resulting in a model that demonstrates superior performance and efficient computational speed.

The Squeeze-and-Excitation (SE) attention mechanism [22] introduced in MobileNetV3 has some drawbacks regarding lightweight. We decided not to use dimensionality-reducing operations in the

channel attention module to minimize the computational effort and avoid the loss of dimensionality reduction. Inspired by the Convolutional Block Attention Module (CBAM) attention module [23], which uses both maximum and average pooling, we propose the DualPooling NoReduction Attention (DPA) mechanism.

For a given input feature map $F_1 \in R^{H\times W\times C/2}$, we first compute maximum pooling and average pooling independently to form parallel branches for feature extraction. With maximum pooling and average pooling, we can maintain comprehensive information within the image while minimizing the influence of spatial details, thereby emphasizing the most salient information.

Next, we use a one-dimensional convolutional kernel (of size k, where k is the range of local interactions between channels) to achieve channel information aggregation and counteract the negative impact of dimensionality reduction on channel attention. Subsequently, the feature maps $M_C^{avg}(F_1)$ and $M_C^{max}(F_1)$ are obtained by activation functions, respectively, and then the elements of the two branches are summed. We use the activation function to improve information fusion and feature representation and avoid the output of the element summation being relatively large or small when summed.

With this design, we can fuse the information from both branches in a more balanced way and finally get the channel's attention $M_C(F_1)$. The DPA attention map $M_C(F_1) \in R^{H\times W\times C/2}$, can be calculated as follows:

$$M_C^{avg}(F_1) = \sigma\left(f_1\left(AvgPool\left(F_1\right)\right)\right), \tag{1}$$

$$M_C^{max}(F_1) = \sigma\left(f_1(MaxPool\left(F_1\right)\right)), \tag{2}$$

$$M_C(F_1) = \sigma(\sigma\left(f_1\left(AvgPool\left(F_1\right)\right)\right) + \sigma\left(f_1(MaxPool\left(F_1\right)\right))) = \sigma\left(M_C^{avg}(F_1) + M_C^{max}(F_1)\right), \tag{3}$$

where σ denotes the sigmoid function and f_1 denotes the one-dimensional convolution operation.

This attention mechanism aims to optimize feature representation, improve model performance, and overcome some limitations of the SE attention mechanism. Its structure is schematically shown in Fig. 2. The channel attention module employs both average and maximum pooling techniques. This approach preserves the comprehensive information of the image while reducing the impact of spatial details, thus allowing a concentrated emphasis on the most salient features.

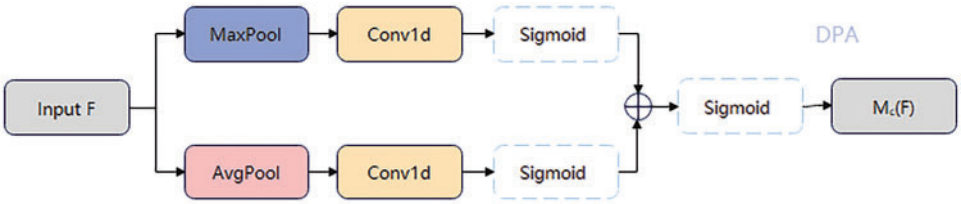

Figure 2: Structure of the DualPooling NoReduction Attention (DPA) module

Despite the significant progress achieved in reducing computational costs by replacing the Squeeze-and-Excitation (SE) attention mechanism in MobileNetV3 with DPA, the DPA framework is currently limited to the analysis of channel-specific feature information, with less emphasis placed on the examination of spatial feature information. It is also important to consider the role of spatial attention mechanisms, which adaptively filter essential spatial regions. The combination of channel and spatial attention mechanisms can result in a more comprehensive and reliable set of attention information, thereby facilitating a more reasonable allocation of computational resources [23]. Notably, both the Squeeze-and-Excitation (SE) and DPA attention modules primarily

focus on channel-wise attention, which constrains their capacity to capture attention in the spatial dimension, thereby limiting the learning potential of the network.

Therefore, effectively integrating channel attention mechanisms and spatial attention mechanisms becomes crucial. We propose a Global DualPooling NoReduction Blend (GDPB) Attention module. This module prioritizes both channel and spatial attention mechanisms. In contrast to conventional blended attention approaches, it initially decreases computational expenses by reducing the number of input channels by fifty percent. Following this reduction, parallel channel and spatial attention strategies are utilized to improve computational efficiency and overall performance. Furthermore, channel shuffling is implemented to mitigate the computational load associated with convolutional layers and to facilitate improved information exchange among channels. The schematic diagram of this module is illustrated in Fig. 3.

