applied sciences

MDPI

Article

An Improved YOLOv5s-Based Helmet Recognition Method for Electric Bikes

Bingqiang Huang [1], Shanbao Wu [1], Xinjian Xiang [1,*], Zhengshun Fei [1], Shaohua Tian [2], Haibin Hu [1] and Yunlong Weng [1]

[1] School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China; bqhuang@zust.edu.cn (B.H.); 222107855033@zust.edu.cn (S.W.); zsfei@zust.edu.cn (Z.F.); 222107855012@zust.edu.cn (H.H.); 222107855032@zust.edu.cn (Y.W.)

[2] Key Laboratory of Intelligent Robot for Operation and Maintenance of Zhejiang Province, Hangzhou Shenhao Technology, Hangzhou 310023, China; hk7855762@163.com

* Correspondence: 188002@zust.edu.cn; Tel.: +86-15868153622

Abstract: This paper proposes an improved model based on YOLOv5s, specifically designed to overcome the challenges faced by current target detection algorithms in the field of electric bike helmet detection. In order to enhance the model's ability to detect small targets and densely populated scenes, a specialized layer dedicated to small target detection and a novel loss function called Normalized Wasserstein Distance (NWD) are introduced. In order to solve the problem of increasing model parameters and complexity due to the inclusion of a small target detection layer, a Cross-Stage Partial Channel Mixing (CSPCM) on top of Convmix is designed. The collaborative fusion of CSPCM and the Deep Feature Consistency (DFC) attention mechanism makes it more suitable for hardware devices. In addition, the conventional Nearest Upsample technology is replaced with the advanced CARAFE Upsample module, further improving the accuracy of the model. Through rigorous experiments on carefully constructed datasets, the results show significant improvements in various evaluation indicators such as precision, recall, mAP.5, and mAP.95. Compared with the unmodified YOLOv5s algorithm, the proposed enhanced model achieves significant improvements of 1.1%, 8.4%, 5.2%, and 8.6% on these indicators, respectively, and these enhancements are accompanied by a reduction of 778,924 parameters. The experimental results on our constructed dataset demonstrate the superiority of the improved model and elucidate its potential applications. Furthermore, promising improvements for future research are suggested. This study introduces an efficient approach for improving the detection of electric bike helmets and verifies the effectiveness and practicality of the model through experiments. Importantly, the proposed scheme has implications for other target detection algorithms, especially in the field of small target detection.

Keywords: intelligent transportation; electric bike helmet detection; YOLOv5s; CSPCM; NWD; CARAFE; DFC; small target detection

Citation: Huang, B.; Wu, S.; Xiang, X.; Fei, Z.; Tian, S.; Hu, H.; Weng, Y. An Improved YOLOv5s-Based Helmet Recognition Method for Electric Bikes. *Appl. Sci.* **2023**, *13*, 8759. https://doi.org/10.3390/app13158759

Academic Editors: Antonio Fernández-Caballero, Danilo Avola, George K. Adam and Jingsha He

Received: 16 June 2023
Revised: 20 July 2023
Accepted: 27 July 2023
Published: 28 July 2023

1. Introduction

The issue of electric bike helmet usage has emerged as a crucial concern for road traffic safety, along with the rapid growth of electric bikes in China. However, the detection of helmet wearing encounters numerous challenges and complexities in practical applications. Firstly, the detection algorithm must possess high precision to reduce the risks of misjudgment and missed detections owing to variations in helmet color, style, material, and other factors. Secondly, helmet wearing detection necessitates the consideration of intricate background interferences and demands swift detection capabilities to ensure both accuracy and practicality in helmet detection. Furthermore, helmet targets often constitute small components within the overall scene, presenting as diminutive objects [1,2], which pose an additional challenge to the accuracy of the detection algorithm.

