China Instrument and Control Society
Instrumentation
INSTRUMENTATION, Vol. 12, No. 1, March 2025
https://doi.org/10.15878/j.instr.202500211

Article

Two-wheeler Helment Wearing Detection Alogrithm Based on Improved YOLOv5

Xiao Han, Xianchun Zhou*

School of Electronics Information Engineering, Tianchang Research Institute, Nanjing University of Information Science and Technology, Nanjing 210044

* Corresponding author email: zhouxc2008@163.com

Abstract: In many non-motor vehicle traffic accidents in China, the main cause of injury or death for drivers is not wearing a helmet. Therefore, the detection and punishment of such riders hold great significance in protecting people's lives and property safety. This paper delves into a deep learning-based method for detecting helmet-wearing on electric vehicles. The approach involves studying and designing an improved YOLOv5 model to identify the violation behavior of not wearing a helmet, including inserting the SE module in the network of the visual attention mechanism into the enhanced backbone network; bidirectional feature fusion is significantly enhanced by substituting the concat module with the Bidirectional Feature Pyramid Network (BiFPN) module, and adding receptive field attention Convolution (RFAConv) to the detection head. The improved YOLOv5 model demonstrates a higher mean Average Precision (mAP) while achieving a relatively smaller model size. This method provides technical support for the real-time and accurate detection of non-vehicle helmet targets; its efficacy has been confirmed through analysis of experimental results. These findings suggest that this method can assist traffic management departments in supervising non-motor vehicles, carrying significant practical value and importance.

Keywords: YOLOv5; Object detection; Convolutional neural network; RFAConv

Citation: Xiao Han, Xianchun Zhou."Two-wheeler Helment Wearing Detection Alogrithm Based on Improved YOLOv5." Instrumentation 12, no.1 (March 2025). https://doi.org/10.15878/j.instr.202500211

1 Introduction

In recent years, as urban traffic continues to develop, non-motor vehicle accidents have become increasingly frequent. Many cyclists often overlook the importance of wearing helmets while riding, posing significant hidden dangers to traffic safety. Therefore, detecting and penalizing such riders is crucial in reducing the severity of road traffic accidents and ensuring the protection of human life.

Helmet-wearing detection for electric vehicles is mainly divided into traditional algorithms and deep learning algorithms. Traditional machine learning algorithms mainly include SVM(Support Vector Machine), KNN(K-Nearest Neighbors) and other classifiers based on HOG(Histogram of Oriented Gradients), LBP(Local Binary Pattern) and other features, and use image features to detect and classify the helmet area. Although these methods are simple and easy to implement, they rely on manual design features and have limited robustness and generalization ability. In recent years, object detection algorithms based on deep learning, such as YOLO(You Only Look Once), Faster R-CNN (Region-based Convolutional Neural Network), and SSD (Single Shot MultiBox Detector), have made significant progress in detecting the presence of the helmet directly in the image and identifying whether it is worn correctly. In particular, lightweight models (such as YOLOv5 and NanoDet) not only improve the detection accuracy but also take into account the real-time performance, which is suitable for embedded device deployment, and promote application development in this field.

The advancement of deep learning and object detection technology has led to the application of more intelligent systems based on the principles of deep learning in traffic recognition scenarios[7]. While traditional detection algorithms such as R-CNN and Fast R-CNN are utilized in the field, and Faster R-CNN performs well in accuracy, their practical applications are limited due to slow running speeds. As a result, the YOLO series of algorithms have been developed[6], attracting attention for their high detection speed and accuracy. Nevertheless, the YOLO algorithm still has limitations in dealing with small targets and targets with fuzzy boundaries, making it imperative to improve the algorithm for better results.

Currently, numerous research institutions and universities, both domestically and internationally, have conducted extensive research on YOLO series algorithms, with a focus on enhancing the detection performance of the YOLO algorithm. This includes efforts to improve its accuracy, speed, and network structure. The application of the YOLO algorithm spans specific fields such as intelligent transportation, security monitoring, and unmanned driving. Efforts are being made to enhance he algorithm's capacity for generalization so that it functions effectively in diverse scenarios and environments. Furthermore, there are studies dedicated to integrating the YOLO algorithm with additional technologies, such as deep reinforcement learning, and utilizing transfer learning to enhance its detection performance[5] further.

The reason for choosing YOLOv5 improvement in this article is that YOLOv5 has achieved a good balance between performance and speed and has relatively mature applications and extensive community support. While newer versions such as YOLOv8 offer improvements in accuracy and efficiency, YOLOv5's architecture is stable and easy to scale, making it efficient enough for many practical applications. In addition, the training and reasoning speed of YOLOv5 is fast, which can meet tasks with high real-time requirements. Therefore, in some specific scenarios, such as the identification of electric vehicle helmet wearing mentioned in this paper, the advantages of YOLOv5 are more prominent, especially when both high performance and development efficiency are needed.

The research content of this paper is structured as follows. Initially, the TWHD helmet detection dataset was obtained via network download to serve as the foundational dataset. Subsequently, 5448 images were randomly divided into a training set and a test set at a ratio of 4∶1, resulting in the final two-wheel helmet detection sample set. After reviewing relevant literature, the YOLOv5 model was reconstructed using the Pytorch framework within the deep learning environment, serving as a candidate model. The YOLOv5 model was then trained on the constructed dataset utilizing the Python programming language and the Pytorch deep learning framework. Finally, based on experimental analysis, the performance metrics of the YOLOv5 model were evaluated. Additionally, other classical object detection models were replicated, and comparative experiments were conducted using the control variable method. Building upon the YOLOv5 model, recent advancements in deep learning were incorporated. The proposed improvements and optimizations include: 1. Incorporating Receptive Field Attention Convolution (RFAConv) into the detection head; 2. Integrating the SE module from the visual attention mechanism network into the improved backbone network; 3. Enhancing bidirectional feature fusion by replacing the original concat module in the Neck network with the BiFPN module.

The contributions of this paper are as follows:

1. Replacing the conventional Convolution with Receptive Field Adaptive Convolution (RFAConv) in the detector's header can significantly enhance the model's adaptability to varying receptive fields and its capability for multi-scale feature extraction. By dynamically adjusting the range of the receptive field, RFAConv can more effectively capture the contextual information of the target, thereby improving detection performance in multi-scale and complex scenarios. This enhancement primarily addresses the limitations of traditional convolution methods, which have inadequate adaptability to changes in target size and shape under fixed receptive fields. Consequently, this improvement contributes to increased detection accuracy and robustness, particularly in scenes with small targets or complex backgrounds.

2. Incorporating SE modules (Squeeze and Excitation modules) into the backbone network substantially enhances the model's capability to emphasize the significance of feature channels. The SE module adaptively amplifies the representation of critical information channels by learning the weights of distinct channels while simultaneously diminishing redundant or less important feature channels. This enhancement primarily addresses the issue of inadequate modeling of inter-channel relationships in traditional convolutional networks, thereby improving the feature representation ability and detection accuracy of the model in object detection tasks, particularly in scenarios with complex backgrounds or multiple objects.

3. Replacing the conventional Concat operation with the BiFPN (Bidirectional Feature Pyramid Network) module in the Neck network can significantly enhance the efficiency and quality of multi-scale feature fusion. By incorporating a weighted fusion mechanism and a bidirectional feature flow design, BiFPN dynamically allocates weights across different feature layers and optimizes the interaction between high-level semantic information and low-level detail information. This enhancement addresses the limitation of the original Concat operation, which merely concatenates feature layers without fully leveraging inter-layer relationships. Consequently, it improves the effectiveness of feature fusion and substantially boosts the detection capability

and overall performance of the model in complex multi-object scenarios.

The first chapter serves as an introduction to this thesis. It outlines the rationale and significance of the chosen topic, reviews the current research status both domestically and internationally, delineates the research content of this paper, and concludes with an overview of the organizational structure. Chapter 2 provides an overview of the foundational theories pertinent to this study, with a particular focus on the theoretical underpinnings of convolutional neural networks. Chapter 3 delves into the TWHD image datasets and models, detailing the processes involved in data collection, annotation, and partitioning, followed by an introduction to common convolutional neural network architectures. Chapter 4 presents the experimental results and analysis, beginning with an outline of the experimental setup. It then proceeds to compare experimental outcomes, culminating in a multidimensional evaluation of the enhanced YOLOv5 model's performance, with a thorough analysis of the final results. Chapter 5 offers a comprehensive summary of the research content and methodology while also outlining potential future directions for this field of study.

2 Related Work

2.1 Artificial Intelligence and Deep Learning

Artificial Intelligence (AI) is the scientific discipline dedicated to developing systems that exhibit intelligent behavior, with the objective of emulating human cognitive functions such as learning, reasoning, perception, language comprehension, and decision-making. The concept of AI was formally introduced at the Dartmouth Conference in 1956 and has since evolved through various stages, including rule-based reasoning, knowledge engineering, and statistical learning.

AI is a multidisciplinary field encompassing numerous subfields, such as Machine Learning, expert systems, computer vision, and deep learning, which is a subset of machine learning. Deep learning employs multi-layer neural networks to achieve efficient modeling of complex problems, serving as one of the key technologies driving advancements in AI.

Deep learning originated from research into artificial neural networks. It utilizes a multi-layered neural network architecture to extract features from vast datasets and perform tasks like classification, prediction, or generation by mimicking the connectivity patterns of human brain neurons. Although initial developments in deep learning began in the early 1980s, progress was hindered by limited computational resources and data scarcity. Not until 2012 did the AlexNet model achieve a breakthrough in image recognition competitions, marking the beginning of a golden era for deep learning. Today, its applications span diverse domains, including speech recognition, natural language processing, recommendation systems, and autonomous driving.

2.2 Convolutional Neural Network Algorithm

Convolutional Neural Networks (CNNs) are deep learning models specifically designed for processing image data, with foundational concepts inspired by the study of biological visual systems. The origins of CNNs can be traced back to the 1960s when neuroscientists Hubel and Wiesel experimentally identified the receptive field mechanism in the cat's visual cortex, providing critical insights for subsequent neural network architectures. In 1980, Fukushima introduced a "recognition" model that incorporated a hierarchical structure akin to convolutional operations, though limited computational resources limited computational resources constrained its practical application. In 1998, LeCun et al. developed the LeNet-5 model, a pioneering convolutional neural network for handwritten digit recognition, which first integrated convolutional layers with pooling layers to achieve high accuracy. With significant advancements in computing power and data availability in the 2000s, the 2012 AlexNet model achieved a major breakthrough in the ImageNet competition, marking the beginning of a golden era for CNNs in computer vision. CNNs excel in extracting local features through convolutional operations and significantly reduce parameter counts via weight sharing, making them indispensable in tasks such as image classification, object detection, and semantic segmentation. The core architecture of cnn includes several key components as shown in Figure 1.