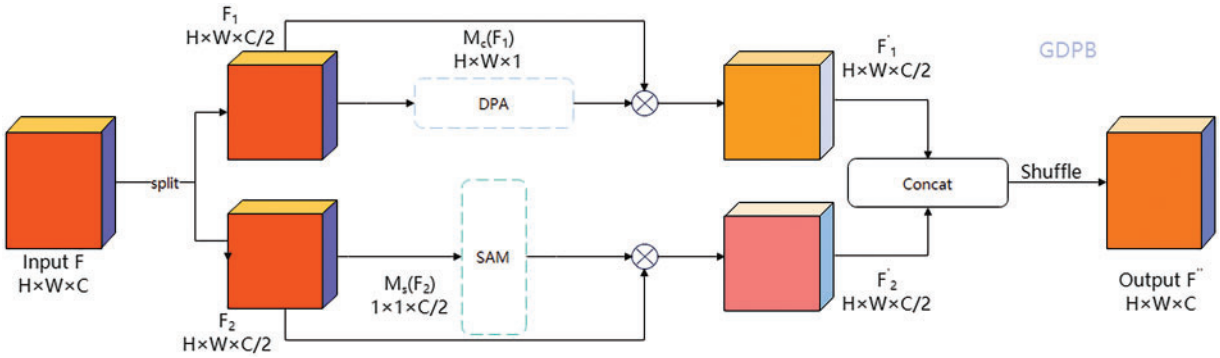

Figure 3: Structure of the Global DualPooling NoReduction Blend (GDPB) attention module

For a given input feature map $F \in R^{H\times W\times C}$, we first decompose it into two feature maps of the same size, F_1 & F_2, where $F_1, F_2 \in R^{H\times W\times C/2}$, and then compute F_1 & F_2 by independent parallel channel and spatial attention. We obtain $M_C(F_1) \in R^{H\times W}$ and $M_S(F_2) \in R^C$, respectively, from $R^{H\times W\times C}$, and then multiply the F_1 and F_2 feature maps element-by-element with $M_C(F_1)$ and $M_S(F_2)$, respectively, to obtain the feature maps $F_1' \in R^{H\times W\times C/2}$, $F_2' \in R^{H\times W\times C/2}$. Next, we stitch the two feature maps F_1' & F_2', together to obtain $F' \in R^{H\times W\times C}$, and finally a 3D feature map $F'' \in R^{H\times W\times C}$ is obtained by channel blending. The overall process of GDPB attention can be summarized as follows:

$$F'' = S([M_C(F_1) \otimes F_1, M_S(F_2) \otimes F_2]) = S\left(\left[F_1', F_2'\right]\right) = S(F'), \tag{4}$$

where $S(\varphi)$ represents channel shuffling, [,] denotes concatenation, f represents convolutional operation, and $\otimes$ represents element-wise multiplication.

As illustrated in Fig. 4, the Spatial Attention Module (SAM) conveys the same concepts of maximum and average pooling. Since spatial attention encompasses various orientations, we employ a stitching technique to integrate these orientations. Subsequently, spatial attention $M_S(F_2)$ is derived through the application of 2D convolution followed by an activation function. The computation is expressed as follows:

$$M_S(F_2) = \sigma(f_2([AvgPool(F_2), MaxPool(F_2)])), \tag{5}$$

where σ represents the sigmoid function, [,] denotes concatenation, and f_2 represents two-dimensional convolution.

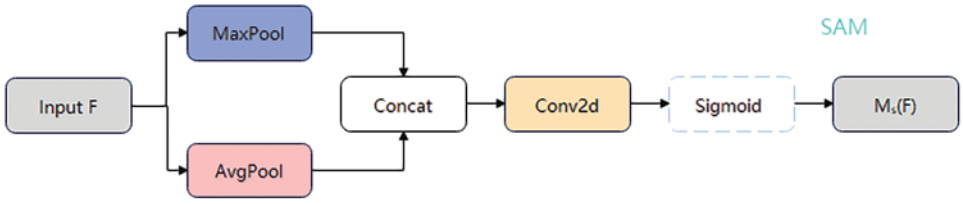

Figure 4: Structure of the spatial attention module (SAM)

We improve MobileNetV3 by introducing the proposed GDPB module for a lighter and superior backbone feature extraction network. The module emphasizes integrating global and local features, enhancing the underlying backbone network's feature extraction and learning capabilities.

3.3 Cross-Stage Partial Networks with Hybrid Convolution and Channel Attention

Li et al. [24] proposed a depth-separable convolution-based GSConv module based on the channel shuffle operation of ShuffleNet [18], the structure of which is schematically depicted in Fig. 5. The feature learning branch of the GSConv module employs a two-stage strategy of dimensionality reduction and enhancement, facilitating the exchange of information between channels through techniques such as concatenation and mashup.

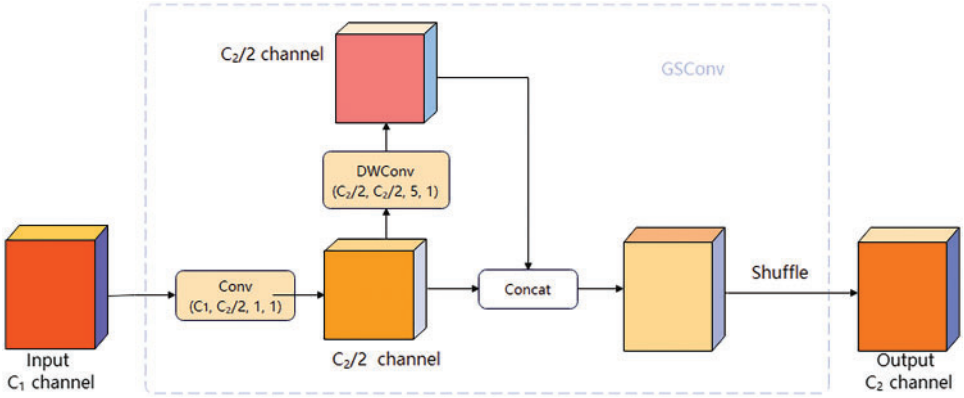

Figure 5: Structure of the GSConv module

In terms of the front neck network, compared with the backbone network, the output feature figure of the front neck network in GL-YOLOv5 has a smaller size and more channels, and the transformations and loops of the features are smoother. This approach better preserves the semantic information inherent to the features in question. Therefore, GL-YOLOv5 arranges the GSConv module in the front neck network to minimize the computational burden. The input feature tensor provided to the GSConv module is initially transformed to half the output channels through a standard convolution operation, utilizing a stride and kernel size of 1. Subsequently, the resultant feature maps are processed using a depth-wise separable convolution (DWConv), which employs a kernel size of 5 and a stride of 1. Finally, the output feature maps are connected to the standard convolution output, and the final output feature maps are generated through a channel blending operation.