The helmet target detection algorithms can be classified as conventional detection methods or deep learning-based detection approaches [3,4]. The former relies on visual detection techniques that utilize manually engineered features such as Histogram of Oriented Gradient (HOG) and adaptive Gaussian mixture models [5,6]. These methods involve separating targets, including helmets, electric bikes, and people, from the background before performing classification. This method not only affects detection accuracy, but also reduces computational speed, making it often unable to meet the real-time detection needs of traffic scenes. On the other hand, neural networks have made a big splash in deep learning and are widely used in areas such as image recognition, natural language processing, and reinforcement learning [7–9]. In terms of image recognition, these algorithms analyze and learn data features by simulating the operations of actual neural networks, offering improved detection performance compared to conventional methods.

Deep learning-based target detection networks can be broadly classified into two-stage networks and one-stage networks. Two-stage networks, such as Fast R-CNN [10] and Faster R-CNN [11], follow a two-step process. They generate a set of candidate regions within an image and subsequently perform feature extraction and classification on these candidate regions. While these networks achieve high accuracy, they suffer from limited real-time performance, large model sizes, and are not well-suited for traffic scenes.

In contrast, one-stage networks, including the YOLO series [12–15], SSD [16], and RetinaNet [17], take a different approach. These networks eliminate the step of generating candidate regions on the image and treat target region localization as a regression problem. They directly divide the image into a grid of candidate regions and predict the target's location and category in a single step, enabling end-to-end detection. Although the accuracy of these networks is slightly lower than that of two-stage networks, they have some advantages, such as reduced model size, improved detection speed, and better suitability for real-world scenarios.

2. Related Work

In recent years, a variety of researchers have proposed different algorithms for the detection of safety helmets. Lin et al. [18] proposed a CNN-based MTL method to identify and track individual motorcycles and detect helmet usage. They introduced the HELMET dataset, which consists of 91,000 annotated frames from 10,006 motorcycles in Myanmar. Hayat et al. [19] developed a real-time computer vision-based system for helmet detection utilizing the YOLOv5x architecture. Their objective was to achieve high accuracy in detecting helmets at construction sites, even under low-light conditions. Li et al. [20] proposed a safety helmet detection method based on a deep convolutional network. Their approach involved decoding video monitoring data, extracting YUV images, and applying a carefully designed convolutional neural network model for detection purposes. Wang et al. [21], on the other hand, utilized TensorFlow's image recognition framework to train a deep learning Single Shot MultiBox Detector (SSD) model and identified the helmet usage accurately. To solve the problems of safety helmet detection within complex factory environments, Sun et al. [22] developed a method that combined multi-feature fusion and Support Vector Machine (SVM) classification techniques. Yan et al. [23] designed a double-channel convolutional neural network (DCNN) model to enhance conventional image processing methods for detecting workers' helmets. Their focus was specifically on extracting image features using convolutional neural networks (CNN). Jia et al. [24] proposed an automatic helmet detection method for motorcyclists based on deep learning. They utilized an improved YOLOv5 detector to detect motorcycles from video surveillance and further determine if the motorcyclists were wearing helmets. They introduced a new motorcycle helmet dataset (HFUT-MH) that surpasses existing datasets in terms of size and comprehensiveness. In a study conducted by Li et al. [25], the problems of resource wastage and low monitoring efficiency in Container Freight Stations (CFS) caused by manual safety helmet detection were solved. They proposed a novel approach using the Broad Learning System (BLS) optimized by a Genetic Algorithm (GA) as an image recognition classifier. The GA-BLS

accurately identified CFS workers without safety helmets in video footage, thus achieving early warnings. Compared to the initial BLS and other methods such as Support Vector Machine (SVM), the GA-BLS achieves lower error rates and reduces operation time. This study provides an effective solution for enhancing safety helmet detection in CFS, optimizing resource utilization, and monitoring efficiency. Cheng et al. [26] introduced SAS-YOLOv3-Tiny, an innovative multi-scale safety helmet detection algorithm. This algorithm achieved an improved accuracy and model complexity trade-off while surpassing the original algorithm in terms of various metrics and computational efficiency. Lastly, Shine et al. [27] presented an automated system for real-time identification of motorcyclists without helmets from traffic surveillance videos. The authors compiled a custom dataset and proposed a two-stage classifier. The first stage extracts motorcycles, followed by a helmet identification stage. Two algorithms, one based on hand-crafted features and the other using a deep CNN, were presented for rider classification. The CNN-based model achieved higher accuracy, while the feature-based model enabled faster detection.