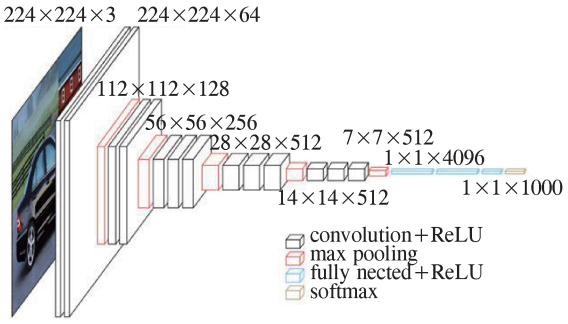

Fig.1 CNN model structure diagram

2.2.1 Convolutional Layer

The convolutional layer constitutes the fundamental building block of Convolutional Neural Networks (CNNs), serving to extract local features from the input data. For a two-dimensional input$X$and a two-dimensional convolution kernel K, the output Y resulting from the convolution operation is computed as follows:

$$Y(i,j)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}X(i+m,j+n)\cdot K(m,n)+b \qquad (1)$$

where (i, j) denotes the position of the output element, M and N represent the height and width of the convolution kernel, and b is the bias term. The convolution layer

applies the convolution kernel to the input via a sliding window mechanism to extract local spatial features.

2.2.2 Activation Functions

Activation functions introduce nonlinearity into the model, enabling it to learn more complex patterns. Common activation functions include ReLU (Rectified Linear Unit):

$$f(x)=(0,\max) \tag{2}$$

Effectively mitigate the issue of gradient vanishing.

2.2.3 Pooling Layer

The Pooling Layer serves to decrease computational complexity while preserving key features. Common pooling operations include: Max Pooling, which involves extracting the maximum value within a pooling window.

$$Y(i,j)=\max_{m,n}\{X(i+m,j+n)\} \tag{3}$$

Average Pooling: Taking the average value of the pooled window:

$$Y(i,j)$$

Where k is the size of the pooled window.

2.3 Object detection algorithm based on deep learning

Object detection algorithms based on deep learning have achieved significant advancements in recent years, with their primary objective being the simultaneous classification and localization of objects within images or videos. In contrast to traditional object detection methods, deep learning algorithms leverage convolutional neural networks (CNN) to automatically extract features from data, thereby markedly enhancing detection accuracy and robustness. The principal deep learning algorithms employed in object detection can be categorized into two main types: region-based detection algorithms and regression-based detection algorithms. Each approach possesses distinct characteristics suited to different application scenarios.

Region-based detection algorithms (R-CNN series): This category is exemplified by R-CNN (Region-based Convolutional Neural Network), which accomplishes the detection task by generating candidate regions followed by target classification and bounding box regression. The R-CNN series encompasses R-CNN, Fast R-CNN, Faster R-CNN, and Mask R-CNN. Region-based methods generally exhibit high detection accuracy and are particularly well-suited for scenarios requiring precise positioning and classification. However, the inference process tends to be slower due to the generation of candidate regions.

Regression-based detection algorithms (e.g., YOLO, SSD): These algorithms reformulate the object detection problem as a regression task, predicting both the category and bounding box of objects simultaneously across multiple grids of the input image. Notable examples include the YOLO (You Only Look Once) series and SSD (Single Shot MultiBox Detector). YOLO achieves object detection through a single network inference, offering exceptional speed and making it ideal for real-time applications such as autonomous driving and video analysis. SSD detects objects on multi-scale feature maps, further enhancing its capability to detect small targets. Compared to the R-CNN series, this class of algorithms offers slightly lower detection accuracy but significantly outperforms traditional methods.

2.3.1 Region-based Detection Algorithm

Region-based Detection Algorithms constitute a pivotal methodology within the domain of object detection. Notably, these algorithms are epitomized by the R-CNN (Region-based Convolutional Neural Network) series. Introduced by Ross Girshick and colleagues in 2014, R-CNN marked a groundbreaking advancement in object detection, pioneering the application of deep learning techniques to this field and achieving substantial performance enhancements. The fundamental principle underlying R-CNN involves initially generating a collection of region proposals via Selective Search. Subsequently, convolutional neural networks (CNNs) are employed to extract features from each candidate region, while support vector machine (SVM) classifiers predict the target categories. Concurrently, a regression model is utilized to refine the bounding boxes. The detection process of RCNN is shown in Figure 2.

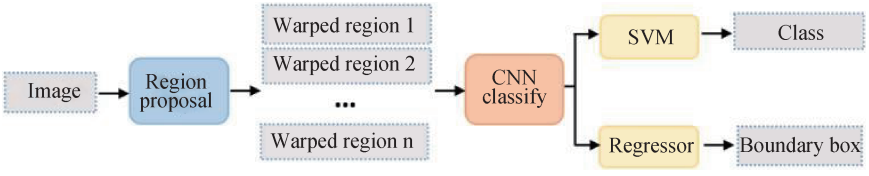

Fig.2 R-CNN detection process

While R-CNN has achieved a significant breakthrough in detection accuracy, its computational efficiency remains suboptimal. This inefficiency can be attributed to several factors: 1) CNN feature extraction is performed independently for each candidate region, leading to substantial redundant computations; 2) The use of separate SVM classifiers and bounding box regressors complicates the training process and increases the number of steps required; 3) The inability to optimize end-to-end limits further performance enhancements.

These limitations of R-CNN have guided subsequent algorithms towards improvements, giving rise to more

efficient models such as Fast R-CNN and Faster R-CNN. Fast R-CNN mitigates redundant computations by sharing feature extraction networks, while Faster R-CNN introduces Regional Proposal Networks (RPN) to enable end-to-end training for both candidate region generation and object detection. Despite its inherent inefficiencies, R-CNN's integration of candidate region detection with deep learning has laid the foundation for modern object detection algorithms.

2.3.2 Detection algorithm based on regression

YOLO (You Only Look Once) is a deep learning-based single-stage object detection algorithm initially introduced in 2016 by Joseph Redmon and colleagues. YOLO transforms the object detection task into a unified regression problem, predicting both the object category and bounding box coordinates within an image through a single network inference. This end-to-end architecture significantly enhances the speed of object detection, making it particularly suitable for real-time applications such as autonomous driving, video surveillance, and drone navigation.

The fundamental concept of YOLO involves dividing the input image into a grid of fixed size, with each grid cell responsible for detecting objects within its designated area. For each grid cell, the network predicts multiple bounding boxes along with their associated confidence scores and class probabilities. By leveraging global feature sharing, YOLO achieves superior inference speed compared to traditional region-based detection algorithms. However, early versions of YOLO faced limitations in detecting small and densely packed objects. Figure 3 shows the YOLO network model.

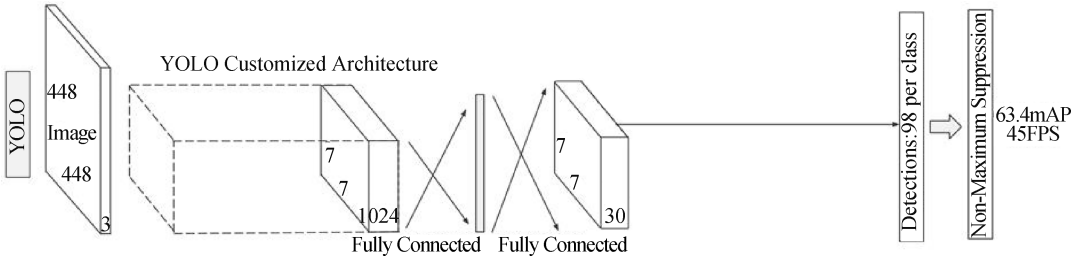

Fig.3 YOLO network model (Liu et al. 2016)

The initial version of YOLO was proposed in 2015 by Joseph Redmon et al. It approaches the task of object detection as a regression problem, directly predicting the bounding box and class probability of the object on the entire image. While YOLOv1 is fast, it is not effective at detecting small targets. Subsequently, YOLOv2 was introduced in 2016 with several improvements, such as utilizing Fully Convolutional Networks (FCN) to enhance detection accuracy, incorporating multi-scale feature maps for detection, and introducing Batch Normalization. Following this, YOLOv3 was released in 2018 to improve detection capabilities further. It implements a multi-scale prediction strategy to enhance detection performance by predicting bounding boxes of different sizes on feature maps of various levels. Additionally, YOLOv3 utilizes techniques like residual connections and Feature Pyramid Networks (FPN) to bolster model representation.

Building upon these advancements, YOLOv4 was unveiled in 2020 by Alexey Bochkovskiy and others. This release saw a further enhancement in detection performance through the incorporation of additional techniques like CSPDarknet53 as Backbone network, Mish activation function, and various data enhancement methods. Conversely, YOLOv5 emerged in 2020 from the Ultralytics team with a new architecture based on PyTorch that deviates from previous versions. Furthermore, YOLOv5 improves upon its predecessors' performance through multiple enhancements, including larger training data, larger network models, and more advanced training techniques.

YOLOX is an enhanced iteration of the YOLO series, released by MegVII in July 2021, aimed at optimizing object detection performance while simplifying model architecture. In contrast to traditional YOLO versions, YOLOX discards the Anchor mechanism prevalent in the YOLO series and adopts an Anchor-free design. This shift not only reduces computational complexity but also enhances the capability for detecting small targets.

The Meituan Vision AI Department unveiled YOLOv6[20] in September 2022. The architectural design incorporates an efficient backbone featuring RepVGG/CSPStackRep blocks, a PAN (Path Aggregation Network) topology neck, and an efficient decoupled head with a hybrid-channel strategy. YOLOv6 surpasses previous state-of-the-art models in terms of accuracy and speed. YOLOv7[21] was released in 2022. Upon its release, it demonstrated superior performance compared to many existing object detectors, with a range of processing speeds from 5 FPS to an impressive 160 FPS.

Ultralytics unveiled YOLOv8 in January 2023, marking a significant advancement in the YOLO series by offering users an extensive array of enhancements and versatile capabilities. YOLOv8[17] introduces five scaled versions to accommodate different application needs: YOLOv8n (nano), YOLOv8s (small), YOLOv8m (medium), YOLOv8l (large), and YOLOv8x (extra large)[16]. YOLOv8 represents the most recent iteration of the YOLO series, integrating the established architecture of YOLOv5 with

the innovative advancements from subsequent versions (such as YOLOv6 and YOLOv7) to achieve superior performance, efficiency, and user-friendliness. Notable enhancements encompass a streamlined network architecture, more effective parameter optimization, enhanced feature fusion capabilities, and improved adaptability to tasks including object detection, image segmentation, and key point detection. YOLOv8 introduces a dynamic anchor mechanism, an optimized loss function, and automated training hyperparameter tuning, which collectively result in significant improvements in detection accuracy and inference speed while reducing resource consumption. Compared to its predecessors, YOLOv8 strikes a better balance between accuracy and speed, making it an ideal choice for real-time detection tasks and devices with limited resources.