The C3 module is a key learning part in the neck network, but due to the large amount of computation and insensitivity to small targets, Xu et al. [25] showed that introducing an attention

mechanism can improve the capacity of the network to learn from small goals. Meanwhile, Li et al. [24] showed that deploying the GSConv module makes the network respond better to attention. Therefore, to minimize computational demands and improve sensitivity to minor targets, we developed the VNACSP module, which integrates a channel attention mechanism into the VoV-GSCSP1 framework. The VNACSP schematic is shown in Fig. 6.

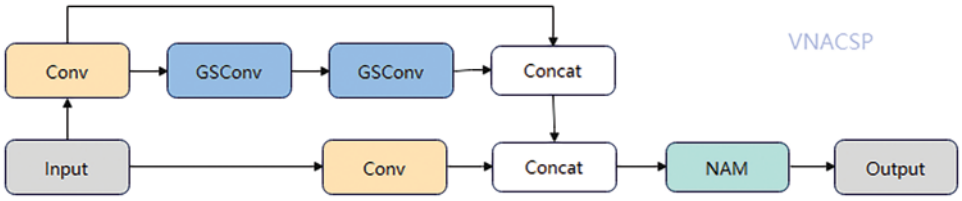

Figure 6: Structure of the VNACSP module. "NAM" in the figure represents the channel attention in the normalization-based attention module

The configuration of the lightweight, efficient channel attention module is illustrated in Fig. 7 and Eq. (6). For this channel attention module, we use the scaling factor in Batch Normalization (BN), which is calculated as shown in Eq. (7). The size and importance of each channel's variance are reflected in the scaling factor. The larger the variance, the richer the information contained in the channel and the higher the importance, while the channel with smaller variance has relatively single information and lower importance.

$$M(F) = \sigma(W_{\lambda}(BN(F))), \tag{6}$$

$$B_{out} = BN(B_{in}) = \alpha \frac{B_{in} - \mu}{\sqrt{\gamma^2 + \epsilon}} + \beta, \tag{7}$$

where $M(F)$ represents the output feature. λ is the scaling factor for each channel, and the weights are obtained through Eq. (8). σ denotes the sigmoid function, μ and γ are the mean and standard deviation of the mini-batch, where α and β are trainable parameters of the affine transformation. (scale and shift).

$$W_{\lambda} = \frac{\lambda_i}{\sum_{j=0} \lambda_j}, \tag{8}$$

Finally, the output feature map $M(F)$ is element-wise multiplied with the input tensor $F \in R^{H \times W \times C}$ to obtain the output tensor $F' \in R^{H \times W \times C}$. The computation can be expressed as follows:

$$F' = M(F) \otimes F. \tag{9}$$

where $\otimes$ denotes the element-wise multiplication.

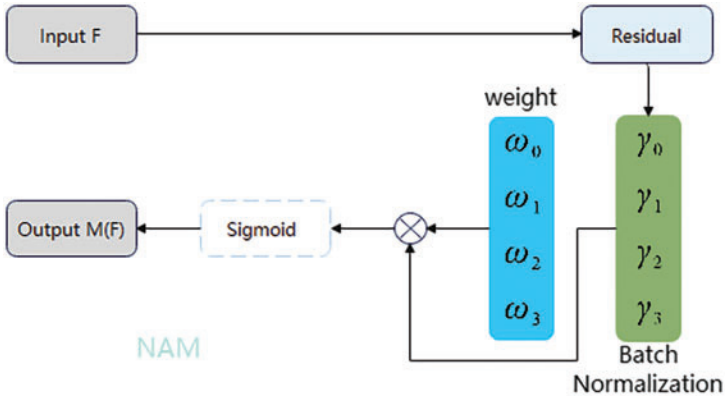

Figure 7: Structure of the NAM module

3.4 Parameter-Free Spatial Adaptive Feature Fusion (PSAFF)

A common issue with traditional feature fusion methods in target detection is their failure to consider the varying importance of different features. Current fusion strategies often treat all features equally, neglecting that some features may be more critical to the task. This leads to models that assign uniform weights to all features, ultimately hindering their ability to capture essential information about key targets effectively. In contrast, during human visual tasks, the brain focuses selectively on significant regions within a scene to process information more efficiently. Building on this biological mechanism, this paper proposes a method called Parameter-Free Spatial Adaptive Feature Fusion (PSAFF) and introduces a Parameter-Free Spatial Adaptive Module (PFSA). The model simulates the important distribution of different regions in the feature map through pooling operations and a Sigmoid activation function, thereby enabling dynamic weight allocation in the spatial dimension. Its structure is schematically illustrated in Fig. 8.