These studies have introduced a series of methods and concepts for detecting motorcycle riders without helmets. However, certain challenges still exist. Table 1 lists the limitations of these approaches.

Table 1. Related work and its limitations.

Related Work	Limitations
Lin et al. [18]	The initial stage of the method focuses on motorcycle detection. Once the motorcycle is detected, the algorithm continues to determine whether the motorcycle is in motion. Finally, in the last step, the algorithm performs helmet detection, which is a complex process that may result in a relatively slow execution speed.
Hayat et al. [19]	The main purpose of this method is to detect worker helmets in construction site scenes. It uses the YOLOv5x algorithm as the benchmark and adopts a large model size. Real-time requirements are not considered in this approach, and it is not designed for recognizing helmets in traffic e-bike scenarios.
Li et al. [20]	This method consists of multiple stages and has high algorithmic complexity. It does not prioritize real-time requirements in its design.
Wang et al. [21]	A design method for helmet system detection based on TensorFlow is proposed to address the issue of frequent accidents at construction sites. However, this method is not suitable for traffic scenarios.
Sun et al. [22]	A helmet detection method is proposed for factory scenes that combines multi-feature fusion and Support Vector Machines (SVM). However, this method has limited generalization ability due to the manual design of features and is not suitable for detecting helmets in traffic scenes.
Yan et al. [23]	Combining traditional machine learning methods with random forest (RF), an intelligent recognition algorithm based on DCNN and RF is proposed for worker helmet detection. It is not suitable for traffic scenarios.
Jia et al. [24]	The method consists of two steps: the first step uses the modified YOLOv5 detector to detect motorcycles from video surveillance; the second step takes the motorcycles detected in the previous step as input and continues to use the modified YOLOv5 detector to detect whether the motorcyclist is wearing a helmet or not. It is essentially a two-stage approach with high complexity and does not consider real-time requirements.
Li et al. [25]	The method utilizes a generalized learning system (BLS) optimized by a genetic algorithm as an image recognition classifier for helmet detection. However, the effectiveness of this method in recognizing helmets in traffic scenes is unknown. The system focuses on labeling and reminding construction site workers without helmets in videos rather than specifically targeting traffic scenarios.
Cheng et al. [26]	YOLOv3-tiny is used as the baseline algorithm to enhance the performance of the model. However, the resulting model size is still slightly larger than that of YOLOv5s, even with the improvements. Therefore, when deploying an application, a slightly larger storage space is required to accommodate the model.

Table 1. *Cont.*

Related Work	Limitations
Shine et al. [27]	The system uses a two-stage classifier to identify motorcycles in surveillance videos. The motorcycles detected in the first stage are then processed in the helmet identification stage. Two classification algorithms based on whether the rider wears helmets have been proposed. One algorithm relies on hand-crafted features, while the other uses deep convolutional neural networks (CNN). However, these models are more complex due to the inclusion of handcrafted features and may exhibit limited generalization ability.

Hence, the purpose of this study is to propose a deep learning-based algorithm for electric bike helmet detection, and the aim is to achieve efficient and accurate detection of electric bike helmets, thereby providing support for improving traffic safety.

3. YOLOv5s Algorithm

YOLOv5 is one of the more classic versions of the YOLO series, with significant improvements in speed and accuracy compared to earlier versions such as YOLOv3 and YOLOv4. The proposed model is based on YOLOv5 and aims to simplify deployment and production using the PyTorch framework. Although its accuracy may be slightly lower compared to the newer YOLOv7 [28], it provides faster detection capabilities and is very suitable for traffic scenarios.