Through iterative refinement and enhancement, the YOLO series achieves an optimal balance between detection accuracy and computational efficiency. By integrating advanced techniques such as multi-scale feature fusion (e.g., Feature Pyramid Networks), adaptive anchoring mechanisms, sophisticated loss functions (e.g., CIoU loss), and optimized neural architectures, the detection performance and robustness of the YOLO algorithm have been markedly improved, particularly for small object detection. Consequently, YOLO has emerged as one of the leading algorithms in the domain of deep learning-based object detection.

2.4 YOLOv5 Algorithm Introduction

The YOLOv5 algorithm is a real-time deep learning-based object detection algorithm. In comparison to conventional target detection algorithms, YOLOv5 offers higher detection speed and accuracy . The network model demonstrates high detection accuracy and fast inference speed, with a detection speed that can reach up to 140 frames per second. Additionally, the file size of the YOLOv5 target detection network model's weight is relatively small, the YOLOv5 model is nearly 90% smaller than that of YOLOv4, indicating its suitability for deployment on embedded devices to achieve real-time detection. Therefore, the advantages of the YOLOv5 network include high detection accuracy, lightweight characteristics, and fast detection speed[4].

The YOLOv5 algorithm divides the entire image into multiple grid cells and assigns one or more boundary boxes to each grid cell[2]. Each bounding box predicts the category of the target, as well as its position and size. YOLOv5 employs convolutional neural networks to extract image features. These extracted features are then input into a fully connected layer for target classification and boundary box regression, enabling precise and accurate data processing.

By integrating and combining features at various levels, the target information across different scales can be effectively captured, thereby, improving the detection algorithm's performance in scenes with significant scale variations. In the following sections, we will present a detailed introduction to the network structure, loss function, and ghost convolution in YOLOv5. The experimental results show that the enhanced YOLOv5 algorithm delivers outstanding performance in image detection tasks.

2.4.1 YOLOv5 network structure

Given the pivotal role of accuracy, real-time performance, and lightweight characteristics in the helmet detection model, this study has enhanced the target recognition network for helmet detection in non-motor vehicles based on the YOLOv5 architecture. The YOLOv5 architecture encompasses four variations: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. The primary point of differentiation among them lies in the number of feature extraction modules and convolution nuclei at specific locations within the network. The size of the model and the number of model parameters increase progressively across these four systems.

The architecture of the YOLOv5 network consists of a sequence of convolution layers, pooling layers, and fully connected layers. The entire network is divided into multiple stages, each comprising specific modules and layers. This study's recognition model imposes stringent requirements on real-time performance and lightweight functionality; therefore, accuracy, efficiency, and scalability must be comprehensively taken into account during its design. Figure 4 shows the network structure of YOLOv5 6.0. The YOLOv5s framework comprises three primary components: the backbone network, neck network, and detect network[9]. The backbone network is a convolutional neural network that consolidates fine-grained images to generate image features.

The third layer of the backbone network is the C3 module, which is designed to extract deep image features and is primarily composed of Bottleneck modules. The C3 module is a new residual module introduced in YOLOv5. The YOLOv5 network structure combines residual connections and bottleneck structures to effectively enhance the expressive ability and convergence speed of the network. By replacing part of the standard convolution layer with C3 modules, we can increase the depth and width of the network, thereby improving model detection performance.

The ninth layer's SPPF (Spatial Pyramid Pooling Fast) module in the Backbone Network is constructed based on Spatial Pyramid Pooling (SPP). The SPP module was proposed by He et al. in 2015. The SPP module effectively avoids image distortion that may occur due to region clipping and scaling operations, solves problems related to repetitive feature extraction in convolutional neural networks, greatly improves candidate box generation speed, saves computing costs. The SPPF module offers improved efficiency as depicted in the figure.

The neck network serves as a feature aggregation

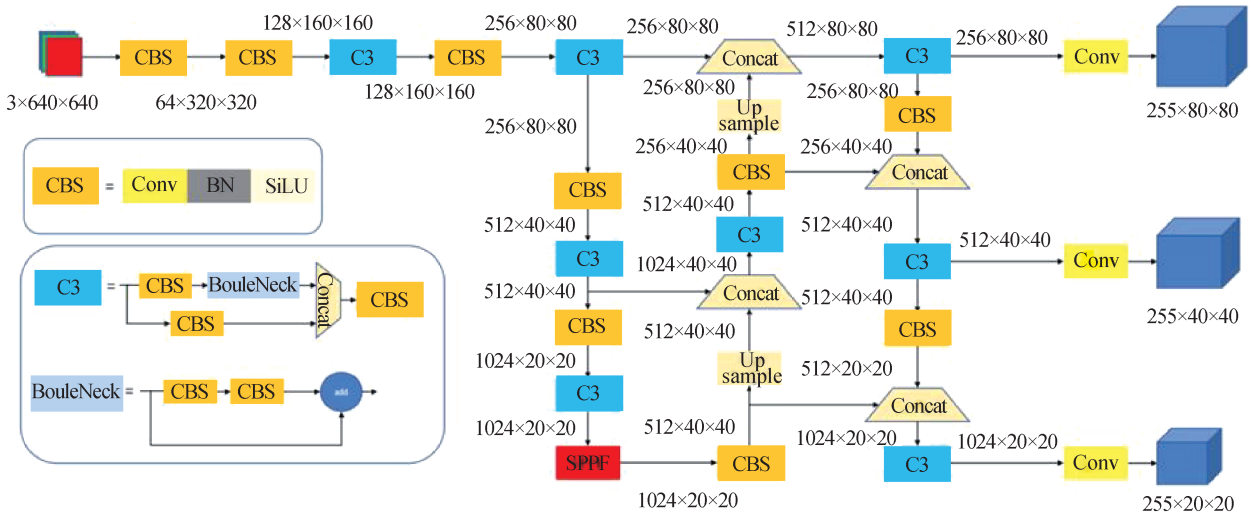

Fig.4 YOLOv5 6.0 network structure

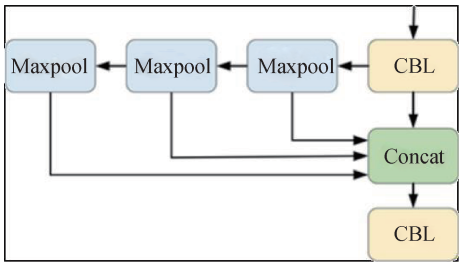

Fig.5 SPPF module structure

layer, combining a series of mixed image features. The primary function of the network is to generate the Feature Pyramid Network (FPN) and transmit the output feature map to the detection network. The network's feature extractor adopts the new FPN structure, which enhances the transmission capability of underlying features and improves the bottom-up path and detection capability for targets of different scales. This enables accurate identification of target objects with varying sizes and scales[1].

The detection network plays a pivotal role in the model's detection process by applying anchor boxes to the feature graph output from the previous layer. It produces a vector containing the class probability of the target object, object score, and position of the bounding box surrounding the object. In YOLOv5s architecture, the detection network comprises three detection layers with input feature maps sized at 80 × 80, 40 × 40, and 20 × 20, enabling it to detect image objects of various sizes. Each detection layer ultimately outputs a 21-channel vector that generates predicted bounding boxes and categories for targets in original images, facilitating helmet target detection in the image.

2.4.2 YOLOv5 Loss Function

The loss function holds great significance in the task of object detection, as it plays a vital role in measuring the difference between the predicted and actual frames. Backpropagation updates the network parameters, thereby improving the performance and accuracy of the model. The YOLOv5 algorithm utilizes multiple loss functions to address various aspects of detection tasks, including object classification loss, bounding box regression loss, and confidence loss[12].

Target classification loss primarily evaluates the presence or absence of target objects within the prediction box, as well as categorizing different classes of targets. YOLOv5 utilizes a cross-entropy loss function to compute target classification loss by minimizing the distinction between predicted classes and real classes.

Bounding box regression loss assesses the precision of detecting frame positioning. YOLOv5 adopts Mean Square Error (MSE) to optimize the detection box location by minimizing differences in the center coordinates, width, and height of prediction boxes.

Confidence loss gauges alignment between predicted boxes and real boxes. YOLOv5 calculates confidence using Binary Cross-Entropy (BCE) to improve confidence accuracy by minimizing disparities between predicted box confidence and real box confidence levels.

2.5 *Fusion of Multi-Scale Features*

Multi-scale feature fusion is a method that addresses the information fusion requirements in image detection tasks. Its underlying principle involves combining feature maps of different scales in an image through effective fusion to acquire feature representations with richer semantic information. In YOLOv5, multi-scale feature fusion is primarily achieved through the use of a feature pyramid. By establishing contextual connections between feature graphs at different levels, the network can Utilize both low-level and high-level features to their full advantage. The bottom and top feature maps are subjected to scaling through up-sampling and down-sampling, followed by feature fusion to generate a comprehensive feature map[8]. By consolidating feature maps of different scales, the network is able to comprehensively perceive target information within the

image, thereby enhancing detection accuracy and robustness in capturing targets with significant scale variations.

Multi-scale feature fusion generally encompasses two types: top-down and bottom-up fusion. Top-down fusion involves the combination of high-level semantic features with low-level detailed features, allowing for the extraction of relatively rich semantic information but weaker extraction of detailed information. Conversely, bottom-up fusion combines low-level detailed features with high-level semantic features, enabling more accurate extraction of detailed information at the expense of weaker extraction of semantic information[13]. In this paper, we modify YOLOv5's concat structure to facilitate bidirectional feature fusion in order to improve model performance.

3 Proposed Model Construction

In this chapter, we present a method for detecting non-motor vehicle helmets using an improved version of YOLOv5s. We have integrated the SE module into the visual attention mechanism network within the enhanced backbone network, improved bidirectional feature fusion, and added Receptive Field Attention Convolution (RFAConv) to the detection head. The SE module enhances the expressive capability of backbone features, while BiFPN improves the integration of multi-level features. RFAConv optimizes the receptive field adaptability of the detector head. Together, these three components form a comprehensive optimization pipeline from feature extraction, through feature fusion, to target detection. Specifically, BiFPN and RFAConv are particularly effective for multi-scale target problems by optimizing feature fusion and receptive fields, respectively, thereby enabling the network to perform well on both small and large targets.