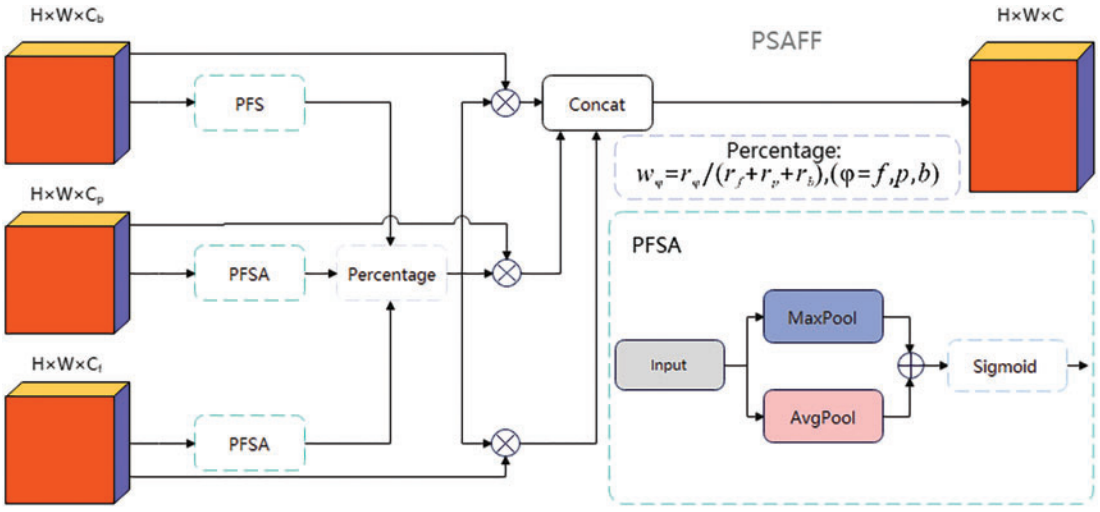

Figure 8: Structure of the PSAFF module

Firstly, having replaced the backbone network, the same level of features contains more available features with channel size. Therefore, in the feature splicing part, we will implement an adaptive

weighting mechanism for the feature information across the three components of the backbone network, specifically within the FPN layer and the PANet segment of the feature integration process. Then, in the feature splicing process, we use spatial attention instead of channel attention to calculate the weights. The weight calculation formula is as follows:

$$r_{\varphi} = \sigma\left(AvgPool\left(F_{\varphi}\right) + MaxPool\left(F_{\varphi}\right)\right), (\varphi = f, p, b), \tag{10}$$

$$w_{\varphi} = \frac{r_{\varphi}}{r_f + r_p + r_b}, (\varphi = f, p, b), \tag{11}$$

where σ represens the sigmoid function, $MaxPool\left(F_{\varphi}\right)$ and $AvgPool\left(F_{\varphi}\right)$ respectively denote spatial dimension-wise max pooling and average pooling, r_{φ} represents the Parameter-Free Spatial Adaptive (PFSA) module.

This allows our model to focus more flexibly on the importance of features at any image location without being constrained by channel correlation. This lightweight design not only enhances the computational efficiency of the model but also increases its adaptability in contexts where spatial information is critical to the task, resulting in significantly enhanced perception of important information. However, we introduce the PSAFF module because we want to solve this problem from a lightweight point of view without adding additional parameters. Its calculation method is as follows:

$$F = [w_f \otimes F_f, w_p \otimes F_p, w_b \otimes F_b], \tag{12}$$

where F_f and w_f are the feature map and associated weights from the FPN layer, F_p and w_p are the feature map and corresponding weights from the PANet part, F_b and w_b are the feature map and associated weights from the same detection layer of the backbone network, $\otimes$ denotes element-wise multiplcation, and [,] represents concatenation.

Compared to traditional fusion methods, this method can adaptively assign different weights to features at different locations based on the amount of information in each feature figure. Such adaptivity can improve the model's ability to recognize important information and focus more on regions containing key target information, thus improving target detection performance.

3.5 K-Means Anchor Clustering

In target detection training, the model needs to acquire knowledge regarding the target objects' location, dimensions, and classification. To facilitate the learning process, we opt to utilize a limited number of anchor frames that exhibit a higher probability of occurrence as reference points, which markedly enhances the stability of the training procedure. The anchor frames employed in the YOLOv5 model are sourced from the COCO dataset; however, it is imperative to customize the initial anchor frames according to the specific dataset utilized due to the considerable variability in target categories and aspect ratios across different datasets.

We used the K-means clustering algorithm [26–28], and by analyzing the shapes and aspect ratios of the two types of labels in the RTHR homemade dataset, we obtained the initial anchor frames as [41, 80], [66,126], [86,178], [141,158], [115, 232], [153, 278], [222, 245], [216, 369], and [330, 335].

The K-means clustering method was chosen because of its ability to effectively categorize the target frames in the dataset into several classes and generate initial anchor frames suitable for the data distribution. Compared with other methods, K-means clustering has the following advantages:

- Adaptive: K-means can adaptively adjust the anchor frames according to the dataset's characteristics so that the anchor frames fit the target distribution of the dataset more closely.

- High computational efficiency: The K-means algorithm is simple and efficient, making it suitable for anchor frame clustering of large-scale data sets.
- Significant effect: The anchor frames generated by clustering can significantly improve the model's training stability and detection accuracy and reduce the loss of inference frames.