The YOLOv5s model is the most lightweight variant among all versions of YOLOv5. Compared to other versions, it has the fastest model size and execution speed. Its architecture consists of four primary components: the image input module (Input), the backbone network module (Backbone), the feature fusion module (Neck), and the prediction module (Head). The architecture of the model is illustrated in Figure 1.

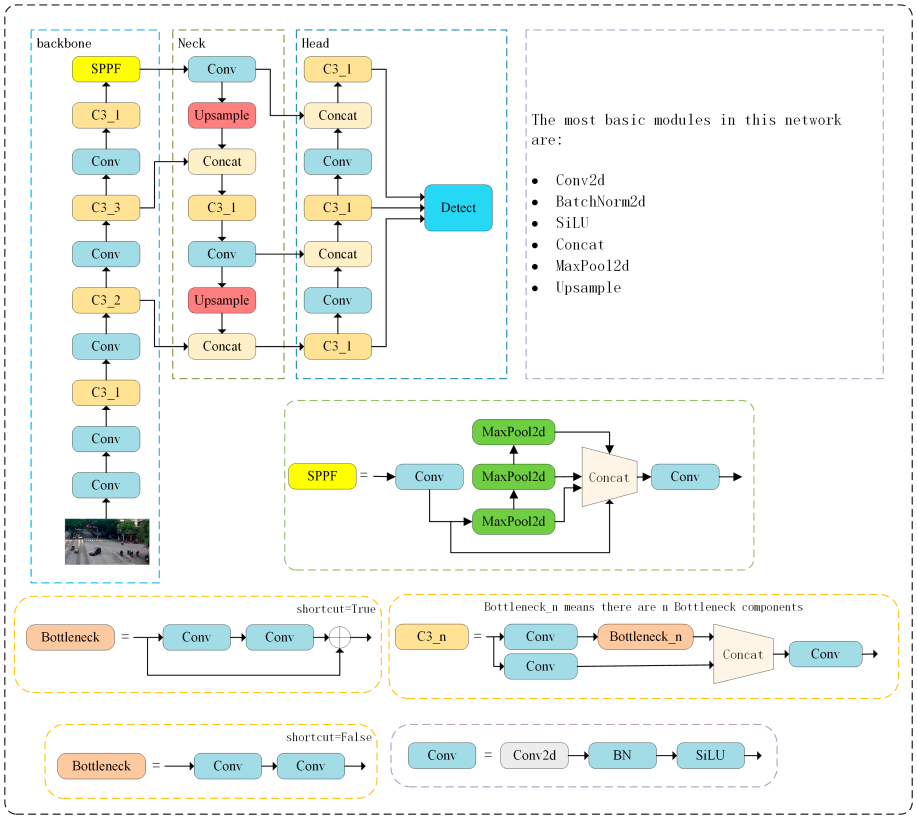

Figure 1. YOLOv5s network structure.

The Backbone Network module of the YOLOv5s model includes the darknet coding module from Darknet. Its main function is to extract relevant features from the input images. In addition, the feature fusion module consists of two sub-modules: Feature Pyramid Networks (FPN) [29] and Path Aggregation Networks (PAN) [30]. These sub-modules combine feature maps of different scales to create a comprehensive feature map rich in semantic information. Then, the fused feature map is transmitted to the prediction module, where object detection and classification tasks are performed.

The image input module of YOLOv5s includes several data augmentation techniques, including Mosaic data enhancement [31], automatic image cropping and stitching, and scaling, to preprocess the images. In addition, this module automatically calculates the most appropriate anchor frame for the model based on the size and aspect ratio of objects present in the training dataset.

When an image enters the backbone network module through the input module, the feature encoding module conducts three downsampling operations on the input image. For example, for an initial image size of 640 × 640, three consecutive downsampling operations of 8×, 16×, and 32× will generate three feature maps with different resolutions. These feature maps are subsequently fed into the feature fusion module.

The feature fusion module combines abstract semantic information and shallow feature details to merge and reconstruct three feature maps of different resolutions. It integrates the feature maps generated by the backbone network with the corresponding dimension upsampled feature map and then forwards them to the prediction module for regression and classification tasks.