The SE module, BiFPN's weighted fusion mechanism, and RFAConv's dynamic adjustment of receptive fields collectively introduce dynamic modeling capabilities into the network, allowing it to adapt to complex scenes more intelligently. Experimental results demonstrate that the improved model exhibits enhanced detection accuracy, especially in scenarios involving complex backgrounds, small targets, and occluded targets. The robustness of the network is significantly strengthened, making it more adaptable to various target detection tasks across different environments. Furthermore, inference efficiency is optimized through lightweight design, ensuring that computational overhead does not increase substantially compared to traditional designs. These optimizations comprehensively improve the performance of YOLOv5 across diverse tasks. The following section provides a detailed description of the adjustments made to the network structure and parameter settings, along with a specific design scheme.

The recognition algorithm for detecting non-motor vehicle helmets not only requires accurate identification of helmet targets in various complex environments, but also necessitates compressing the model size for deployment on hardware devices. In this study, we optimized and enhanced the backbone network architecture of YOLOv5s, resulting in a relative reduction in the number of network weight parameters and overall size, while preserving detection accuracy.

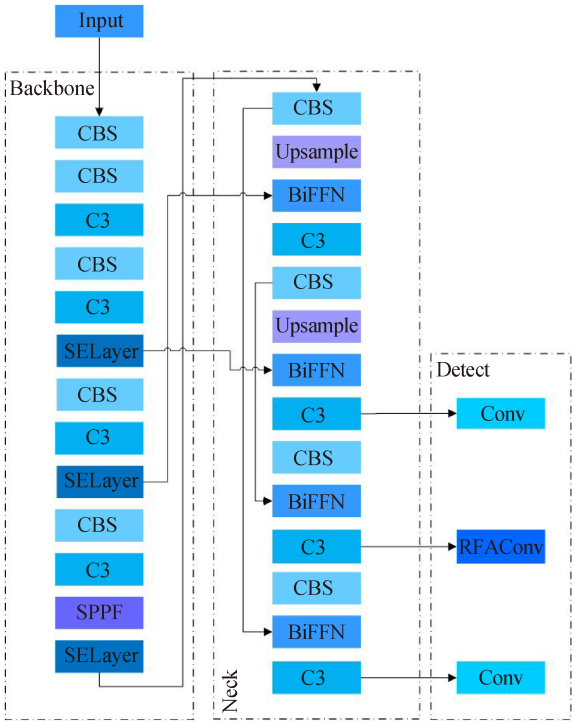

Fig.6 Improved YOLOv5 network structure

3.1 Squeeze-and-excitation module

To improve the accuracy of detecting the helmet target, we incorporate the attention mechanism from machine vision into the design of the target recognition network. The Squeeze-and-Excitation (SE) module[19] was introduced by Jie Hu et al. in 2018 in their seminal paper "Squeeze-and-Excitation Networks" (CVPR 2018). This lightweight and efficient attention mechanism aims to enhance the channel-wise relationship modeling capabilities of convolutional neural networks. The SE module refines feature representation through two key steps: 1. Squeeze: Global spatial information from each channel is aggregated[10], typically via global average pooling, compressing each channel's features into a scalar that captures the channel's global semantic information. 2. Excitation: A bottleneck structure composed of fully connected layers learns inter-channel relationships and generates a set of weights. These weights are subsequently applied to modulate the original feature strengths. Through these steps, the SE module effectively amplifies important features while suppressing less relevant ones.

The SE module significantly improves the network's ability to focus on critical features, thereby enhancing

task performance. Its low computational overhead allows for seamless integration into existing convolutional architectures such as ResNet and Inception, with minimal impact on the model's parameter count and inference time. Widely adopted in tasks like image classification, object detection, and semantic segmentation, the SE module has become an integral component of many deep learning models. Fine-grained channel modeling facilitated by the SE module has led to notable improvements in classification accuracy on datasets such as ImageNet, underscoring its innovation and practicality in the field of deep learning. Despite its low computational cost, this module effectively enhances model expression ability and optimizes learning content. Therefore, we integrated it into the enhanced YOLOv5 architecture, and a total of three SE modules were installed behind the second and third C3 modules and the final SPPF module of backbone network to improve model detection accuracy. Figure 7 shows the SE module structure.

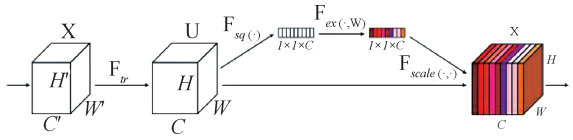

Fig.7 SE module

3.2 Receptive Field Attention Convolution

The received field attention convolution (RFAConv) is also used to replace the conventional convolution in the improved YOLOv5 model detection head. The Receptive Field Attention Convolution (RFAConv) module is an innovative convolutional architecture introduced in the field of computer vision, designed to enhance the capacity of convolutional neural networks (CNNs) to capture information across various scales and contexts. The fundamental concept involves dynamically modulating the contribution of different receptive fields within the feature map by incorporating a receptive field attention mechanism, thereby improving the network's representational power. Building upon standard convolution operations, RFAConv integrates a dedicated receptive field attention module that processes input features through multi-scale convolution kernels or a specialized weighting scheme. Specifically, it employs a variety of feature extractors with distinct receptive fields (e. g., varying scale convolution kernels) to process the input in parallel. An attention mechanism assigns weights to the features from different receptive fields, emphasizing critical regional information. These weighted features are then fused into the final output, enhancing adaptability to complex scenes. Existing spatial attention techniques, such as the Convolutional Block Attention Module (CBAM) and the Coordinated Attention Module(CA) [15], have been widely used in various applications, focus solely on spatial features and fail to fully address the issue of parameter sharing in convolution kernels[3]. In contrast, RFA not only concentrates on receptive field spatial features but also provides efficient attention weights for large convolution kernels.

By integrating diverse receptive fields, RFAConv can simultaneously capture both local details and global semantic information, making it well-suited for a wide range of visual tasks. The attention mechanism enables the network to adaptively adjust the contribution of each receptive field based on the input content, thus increasing model flexibility. Compared to traditional multi-branch architectures, RFAConv achieves significant performance improvements at a lower computational cost. It is particularly effective for tasks such as image classification, object detection, and semantic segmentation, especially in scenarios characterized by multi-scale features. YOLOv5 contains three detection heads. In this paper, considering the moderate size of the target image to be recognized, only the medium detection head is selected and replaced with the RFAConv module. Figure 8 shows the RFAConv module structure.

3.3 Bi-directional Feature Pyramid Network

The fusion of feature maps at different scales is a crucial method for enhancing the recognition performance of target detection networks. Low-level feature maps possess higher resolution and provide more detailed location information about the target object, but they lack semantic content and often contain more noise. On the other hand, high-level feature maps are rich in semantic information[14], yet their resolution is lower and their ability to perceive image details is relatively poor. Thus, effectively merging high and low-level features is essential for improving model detection performance.

In this paper, the improved YOLOv5 model uses the bidirectional feature pyramid network module (BiFPN) [18] to replace the four original Concat modules in the Neck network. BiFPN (Bidirectional Feature Pyramid Network) is a feature fusion module for object detection tasks introduced by the Google research team in their EfficientDet paper. The primary objective is to enhance the traditional Feature Pyramid Network (FPN) by integrating features of varying scales more effectively through bidirectional connections and weighted fusion mechanisms.

BiFPN facilitates bidirectional information flow in both the upper and lower paths of the feature pyramid network, incorporating top-down and bottom-up feature transfers. This mechanism enhances multi-scale feature integration. BiFPN adaptively adjusts the contributions of different features through learnable weighted fusion of various input features, as opposed to merely adding or averaging them. By eliminating redundant node connections and introducing lightweight depthwise separable convolutions, BiFPN substantially reduces computational complexity while preserving high performance.

BiFPN is more efficient than traditional FPN,

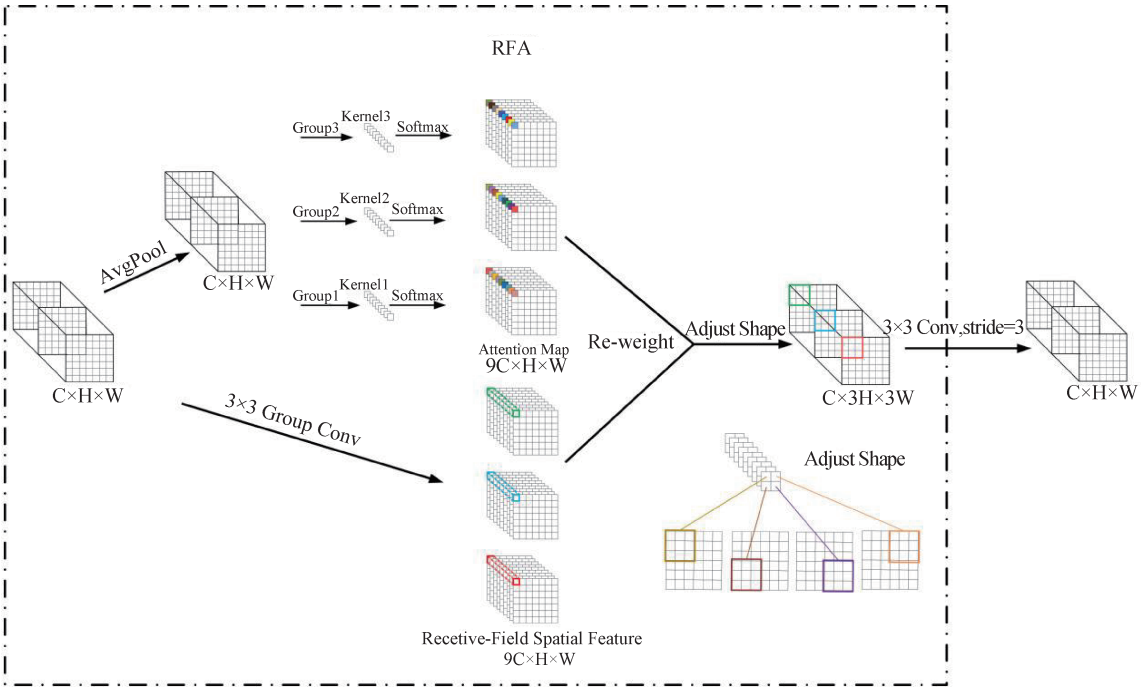

Fig.8 Receptive field convolution of attention (rfaconv)

suitable for resource-constrained scenarios, and achieves higher detection accuracy in object detection tasks, which is widely used in EfficientDet models and other lightweight deep learning networks.