In contrast, other methods, such as manually setting anchor frames or using a priori knowledge to generate anchor frames, may be unable to adequately adapt to the features of specific datasets, leading to unsatisfactory training results. We can effectively reduce the loss of inference frames and improve target detection accuracy by training the model based on the anchor frames that K-means clustering yields.

4 Experiment

The experimental setup utilizes the Ubuntu operating system, with the model algorithm implemented through the PyTorch deep learning framework. The hardware configuration includes an NVIDIA GeForce RTX 3080 graphics card with 10 GB of graphics memory and an Intel(R) Xeon(R) Platinum 8255C CPU supported by 40 GB of RAM. The model is designed to accept an initial input image size of 640 × 640 pixels. The training process is conducted for 300 epochs, with a batch size of 16. The initial learning rate is established at 0.03, while the cyclic learning rate is set to 0.12, and the momentum for the learning rate is maintained at its default value of 0.937. Additionally, the weight decay parameter, warm-up period, and warm-up momentum are configured to their default values of 5×10^{-4}, 3, and 0.8, respectively. The Stochastic Gradient Descent (SGD) algorithm is employed as the optimization function, and training is executed utilizing cosine annealing, with data augmentation enabled.

4.1 Experimental Dataset

Since there is no publicly available dataset of two-wheeler helmets, we collected helmet images from various traffic scenarios, including congestion, weather and lighting conditions, and shooting angles, through web crawlers and real-life photography. After de-duplication and manual screening, 3890 images were finally retained. We manually labeled these images using LabelImg software to generate a data file in YOLO format containing target category and target frame coordinate information.

In general, the labeling method for similar datasets is to label the car, the human body, and the head separately to detect whether the helmet is being worn on the head or not by distance, etc. However, there are defects in this separate labeling: in complex environments, especially in crowded areas or traffic congestion, it may not be able to determine the wearing of helmets accurately, and it is easy to misidentify one person's helmet as another person's, leading to recognition errors. In contrast, labeling the person, car, and head can consider the correlation between them more comprehensively, improve the accuracy of discriminating the helmet-wearing situation, better match real-world scenarios, and make models more robust.

Therefore, we adopted the method of labeling the vehicle, person, and helmet as a whole and set two labeling categories: "Wear_helmets" (riders wearing safety helmets on e-bikes) and "not_helmets" (riders not wearing safety helmets on e-bikes). We created a dataset called the Ride Together Helmet Recognition Dataset (RTHR), containing 3890 images labeled with more than 5500 labels; given that this dataset is independently created and relatively small in scale, the authors opted to partition it into a training set consisting of 3112 images and a validation set containing 778 images, rather than adhering to the conventional tripartite division of training, validation, and testing sets. This decision

was made considering the model's robustness and the efficacy of the training outcomes. By adopting this approach, the model is afforded increased training opportunities despite the limited dataset, while the validation set serves to assess its generalization capabilities. In the event that additional data becomes accessible in the future, the introduction of a distinct test set may be considered for more comprehensive evaluation. An example of partial dataset labeling for the sample is shown in Fig. 9.

Figure 9: Example of annotated images in the partial dataset

4.2 Evaluation Metrics

Algorithms are evaluated based on two primary criteria: computational cost and accuracy. The number of parameters (M) and giga floating point operations per second (GFLOPS) are the primary measures of computational cost. A lower number of parameters and GFLOPS indicates that the model requires fewer computational resources and less advanced hardware capabilities. Accuracy is primarily characterized by several metrics, including precision, recall, average precision (AP), and mean average precision (mAP). The calculation methods for each of these evaluation metrics are as follows:

$$Precision = \frac{tp}{tp + fp}, \tag{13}$$

$$Recall = \frac{tp}{tp + fn}, \tag{14}$$

$$AP = \int_0^1 P(R)\, dR, \tag{15}$$

$$mAP = \frac{\sum_{i=1}^{K} AP_i}{K}, \tag{16}$$

where tp denotes the quantity of true positive samples that have been accurately identified, fp signifies the number of samples that have been erroneously classified as positive despite being negative, fn indicates the number of true positive samples that have been mistakenly classified as negative, and K represents the aggregate number of identified target categories.

4.3 Ablation Experiments

To evaluate the effectiveness of our optimization efforts for each model component, we have conducted a series of stepwise ablation experiments designed to improve performance, using YOLOv5s as the baseline model in all subsequent experiments. The experimental data are shown in Table 1.

Table 1: Step-by-step ablation experiments

Models	Backbone	Modules		Algorithms	Params (M)	GFLOPS	Precision	Recall	Model size (MB)	$map_{0.5}$
	MN3	VNACSP	PSAFF	K-means						
Baseline					7,015,519	15.8	84.9%	83.6%	14.5	89.6%
Case 1	✓				1,345,687	3.8	86.5%	82.8%	3.2	90.5%
Case 2	✓	✓			899,855	2.9	84.9%	84.8%	2.3	90.8%
Case 3	✓		✓		1,376,407	3.9	86.5%	85.1%	3.2	91.1%
Case 4	✓	✓	✓		922,895	2.9	87.8%	82.4%	2.4	91.4%
Case 5	✓	✓	✓	✓	922,895	2.9	89.1%	85.1%	2.4	92.1%

Note: ✓ indicates the use of a module or algorithm.