The image prediction module consists of three prediction layers, each of which generates prediction frames for three different scale feature maps obtained from the feature fusion module. These prediction frames are subjected to non-maximal suppression to obtain the final localization results for the target object.

4. Materials and Methods

4.1. Image Acquisition

Due to the lack of a comprehensive open-source dataset specifically designed for electric bike helmet recognition, the dataset used in this study was independently created, consisting of 3035 images. Among them, 1909 images were extracted from road surveillance videos (as shown at the top of Figure 2). For this subset, one frame was extracted every 5 frames from the video to obtain the original samples. To ensure diversity, similarity comparison methods are used to filter out images with high similarity. The remaining 1126 images were sourced from internet images (as shown at the bottom of Figure 2). These images were labeled using the labeling tool, and the labels were divided into three categories: E-bike, with-helmet, and without-helmet.

Figure 2. Example of a dataset image.

For the experiments conducted in this paper, 2428 images were selected as the training and validation sets, while 607 images were selected as the test set. The visualization results of the dataset are shown in Figure 3. The upper left image shows the distribution of category classification, the upper right image shows the distribution of label boxes, the lower left image shows the distribution of label box centroid positions, and the lower right image shows the distribution of dataset sizes.

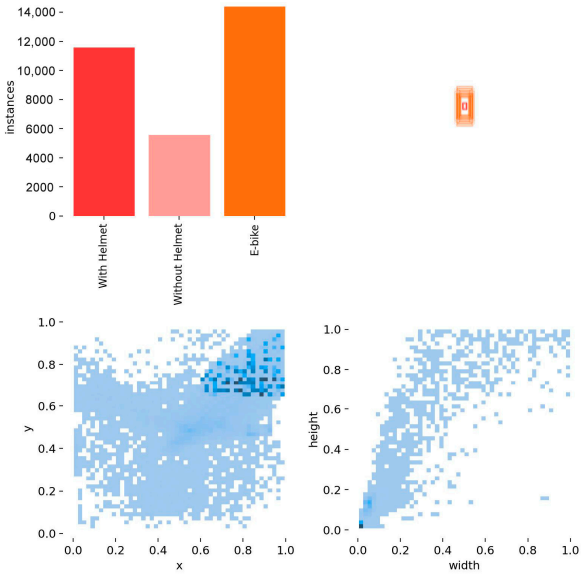

Figure 3. Visualization results of the dataset.

4.2. Improved YOLOv5s Helmet Detection Algorithm

In this section, the feature extraction module and feature fusion module of the YOLOv5s object detection framework were optimized and improved to achieve the best balance between detection performance and lightweight design. The overall network structure is shown in Figure 4.