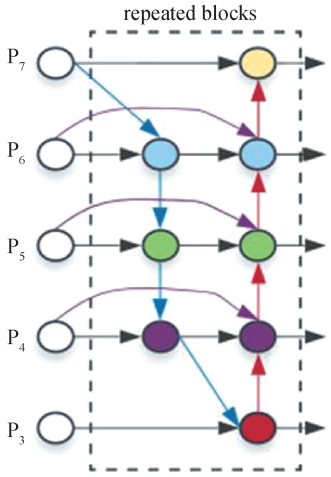

Fig.9 BiFPN network model structure

4 Results and Discussions

4.1 Evaluation Indicators

In this study, we have evaluated the trained object recognition model, employing objective performance indexes such as precision, recall rate, mean average precision (mAP), and F1 score.

In order to calculate the mAP value, it is first necessary to calculate the Precision and Recall of the model. The accuracy and recall are calculated as shown in formula (4) and (5).

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

Where *TP* represents a true case, *FP* represents a false positive case, and *FN* represents a false negative case.

Accuracy and recall rate are important indicators for evaluating the prediction performance of the model. The former represents the accuracy of the model's prediction results, while the latter reflects the model's ability to detect samples, that is, the recall rate. In addition, average accuracy (AP) is also a key indicator of the predictive performance of the model, and its calculation formula is shown in (6).

$$\mathrm{AP} = \frac{\sum_{i=1}^{n} P}{N} \tag{6}$$

Where *P* represents the prediction accuracy of the target category in each image, and *n* represents the number of images containing the target category. mAP is the average of AP values for all categories, and its calculation formula is shown in (7), where *N* is the number of total categories. In this paper, mAP@0.5 is selected as the experimental evaluation index, that is, the calculated mAP value when the threshold of IoU is set at 0.5.

$$mAP = \frac{\sum_{i=1}^{n} P}{N} \tag{7}$$

F1 score is a measure commonly used in classification problems, which provides a single numerical value that reflects the overall performance of the model by combining Precision and Recall, as shown in formula (8).

$$F1score = \frac{2 \times precision \times recall}{precision + recall} = \frac{2 \times TP}{2 \times \mathrm{TP} + FN + FP} \tag{8}$$

4.2 Datasets

Firstly, we need to choose the appropriate data set for training and testing the algorithm. For this study, the TWHD dataset was selected as the experimental dataset, which consists of 5448 images. The training set will consist of 80% of the data, while 20% will be used for the test set. The TWHD dataset is a combination of images from the OSF dataset, bike helmet dataset, and web crawler. Specifically, 4710 images were randomly extracted from the OSF dataset and re-labeled. Additionally, 738 images from both the bike helmet dataset and web crawlers were included to enhance the diversity of backgrounds in order to enable better distinction between bicycles and two-wheelers (electric vehicles and motorcycles) by neural networks. Figure 10 shows part of the TWHD dataset.

Fig.10 Presentation of the TWHD Dataset

In this paper, the input image size is 640×640. All images are annotated in Pascal VOC labeling format, capturing details such as whether two-wheeler vehicles with drivers/passengers are wearing helmets or not. Only riders/pillion riders on two-wheelers are labeled regarding their use of helmets; cyclists and pedestrians are not annotated for this attribute. This richly detailed database holds enormous potential for researching object detection algorithms related to identifying two-wheeled vehicles on road surfaces due to its substantial image data content along with corresponding annotation information.

Data enhancement is also implemented in this paper. The parameters hsv_h, hsv_s, and hsv_v are set to 0.015, 0.7, and 0.4, respectively, which control the degree of enhancement for hue, saturation, and brightness within the HSV color space. A translation value of 0.1 combined with a scaling factor of 0.5 signifies the extent to which the image is translated and resized. Additionally, a mosaic probability of 0.6 indicates the likelihood of performing image stitching.

Overall, this makes it an ideal resource suitable for further research in object detection algorithms related to identifying two-wheeled vehicles on road surfaces.

4.2.1 Final Distribution of Data

Figure 11 shows the label chart. Figure 12 shows the label correlation graph.

4.3 Experimental settings

4.3.1 Environment Configuration

The experimental environment established on my computer operated within the Windows 10 framework. The hardware configuration includes an Intel Core i5-9300H CPU @ 2.40GHz, 8GB of RAM, and an NVIDIA GeForce GTX 1650 graphics card. Most of the work described in this article, including code writing and commissioning, was conducted on my personal computer. However, due to its limited specifications, the detection rate is not optimal and the training speed is relatively slow. To ensure a comprehensive evaluation of performance, I also rented a cloud server for training and experimentation. This system is equipped with 64 gigabytes of memory, an Intel(R) Xeon(R) CPU E5-2686 v4, and an NVIDIA GeForce GTX 3090 graphics card.

In this experiment, we utilized Python version 3.9 along with Anaconda3; OpenCV version: 4.5.2; NumPy version: 1.26; PyTorch version: 1.12; and CUDA version: 11.3 for development purposes using PyCharm as our integrated development environment (IDE).

4.3.2 Strategies for Model Training

To facilitate data visualization during analysis, we employed TensorBoard as our analytical tool. In the training phase, the image Size of the input model is 640×

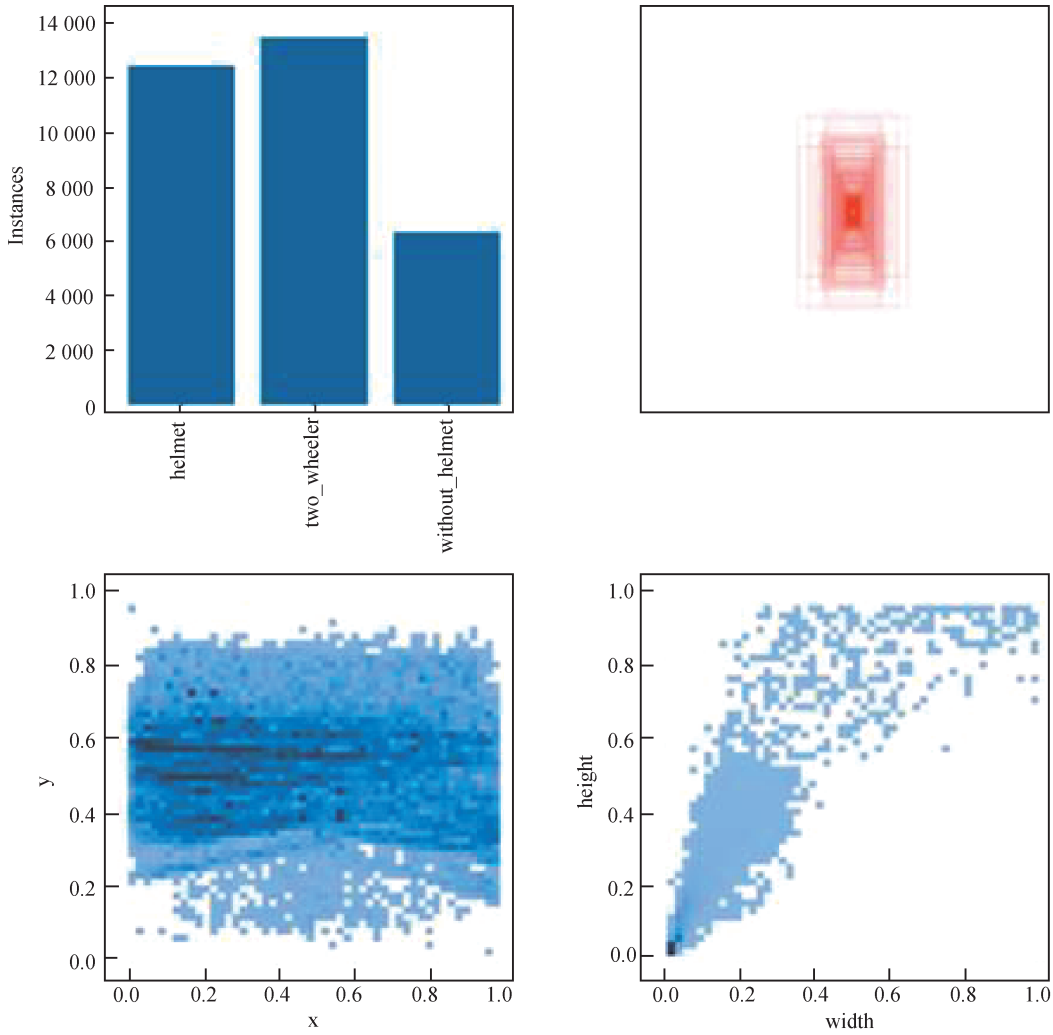

Fig.11 Label chart

640, and the Batch Size is set to 16. The initial learning rate is established at 0.01, while the final learning rate is set at 0.1, the momentum is set to 0.937, weight decay is fixed at 0.0001, and the optimizer employed is Stochastic Gradient Descent (SGD). The number of warmup epochs is set to 3.0, with an initial momentum of 0.8 and a starting bias learning rate of 0.1 for the warmup period. The weight for box loss is established at 0.05, while the weight for classification loss is set at 0.5.

4.4 Experiments results

Figure 13 and 14 present the recognition results of the proposed model on two-wheelers, with and without helmets. The pink frame represents the two-wheeler and the driver as a whole, while the red frame indicates the head of the driver wearing a helmet, and the orange frame highlights the head of the driver without a helmet.

In the figure, the left three are Loss curves of training set and verification set, and the right are evaluation indexes such as recall and precision, which will be used in this paper. loss is divided into cls_loss, box_loss and obj_loss[11]. cls_loss: The original text describes the use of monitoring for class classification and calculation to determine the accuracy of anchor frames and corresponding calibration classifications. A smaller value indicates a more precise classification result. box_loss: The task involves monitoring the regression of the detection box and calculating the error between the prediction box and the calibration box (CIoU). A smaller value indicates a more accurate prediction for the box. obj_loss: monitors whether objects exist in the grid and calculates the network confidence. If obj_loss is smaller, target detection is more accurate.

During the network model training phase, the iteration batch was set to 16. The training loss and validation loss curves[1] indicate a rapid decrease in the loss value during the initial 20 epochs of network training, followed by stabilization after approximately 50 epochs. As a result, in this study, the model output after 100 training sessions was established as the helmet target recognition model. The aforementioned chart demonstrates that the model is well-trained and does not suffer from overfitting.