We established the Baseline model as the standard configuration of YOLOv5s. Its unaltered state shows Precision, Recall, and mAP50 values of 84.9%, 83.6%, and 89.6%, respectively. These metrics indicate potential improvements in reducing false detections, missed detections, and overall accuracy, particularly for small targets and complex backgrounds.

In Case 1, we integrated an enhanced MobileNetV3 (MN3) into YOLOv5s, significantly reducing parameters from 7,015,519 to 1,345,687 and computational demands from 15.8 GFLOPS to 3.8 GFLOPS. Precision and mAP50 improved to 86.5% and 90.5%, respectively, while recall slightly declined to 82.8%. This suggests that MN3 effectively preserves feature extraction and reduces costs but may struggle with small targets in complex backgrounds, ultimately enhancing detection performance while conserving resources.

In Case 2, we integrated the VNACSP module into the MN3 framework to enhance small target detection. This integration reduced parameters and computational requirements to 899,855 and 2.9 GFLOPS, respectively. Although precision slightly decreased, recall improved to 84.8%, resulting in a 0.3% increase in mAP50. The VNACSP module demonstrates greater sensitivity to small targets, maintaining high recall while lowering computational complexity and effectively reducing missed detections.

In Case 3, we introduced the PSAFF module, based on MN3, to enhance the model's ability to perceive contextual information over long distances in complex scenes. Results show a significant recall improvement of 85.1% and a modest mAP50 increase of 0.6%, indicating enhanced target recognition, especially for distant targets. The PSAFF module also stabilizes detection in complex environments through a parameter-free spatial adaptive feature fusion mechanism, reducing missed detections.

In Case 4, applying MN3, VNACSP, and PSAFF simultaneously resulted in precision and mAP50 values of 87.8% and 91.4%, respectively, while reducing parameters and computational requirements by 86.8% and 81.6% compared to the baseline. However, recall slightly decreased to 82.4%, likely due to feature extraction redundancy from concurrent module operation, causing minor overfitting and hindering the detection of challenging targets. This emphasizes the need for improved balance among modules in future work to prevent overemphasizing specific features.

In Case 5, we applied the K-means clustering algorithm to optimize a priori frames, leading to greater improvements in precision (89.1%) and recall (85.1%), with mAP50 rising to 92.1%. This integration aligns a priori frames with target sizes, enhancing localization and classification accuracy.

A well-designed a priori frame can significantly boost model performance, particularly in detecting small targets with varying sizes.

Ablation experiments revealed the impact of various modules on model performance. Integrating these modules reduces computational complexity and significantly boosts detection performance, demonstrating practical value. Additionally, the model's lightweight design results in an 83.4% reduction in storage space for the generated file, offering a cost-effective solution for real-world deployment.

The GL-YOLOv5 safety helmet detection model's outcomes, shown in Fig. 10, demonstrate its ability to recognize helmet usage from various angles in complex environments accurately. It effectively identifies the helmet status of individuals in crowded situations, even when heavily obscured, and excludes irrelevant targets in non-riding scenarios (e.g., when pushing a bicycle), showcasing its robustness and high accuracy.

Figure 10: Test results

4.4 Comparative Experiments

In recent years, numerous innovative algorithms for target detection and optimization have been developed. As this paper was finalized, cutting-edge state-of-the-art (SOTA) models like RTDETR, YOLOv9, and YOLOv10 emerged, advancing target detection technology. We selected several well-known algorithms for our analysis to compare the optimization performance of these algorithms and to evaluate our GL-YOLOv5 model. To ensure the fairness of the experimental results, we tried to keep the same hyperparameter configurations during the comparisons as in the ablation experiments. Table 2 details the experimental data. Our experiments cover the baseline model (YOLOv5s) and the improved GL-YOLOv5, as well as a variety of state-of-the-art models such as RTDETR [29], YOLOv3-SPP [11], YOLOv7 [30], YOLOv8, YOLOv9 [31], and YOLOv10 [32]. RTDETR, introduced in 2024, is noted for its speed and accuracy; the RTDETR we used is based on ResNet50 as the backbone. YOLOv7, released in 2022, is recognized for its sophistication. We adjusted the depth multiplier and channel layers for YOLOv3-SPP and YOLOv7 to align with YOLOv5s specifications, resulting in YOLOv3s-SPP and YOLOv7s. YOLOv8 was introduced in 2023, while YOLOv9 and YOLOv10 represent the most recent state-of-the-art detection models proposed in 2024. For our comparative experiments, we selected the minimal versions of these models, specifically 8n, 9t, and 10n. These experiments aim to evaluate GL-YOLOv5 against both classical and contemporary algorithms.