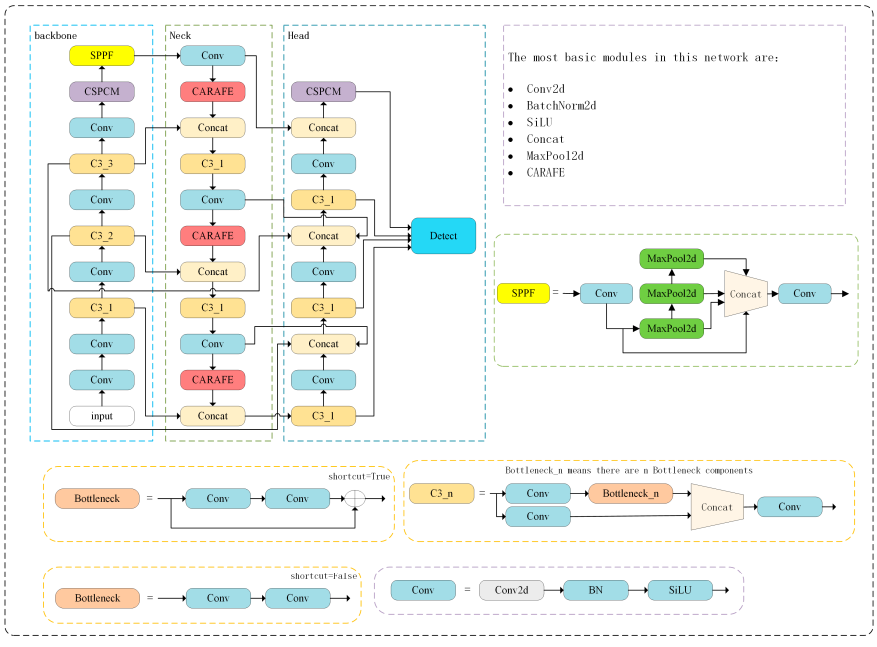

Figure 4. Algorithm structure in this paper.

In order to enhance the feature expression ability of the model and improve its detection ability for targets of different sizes and locations, the Cross-Stage Partial Channel Mixing (CSPCM) module was introduced in layer 8 and layer 30. This module combines the Deep Feature Consistency (DFC) attention mechanism to facilitate the fusion of feature channels. The addition of CSPCM helps improve feature representation in the model.

Furthermore, the original upsampling operation in the YOLOv5s model was replaced with the Content-Aware ReAssembly of Features (CARAFE) module [32]. CARAFE can adaptively learn the relationships between individual pixels, thereby reducing information loss during feature recombination. This enhancement significantly improves the model's detection ability for small targets while also reducing false positives and missed detections.

In addition, considering the size and location of the target, the Normalized Wasserstein Distance (NWD) loss function [33] was introduced. This loss function improves the detection accuracy of small targets. Moreover, a dedicated small target detection layer was added to optimize feature extraction and detection of small targets. This improvement further improves the model's detection accuracy for small targets.

In summary, these optimization improvements ensure that the model in this paper achieves the optimal balance between detection performance and lightweight design. It has excellent practicality and significant application value.

4.2.1. Multi-Scale Small Target Prediction Layers

The initial YOLOv5 architecture only includes three prediction layers for processing the detection of three different object categories, namely large, medium, and small targets. However, when faced with real-life scenarios involving helmets, the pixel ratio is often very small. Therefore, the small target prediction layer present in YOLOv5 often finds it difficult to effectively handle this situation. In order to address the challenge of detecting small electric bike helmets, this paper introduces a new small target detection layer to improve recognition accuracy. This layer is merged into three different scale detection layers to achieve multi-scale detection.

As shown in Figure 4, after the 18th layer, the feature map undergoes a series of upsampling operations and additional processing to enable continuous expansion, resulting in a 160×160 feature map. Then, the extended feature map is combined with the second layer of the backbone network to perform feature fusion so as to generate a larger feature map specifically designed for detecting small targets. Subsequently, the feature map is downsampled to generate four different scale detection layers. The sizes of these detection layers are 160×160, 80×80, 40×40, and 20×20, arranged from small to large according to the size of the detected targets.

This approach not only deepens the network structure and captures deeper feature information but also enables more accurate detection of small targets, thereby improving the overall performance of the detection algorithm.

4.2.2. Modify The CRARFE Upsampling Method

Traditional upsampling methods such as Nearest Upsample [34] and Linear Upsample [34] mainly focus on spatial pixel positions when determining the upsampling kernel. These methods have limitations in fully utilizing semantic information from the feature maps, resulting in limited perception domains. Deconvolution is another upsampling technique that has two key drawbacks: it applies a unified kernel throughout the entire image, ignores local changes, and involves a large number of parameters.

To address the issue of insufficient semantic correlation in upsampling, this paper introduces CARAFE as an alternative to Nearest Upsample. CARAFE is a pixel-level semantic segmentation network upsampling method that reconstructs feature maps by learning pixel correlations, resulting in higher resolution feature maps. CARAFE consists of two essential modules: the Kernel Prediction Module (KPM) and the Content-Aware Reassembly Module (CARM). The information flow of CARAFE is shown in Figure 5.