4.4.1 Ablation experiment

To validate the effectiveness of the RFAconv

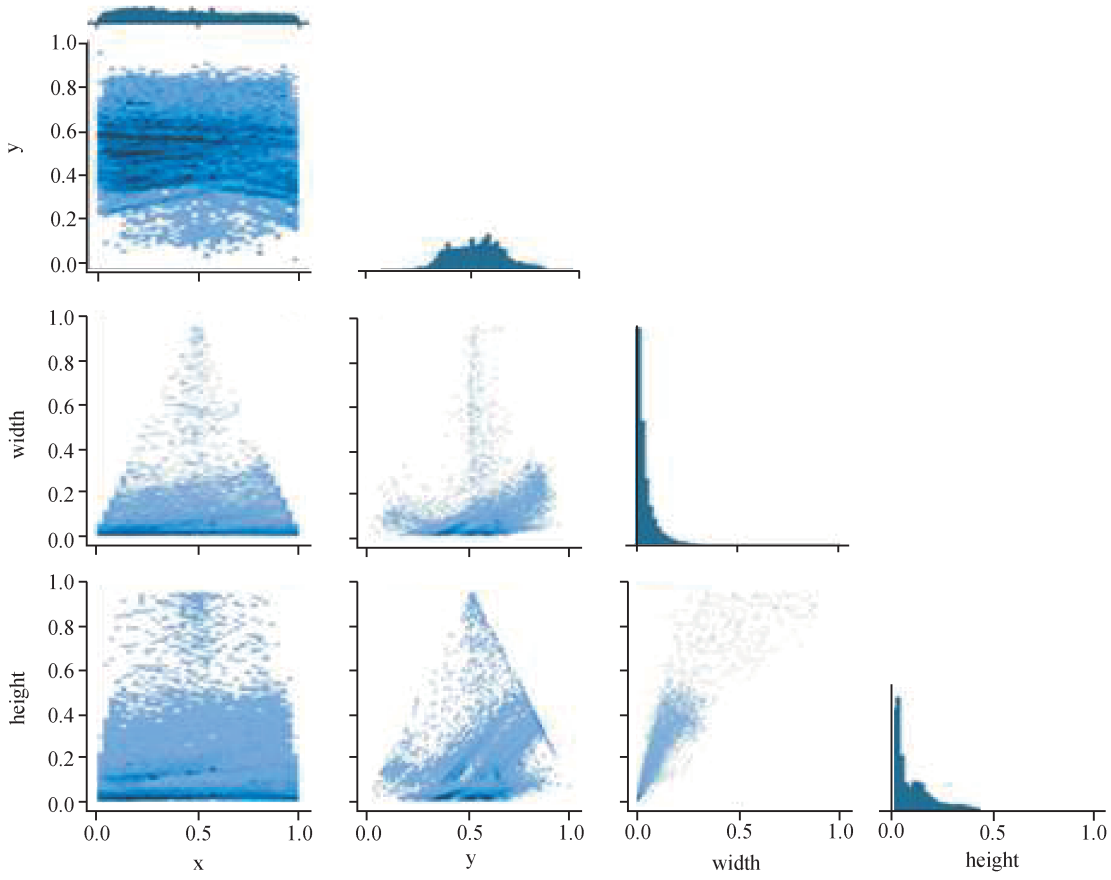

Fig.12 Label correlation graph

Table 1 Device information

Device Name	Device Information
System	Microsoft Windows 10
CPU	Intel(R) Xeon(R) CPU E5-2686 v4
GPU	GeForce RTX 3090
Memory	64 GB
Programming language	Python
Programming IDE	Pycharm
Deep learning framework	PyTorch

module in the network, this paper conducts an ablation experiment of the RFACONV module. Figures 15-19 present a comparison between the final algorithm proposed in this study and the algorithm for removing the convolution of receptive field attention.

Confusion matrix is a standard format for accuracy evaluation, which can be used to visualize the recognition performance of the convolutional neural network model obtained in this study for various categories of data. Each column of the confusion matrix represents the true category, each row represents the prediction category, and the diagonal value from the top left to the bottom right represents the probability of a correct prediction. Thus, the confusion matrix visually shows how the model confuses classes. In this paper, the confusion matrix of normalized processing is drawn as shown in Figure 15. In Figure 15 (a), the helmet recognition rate is 0.73, the non-motor vehicle recognition rate is 0.91, and the personal helmet-free recognition rate is 0.58. For figure 15 (b), they are 0.68, 0.91 and 0.51, respectively. It is clear that adding attentional convolution of receptive fields to our algorithm significantly improves the prediction accuracy compared to algorithms without this feature.

Figure 16 illustrates the precision rate-confidence curve, where the horizontal axis represents confidence and the vertical axis represents precision. The precision rate is defined as the proportion of correctly predicted positive cases by the classifier to all predicted positive cases. As confidence increases, both sides of the graph demonstrate improved precision. The precision of the left figure at a 0.5 confidence level is 0.885, while the precision of the right figure at the same confidence level is 0.863.

Figure 17 depicts the recall rate-confidence curve, with the recall rate referring to the proportion of correctly predicted positive cases by the classifier to actual positive cases. As illustrated in the figure, at a confidence level of 0.5, the recall rate on the left is recorded at 0.622, which surpasses the value of 0.585 presented on the right.

FIG. 18 depicts the F1 curve, with the F1 value serving as an index for comprehensively evaluating the

Fig.13 Examples of network model recognition results

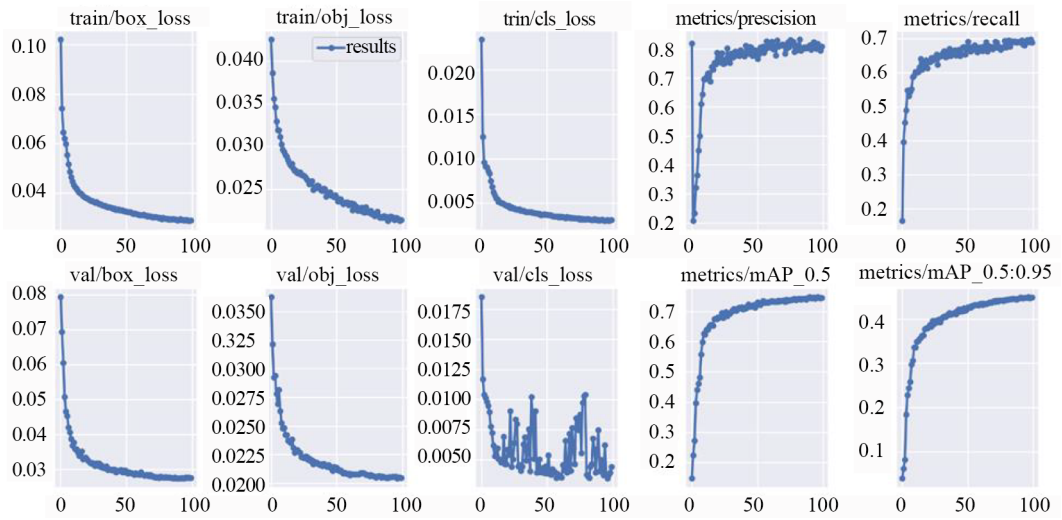

Fig.14 Loss curve comparison and performance index curve

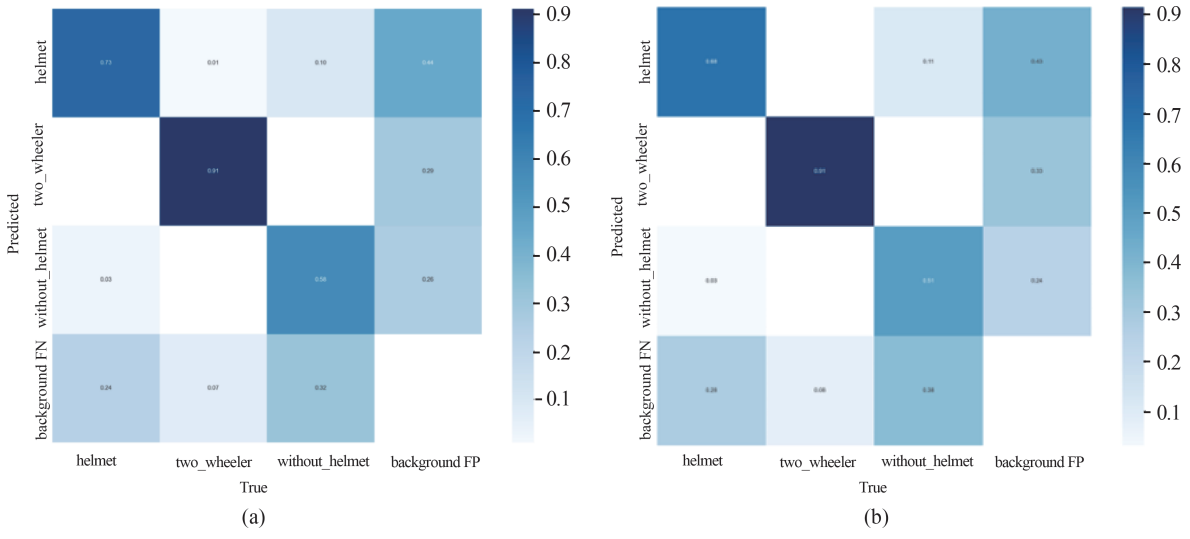

Fig.15 Confusion matrix (Normalization) (a) Enhanced Algorithm (b) Enhanced algorithm without RFAconv