Table 2: Comparison of multiple algorithms

Models	Params (M)	GFLOPS	Precision	Recall	$map_{0.5}$
YOLOv3s-SPP	9,566,319	23.3	85.4%	85.4%	90.1%
YOLOv5s (Baseline)	7,015,519	15.8	84.9%	83.6%	89.6%
YOLOv7s	9,137,726	26.0	86.7%	82.3%	89.7%
YOLOv8n	3,006,038	8.1	85.8%	82.5%	89.9%
RTDETR-ResNet50	41,938,794	125.6	88.8%	85.9%	89.3%
YOLOv9t	1,971,174	7.6	86.9%	84.2%	91.4%
YOLOv10n	2,707,820	8.4	86.4%	84.2%	91.2%
GL-YOLOv5	922,895	2.9	89.1%	85.1%	92.1%

The experimental findings indicate that while models such as YOLOv8n, YOLOv9t, and YOLOv10n may approach or exceed GL-YOLOv5 in certain performance metrics, they exhibit significantly higher computational complexity. GL-YOLOv5 demonstrates significant advantages in performance metrics. With a parameter count of only 922,895 M and a computational load of 2.9 GFLOPS, it operates efficiently even in resource-constrained environments. Notably, despite having a substantially lower parameter count than other models, GL-YOLOv5 achieves mAP50 at 92.1%. This performance surpasses that of YOLOv3s-SPP, YOLOv5s (Baseline), YOLOv7s, YOLOv8n, YOLOv9t, YOLOv10n, and RTDETR-ResNet50 models. These results highlight the computational efficiency of GL-YOLOv5, which reduces computational requirements and provides excellent detection accuracy.

To provide a more intuitive assessment of each model's performance, we present both quantitative visualization results in Fig. 11. Fig. 11a illustrates the trend of mAP50 changes over epochs during the training phase for GL-YOLOv5 compared to other models.GL-YOLOv5 demonstrates a stable growth trend during training, achieving the highest performance and confirming the effectiveness of its improvements. This stability reflects effective convergence and the model's adaptability and robustness to the dataset. Fig. 11b compares mAP50 with the number of parameters for each model. Despite fewer parameters, GL-YOLOv5 achieves a higher mAP50 than most models, demonstrating superior detection accuracy and a lightweight design. Its performance is impressive compared to more complex models, underscoring its efficiency and practicality in real-world applications.

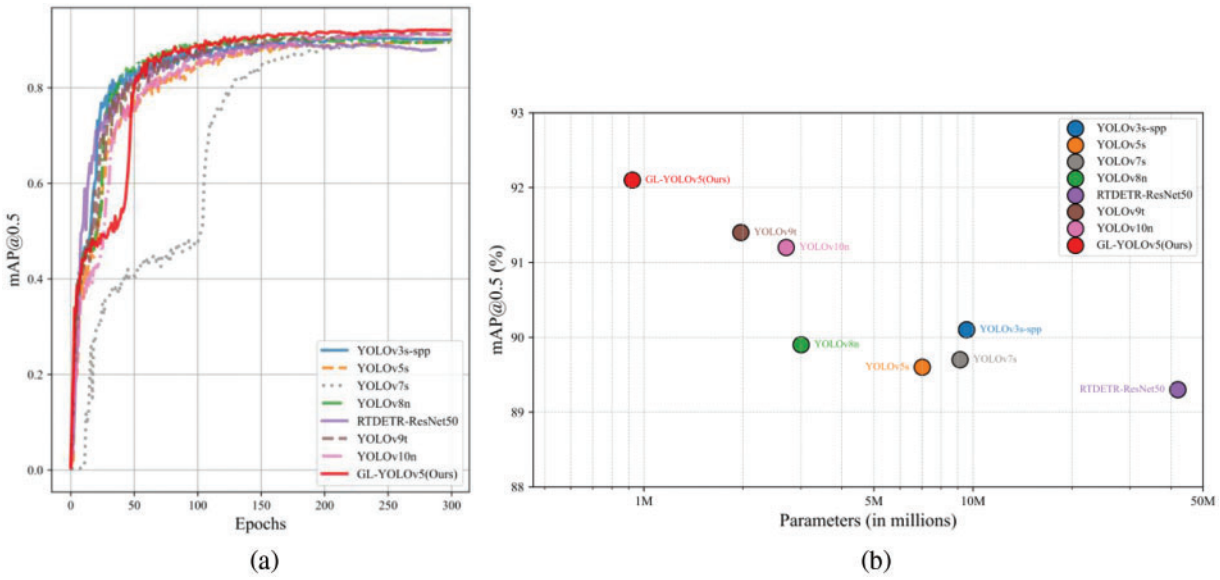

Figure 11: Quantitative visualization, with (a) showing a plot of GL-YOLOv5 against other models for mAP50 *vs*. epochs and (b) illustrating mAP50 *vs*. parameters

In Fig. 12, we show the detection results of different models for the same image. It can be seen that YOLOv3s-SPP, YOLOv5s, and YOLOv8n miss detection when dealing with heavily occluded targets on the left side, while YOLOv7s misses detection of occluded targets on the right side. YOLOv9t and YOLOv10n are underperforming, failing to detect the occluded targets on the left and right sides. Our GL-YOLOv5, on the other hand, performs well and successfully detects these occluded targets. The GL-YOLOv5 demonstrates its robustness and superiority in dealing with complex backgrounds and occlusion situations, emphasizing its excellent performance in real-world applications.

Figure 12: Comparison of model detection effects

In conclusion, GL-YOLOv5 shows significant promise for practical application scenarios due to its lightweight design and efficient detection capabilities.

5 Conclusion

In this study, we introduce a novel lightweight target detection model, GL-YOLOv5, which addresses the challenges associated with high computational costs and insufficient accuracy prevalent in existing algorithms. Notably, GL-YOLOv5 demonstrates improved performance in helmet detection, achieving a significant reduction in model size and parameter count while enhancing accuracy.

Despite advancements in model compression and detection precision, GL-YOLOv5 has its limitations. For example, although the model is optimized for lightweight performance, it may face performance constraints when dealing with complex targets in extreme conditions. Additionally, variability in training data and testing environments can affect the model's generalization capabilities.