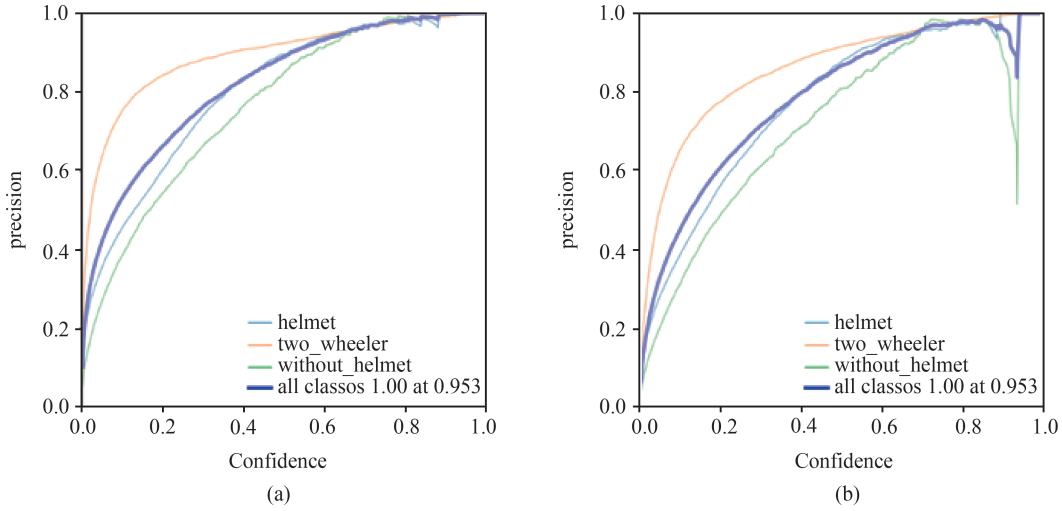

Fig.16 P_curve (a) Enhanced Algorithm (b) Enhanced algorithm without RFAconv

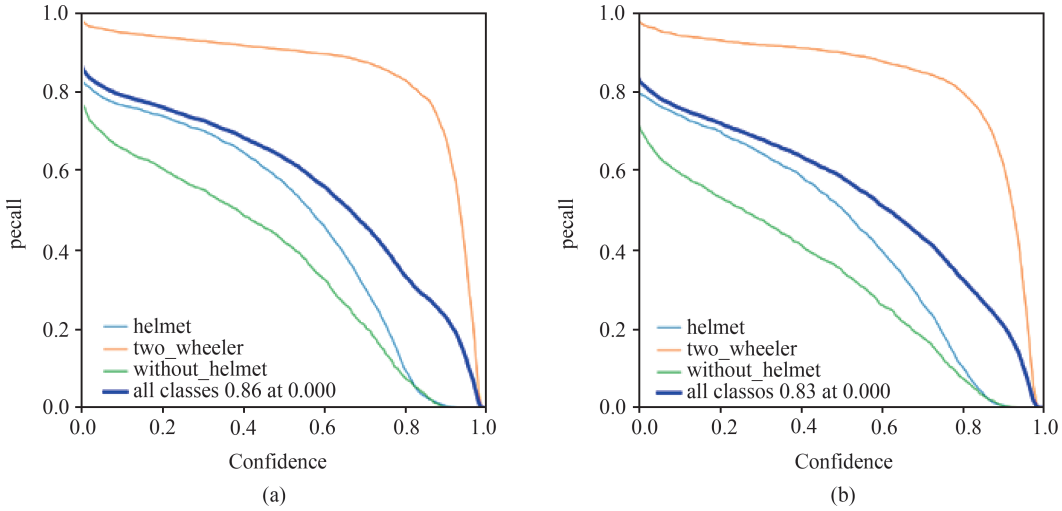

Fig.17 R_curve (a) Enhanced Algorithm (b) Enhanced algorithm without RFAconv

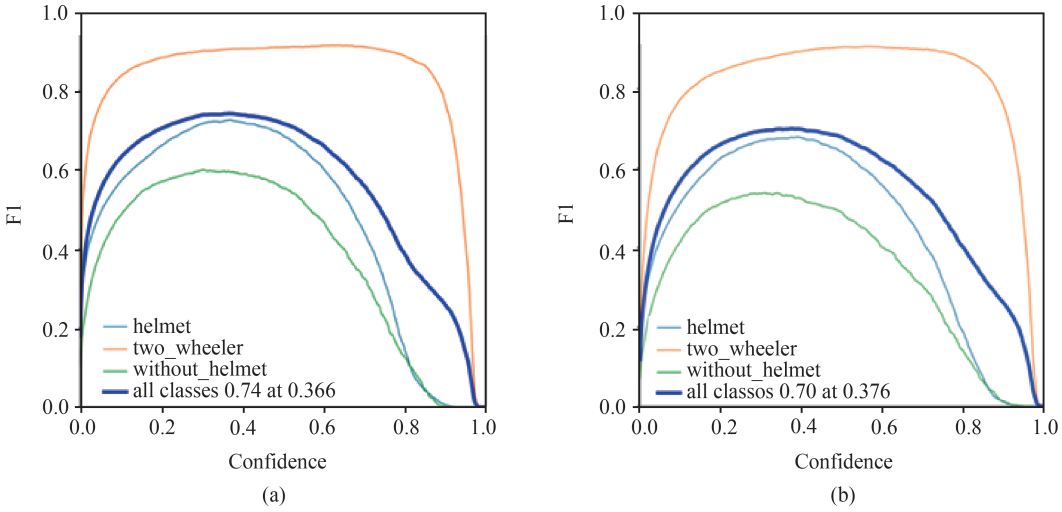

Fig.18 F1_curve (a) Enhanced Algorithm (b) Enhanced algorithm without RFAconv

performance of a classification model. It is utilized to gauge model performance while maintaining a balance between precision and recall. The range of the F1 value is from 0 to 1, where 1 indicates optimal performance and 0 indicates suboptimal performance. It is important to highlight that, at a confidence level of 0.5, the F1 score on the left side is 0.719, which surpasses the value of 0.681 observed on the right side.

Figure 19 depicts the precision-recall curve, which provides the basis for deriving the mAP value. Here, mAP represents mean average precision. As illustrated in the figure, a higher level of precision corresponds to a lower recall rate. The mean Average Precision (mAP) value for the left image is 0.747, exceeding that of the right image, which stands at 0.701. To sum up, the experimental results are shown in the table below.

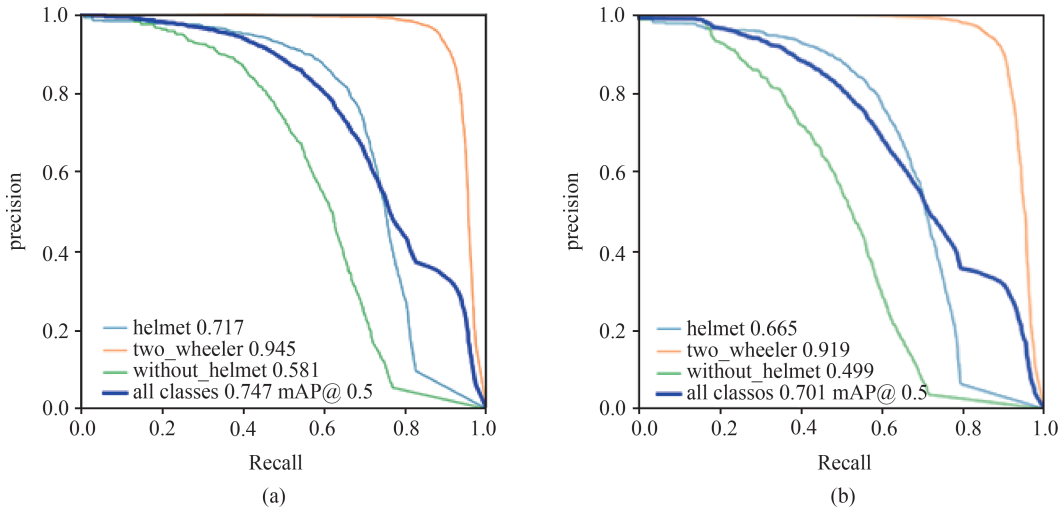

Fig.19 PR_curve (a) Enhanced Algorithm (b) Enhanced algorithm without RFAconv

Table 2 Comparison of ablation experiment

Network	Precision	Recall	mAP50	F1 Score
Ours	88.5	62.2	74.7	71.9
Ours (without RFAconv)	86.3	58.5	70.1	68.1

In summary, based on the findings presented in Figures 15 to 19, it is evident that all data from the final algorithm proposed in this study outperform those of the algorithm that omits consideration of the attention mechanism within the convolutional receptive field. This indicates that the RFAconv module significantly enhances the predictive performance of our algorithm.

4.4.2 Discussion and analysis

To sum up, in a test dataset containing 1090 images, precision, recall, F1 and mAP tests were carried out for three types of targets with different thresholds. The precision, recall, mAP, and F1 values for models using different confidence thresholds are shown in the table below.

Table 3 Comparison of target recognition outcomes

Network	Test set	Precision	Recall	mAP50	F1 Score
YOLOv5	helmet	90.4	64.1	79.6	74.1
	without_helmet	74.4	31.9	47.6	45.2
	two_wheeler	92.2	88.1	94.1	89.6
	total	85.6	59.3	73.8	70.2
Ours	helmet	89.6	56.7	71.7	68.9
	without_helmet	83.7	40.7	58.1	55.6
	two_wheeler	92.2	89.6	94.5	91.1
	total	88.5	62.2	74.7	71.9

This article establishes the confidence level at 0.5. The results show that the precision rate, recall rate, mAP value and F1 score of the proposed model are 89.6%, 56.7%, 71.7% and 68.9%, respectively. For the targets without helmets, the identification rates were 83.7%, 40.7%, 58.1% and 55.6%, respectively. The identification rates were 92.2%, 89.6%, 94.5% and 91.1%, respectively. The test results indicate that the proposed enhanced network model effectively identifies the target in the current image. The overall recall rate, precision rate, mAP and F1 scores were 62.2%, 88.5%, 74.7% and 71.9%, respectively, higher than the corresponding indexes of the original yolov5 model.

4.5 Model comparison

To further evaluate the target recognition performance of the proposed algorithm, we conducted a comparative analysis using a test set comprising 1090 images. This analysis compared the improved network against the original YOLOv5, YOLOX, YOLOv6, YOLOv7, and YOLOv8 models. The subsequent table presents the mean Average Precision (mAP) for each network model, along with the average recognition speed, model size, and parameter count.

YOLOv5 is currently one of the most extensively utilized object detection algorithms. Leveraging deep learning technology, it incorporates modules such as the Feature Pyramid Network (FPN) and Spatial Pyramid Pooling - Fast (SPPF) structure into the Convolutional Neural Network (CNN) framework, thereby achieving a balance between high accuracy and rapid detection speed. The experimental results obtained in this study are as follows: the mean Average Precision (mAP) is 73.8%, accuracy is 85.6%, recall rate is 59.3%, F1 value is 70.2%. The frame rate for real-time monitoring is 33

Table 4 Model comparison result

Network	mAP50	Precision	Recall	F1 Score
YOLOv5	73.8	85.6	59.3	70.2
YOLOX	72.1	88.7	58.1	63.5
YOLOv6	69.4	73.3	52.7	61.3
YOLOv7	76.4	72.3	60.2	65.7
YOLOv8	81.0	93.0	64.1	75.9
Ours	74.7	88.5	62.2	71.9

frames per second (fps), the number of parameters is 7.3 million, and the model size is 17.1 megabytes (MB).

YOLOX[22] is an advanced object detection algorithm built upon the YOLO architecture. It employs an anchor-free design, sophisticated data augmentation techniques, and refined training strategies to enhance detection accuracy and generalization capabilities substantially. Additionally, the introduction of a Decoupled Head and a more flexible network structure further improves the model's performance and efficiency. In this study, the performance indicators of YOLOv6 are as follows: average accuracy (mAP) is 72.1%, accuracy is 88.7%, recall rate is 58.1%, F1 value is 63.5%. real-time monitoring frame rate is 27 frames per second (fps), number of parameters is 9.0 million, and the model size is 26.8 MB.

YOLOv6 represents an advanced iteration within the YOLO series, significantly enhancing real-time target detection performance through optimized network architecture and the incorporation of adaptive convolution techniques. These improvements are particularly evident in the detection of small objects and in dense scene scenarios, where both speed and accuracy have been markedly improved. In this study, the performance indicators of YOLOv6 are as follows: average average accuracy (mAP) is 69.4%, accuracy is 73.3%, recall rate is 52.7%, F1 value is 61.3%. real-time monitoring frame rate is 22 frames per second (fps), number of parameters is 11.1 million, and the model size is 34.1 MB.

YOLOv7 is an advanced real-time object detection algorithm that builds upon and refines the strengths of its predecessors in the YOLO family. It achieves superior detection accuracy, faster inference speed, and enhanced generalization capability. The algorithm's innovative integration of novel network architectures, optimized loss functions, and sophisticated training strategies has led to outstanding performance across multiple benchmark datasets, especially in terms of the balance between speed and accuracy. In this study, the performance indicators of YOLOv7 are as follows: average accuracy (mAP) is 76.4%, accuracy is 72.3%, recall rate is 60.2%, F1 value is 65.7%. real-time monitoring frame rate is 24 frames per second (fps), the number of parameters is 10.7 million, and model size is 33.2 MB.

YOLOv8 represents the latest generation of YOLO-based object detection models developed by Ultralytics, delivering state-of-the-art performance. In this study, the performance metrics of YOLOv8 are as follows: the mean Average Precision (mAP) is 81.0%, accuracy is 93.0%, recall rate is 64.1%, F1 value is 75.9%. The frame rate for real-time monitoring is 20 frames per second (fps), the number of parameters is 11.9 million, and the model size is 34.9 MB.

The mAP value of the improved YOLOv5 model in this paper is 74.7%, accuracy is 88.5%, recall rate is 62.2%, F1 value is 71.9%. The detection rate is 25fps, the number of parameters is 10.5*10^6, and the model size is 32.1MB. The detection results of the enhanced YOLOv5 algorithm proposed in this study have been compared with those of the original algorithm. The findings indicate that the average precision (mAP) of the improved YOLOv5 model exceeds that of the YOLOv5, YOLOv6, and YOLOx models by 0.9%, 5.3%, and 2.6%, respectively. In addition, although our model has a slightly lower mAP value compared to recent advances such as YOLOv7 and YOLOv8, its size accounts for 96.7% and 92.0% of those models, respectively, and 94.1% of YOLOv6.

Additionally, the average recognition speed of the improved model stands at 25 frames per second (FPS), which meets the real-time detection requirements, and is 1.14, 1.04 and 1.25 times that of YOLOv6, YOLOv7 and YOLOv8, respectively.

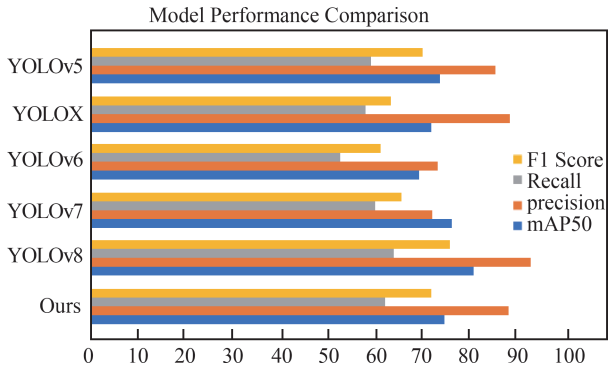

Fig.20 Model performance comparison

Table 5 Performance comparison of various target detection networks

Network	mAP50	Detection rate(fps)	Parameter quantity	Model size(MB)
YOLOv5	73.8	33	7.3×10^6	17.1
YOLOX	72.1	27	9.0×10^6	26.8
YOLOv6	69.4	22	11.1×10^6	34.1
YOLOv7	76.4	24	10.7×10^6	33.2
YOLOv8	81.0	20	11.9×10^6	34.9
Ours	74.7	25	10.5×10^6	32.1

The existing helmet-wearing recognition algorithm in highway monitoring can divide targets into more categories, thereby increasing the scale and complexity of the recognition model and extending network detection time. However, a lightweight model size is beneficial for future hardware deployment, while recognition speed directly impacts device efficiency. In this study, we introduce a relatively lightweight model that achieves a high mean Average Precision (mAP) value and maintains an average detection speed of 25 frames per second. This performance effectively meets the demands for real-time helmet recognition.

Further evaluation assesses algorithm stability and performance under complex scenarios involving interference factors such as occlusion and lighting changes, among others. The experimental results demonstrate that the enhanced YOLOv5 algorithm exhibits superior robustness in addressing these interference factors.

5 Conclusions

In this paper, we reproduce a variety of convolutional neural network models that exhibit high recognition accuracy and rapid convergence speed to address the issue of helmet detection for non-motorized cyclists on the road. First, we download an initial sample set of ship images from an open-source dataset. The data images are then divided into training and test sets in a ratio of 4∶1. Secondly, through literature research and theoretical analysis, we compare and analyze the advantages and disadvantages of classical convolutional neural network models. This includes YOLOv5, YOLOX, YOLOv6, YOLOv7, YOLOv8, as well as an improved version of YOLOv5. The improved YOLOv5 model integrates the SE module into the visual attention mechanism network in the enhanced backbone network, improves bidirectional feature fusion and adds receiving field attention convolution (RFAConv) to the detection head.

To verify the capability of the designed convolutional neural network model in recognizing ship targets, we developed code utilizing the existing GPU computing platform and PyTorch framework to implement the convolutional neural network. The performance of this model was evaluated from multiple dimensions. Experimental results indicate that the final accuracy achieved by the improved YOLOv5 is 74.7%. The experimental results show that the improved method significantly improves the accuracy and performance of target detection. However, there is still room for improvement. For example, while it is important in practical applications for devices to be able to work at night, the algorithm proposed in this study was specifically designed for target recognition during the day, thus limiting its applicability to helmet target recognition at night. In addition, due to the limited number of experimental samples and the potential impact of the experimental environment, it is necessary to expand the sample scope and optimize environmental factors in future studies to improve the robustness and accuracy of the improved algorithm.

The enhanced YOLOv5 model, developed through deep learning and convolutional neural networks as proposed in this paper, demonstrates the capability to automatically and accurately identify objects such as helmets. This advancement establishes a theoretical foundation for the application of deep learning within this domain. In this paper, the research on non-motor vehicle helmet image data has yielded promising results. The methodologies and conclusions presented herein can be analogously applied to other studies. The initial step involves collecting a substantial amount of image data in this domain and establishing a standardized and comprehensive image database, thereby providing robust data support for scholars both domestically and internationally.

Moving forward, efforts should focus on acquiring nighttime images using artificial lighting and adding them to the training set for model training purposes. This expansion will widen the application scope for our algorithm and device by enabling automatic recognition across both day and night scenarios. Furthermore, given that our algorithm facilitates real-time recognition of worn and unworn helmet targets, future work will also encompass object recognition based on enhancements made to the detection network structure. Moreover, exploring object recognition in UAV remote sensing images using our proposed detection network architecture is another area deserving of attention in future research endeavors.

Author Contribution:

Xiao Han: Conceptualization, Formal analysis, Investigation, Methodology, Software, Visualization, Writing-original draft; Xianchun Zhou: Writing-review & editing.

Funding Information:

This research received no external funding.

Data Availability:

The authors declare that the main data supporting the findings of this study are available within the paper.

Conflicts of Interest:

The authors declare no competing interests.

Publication Dates:

Received 14 July 2024; Accepted 04 February 2025; Published online 31 March 2025

References

[1] Yan B, Fan P, Lei X, et al. A real-time apple targets detection method for picking robot based on improved YOLOv5[J].

Remote Sensing , **2021**, 13(9): 1619.

[2] Li L, Wang Z, Gbh-yolov T Z. Ghost convolution with bottleneckcsp and tiny target prediction head incorporating yolov5 for pv panel defect detection., **2023**, 12, 561[J]. DOI: https://doi. org/10.3390/electronics12030561.

[3] Zhang X, Liu C, Yang D, et al. Rfaconv: Innovating spatital attention and standard convolutional operation[J]. *arXiv preprint* arXiv:2304.03198, **2023**.

[4] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. **2020**: 10781-10790.

[5] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEEconference on computer vision and pattern recognition. **2015**: 1-9.

[6] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks[J]. *Communications of the ACM* , **2017**, 60(6): 84-90.

[7] Chollet F. Xception: Deep learning with depthwise separable convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. **2017**: 1251-1258.

[8] Jin H, Pan Y, Lu J. Computer Networks and IoT[J].

[9] Jiang X, Hu H, Qin Y, et al. A real-time rural domestic garbage detection algorithm with an improved YOLOv5s network model[J]. *Scientific Reports* , **2022**, 12(1): 16802.

[10] Wei S, Qu Q, Wu Y, et al. PRI modulation recognition based on squeeze-and-excitation networks[J]. *IEEE Communications Letters* , **2020**, 24(5): 1047-1051.

[11] Yang H, Yang L, Wu T, et al. Automatic detection of bridge surface crack using improved Yolov5s[J]. *International Journal of Pattern Recognition and Artificial Intelligence* , **2022**, 36(15): 2250047.

[12] Sun Y, Zhong W, Li Y, et al. A Defect Detection Method of Drainage Pipe Based on Improved YOLOv5s[C]// International Conference on Applied Intelligence. Singapore: Springer Nature Singapore, **2023**: 144-155.

[13] *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 6th International Workshop* , BrainLes 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Revised Selected Papers, Part I [M]. Springer Nature, 2021.

[14] Li C, Cui G, Zhang W, et al. Defect detection in vehicle mirror nonplanar surfaces with multi-scale atrous single-shot detect mechanism[J]. *AIP Advances* , **2021**, 11(7).

[15] Yuhao X, Jian W, Yisu F, et al. An Enhanced YOLOv5s-Based Algorithm for Defect Detection in Steel Box Beam Sections[C]//2023 International Conference on the Cognitive Computing and Complex Data (ICCD). IEEE, **2023**: 23-27.

[16] Hussain M. Yolov1 to v8: Unveiling each variant-a comprehensive review of yolo[J]. *IEEE Access* , **2024**, 12: 42816-42833.

[17] Varghese R, Sambath M. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness[C]// 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). IEEE, **2024**: 1-6

[18] Tan M, Pang R, Le Q V. Efficientdet: Scalable and efficient object detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. **2020**: 10781-10790.

[19] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]// Proceedings of the IEEE conference on computer vision and pattern recognition. **2018**: 7132-7141.

[20] Li C, Li L, Jiang H, et al. YOLOv6: A single-stage object detection framework for industrial applications[J]. *arXiv preprint* arXiv:2209.02976, **2022**.

[21] 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[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. **2023**: 7464-7475.

[22] Ge Z. Yolox: Exceeding yolo series in 2021[J]. *arXiv preprint* arXiv:2107.08430, **2021**.