Therefore, future research will enhance the model's robustness and accuracy across various complex scenarios by incorporating additional contextual information, refining data augmentation techniques, and exploring more advanced attention mechanisms.

In summary, GL-YOLOv5 not only provides an efficient and accurate solution for helmet detection tasks. It lays the groundwork for the future application of deep learning target detection methodologies on mobile and embedded platforms. This research offers innovative insights for achieving real-time and effective target detection, establishing a strong foundation for further advancements and applications in related fields.

Acknowledgement: None.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: Yuefan Liu, Ducheng Zhang; Data collection: Ducheng Zhang; Analysis and interpretation of results: Yuefan Liu, Ducheng Zhang; Draft manuscript preparation: Ducheng Zhang, Chen Guo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The code, dataset, and outcomes of this study can be obtained from the corresponding authors upon request.

Ethics Approval: Not applicable.

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

References

[1] Y. Chen, H. Wang, Y. Pang, J. Han, E. Mou and E. Cao, "An infrared small target detection method based on a weighted human visual comparison mechanism for safety monitoring," *Remote Sens.*, vol. 15, no. 11, 2023, Art. no. 2922. doi: 10.3390/rs15112922.

[2] S. Fang, B. Zhang, and J. Hu, "Improved mask R-CNN multi-target detection and segmentation for autonomous driving in complex scenes," *Sensors*, vol. 23, no. 8, 2023, Art. no. 3853. doi: 10.3390/s23083853.

[3] M. Stephan and A. Santra, "Radar-based human target detection using deep residual U-Net for smart home applications," in *Proc. 2019 18th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Miami, FL, USA, Dec. 9–12, 2019, pp. 9–12.

[4] WHO, *Launch of the Global Status Report on Road Safety 2023*. Geneva, Switzerland: WHO, 2023.

[5] B. Liu, R. Ivers, R. Norton, S. Blows, and S. K. Lo, "Helmets for preventing injury in motorcycle riders," *Cochrane Database Syst. Rev.*, vol. 43, no. 1, 2008. doi: 10.1002/14651858.CD004333.pub3.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, 2016. doi: 10.1109/TPAMI.2015.2437384.

[7] R. Girshick, "Fast R-CNN," in *Proc. 2015 IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 7–13, 2015, pp. 1440–1448.

[8] S. Ren, K. He, R. Girshick, J. Sun, "Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[9] 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. (CVPR)*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 779–788.

[10] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*, Honolulu, HI, USA, Jul. 21–26, 2017, pp. 7263–7271.
[11] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[12] W. Li *et al.*, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 11–14, 2016, pp. 21–37.
[13] A. G. Howard *et al.*, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 18–22, 2018, pp. 4510–4520.
[15] A. Howard *et al.*, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Republic of Korea, Oct. 27–30, 2019, pp. 1314–1324.
[16] C. Cui *et al.*, "PP-LCNet: A lightweight CPU convolutional neural network," 2021, *arXiv:2109.15099*.
[17] S. Mehta, M. Rastegari, L. Shapiro, and H. Hajishirzi, "ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 16–20, 2019, pp. 9190–9200.
[18] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 18–23, 2018, pp. 6848–6856.
[19] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "ShuffleNet V2: Practical guidelines for efficient cnn architecture design," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 8–14, 2018, pp. 116–131.
[20] Z. Chen, J. Yang, L. Chen, and H. Jiao, "Garbage classification system based on improved ShuffleNet v2," *Resour. Conserv. Recycl.*, vol. 178, 2022, Art. no. 106090. doi: 10.1016/j.resconrec.2021.106090.
[21] T. Li, J. Wang, and T. Zhang, "L-DETR: A light-weight detector for end-to-end object detection with transformers," *IEEE Access*, vol. 10, pp. 105685–105692, 2022. doi: 10.1109/ACCESS.2022.3208889.
[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 18–22, 2018, pp. 7132–7141.
[23] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 8–14, 2018, pp. 3–19.
[24] H. Li, J. Li, H. Wei, Z. Liu, Z. Zhan and Q. Ren, "Slim-neck by GSConv: A better design paradigm of detector architectures for autonomous vehicles," 2022, *arXiv:2206.02424*.
[25] K. Xu, "Show, attend and tell: Neural image caption generation with visual attention," 2015, *arXiv: 1502.03044*.
[26] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. R. Stat. Soc. Ser. C (Appl. Statist.)*, vol. 28, no. 1, pp. 100–108, 1979.
[27] D. S. Modha and W. S. Spangler, "Feature weighting in k-means clustering," *Mach. Learn.*, vol. 52, no. 3, pp. 217–237, 2003. doi: 10.1023/A:1024016609528.
[28] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, 2020, Art. no. 1295. doi: 10.3390/electronics9081295.
[29] Y. Zhao *et al.*, "DETRs beat YOLOs on real-time object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, Jun. 17–23, 2024, pp. 16965–16974.
[30] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*, Vancouver, BC, Canada, Jun. 17–23, 2023, pp. 7464–7475.
[31] C. Y. Wang, I. H. Yeh, and H. Y. M. Liao, "YOLOv9: Learning what you want to learn using programmable gradient information," 2024, *arXiv:2402.13616*.
[32] A. Wang *et al.*, "YOLOv10: Real-time end-to-end object detection," 2024, *arXiv:2405.14458*.