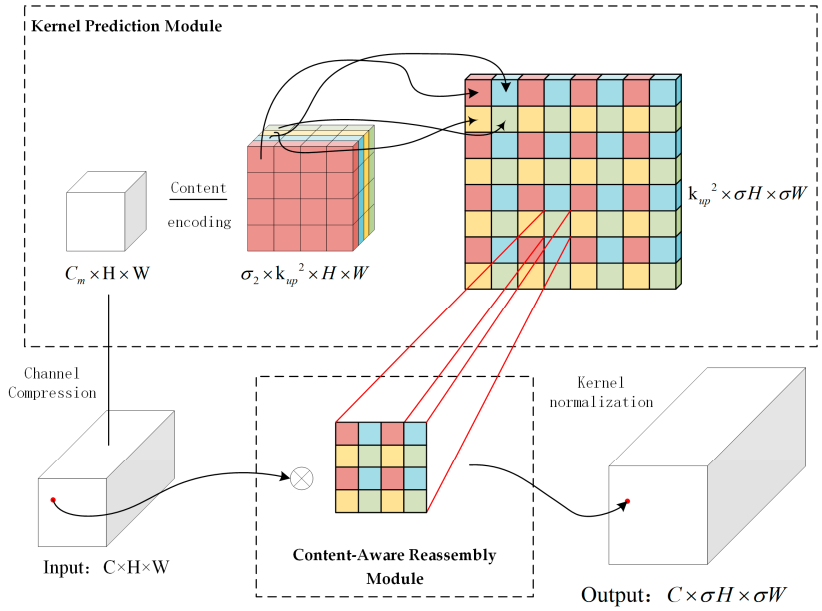

Figure 5. CARAFE upsampling module.

The KPM utilizes separable convolution to generate convolutional kernels for downsampling input feature maps and convolving them with the downsampled feature maps. Initially, the input feature map of size $H \times W \times C$ is decomposed using a 1×1 convolutional layer for channel compression. This compression reduces the number of channels to C_m, thereby reducing the number of parameters used for subsequent operations. The compressed feature map is then convolved using a convolutional layer with dimensions $K_{encoder} \times K_{encoder}$. This convolutional layer is responsible for line content encoding. Furthermore, the feature map is extended in the spatial dimension to obtain an upsampling kernel of size ${k_{up}}^2 \times \sigma H \times \sigma W$. The upsampling kernel is normalized using the Softmax function to ensure that the sum of the convolutional kernel weights is equal to 1. This process helps to learn and reconstruct pixel correlations within the convolutional kernel. The generation of convolution kernels depends on two key hyperparameters: k_{up}, which determines the size of the reconstructed convolution kernel, and σ, which represents the reconstruction factor.

The CARM is responsible for reconstructing the feature maps from the previous layer to achieve upsampling. It begins by applying a pixel shuffle operation [35] to the downsampled feature map, followed by converting it to a higher-resolution feature map using separable convolution. The resulting feature map is then normalized using the Softmax function and divided into smaller blocks. A dot-product of each pixel in the upper feature map and its corresponding smaller blocks is performed, resulting in an output feature map of size $C \times \sigma H \times \sigma W$.

By integrating KPM and CARM modules, CARAFE achieves pixel-level feature recombination to generate higher resolution feature maps. In practical segmentation tasks, CARAFE has demonstrated excellent segmentation accuracy and computational efficiency in various scenarios.

4.2.3. Loss Function Improvement

(1) NWD loss

In this paper, a new loss function, called Normalized Gaussian Wasserstein Distance (NWD), is introduced into the target detection model based on YOLOv5 to measure the distance between the predicted frame and the ground live frame. The NWD loss uses a Gaussian distribution to calculate the distance between two probability distributions and normalizes them to ensure comparability. For two 2D Gaussian distributions

$\mu_1 = N(m_1, \sum 1)$ and $\mu_2 = N(m_2, \sum 2)$, the second-order Wasserstein distance between μ_1 and μ_2 is calculated as shown in Equation (1):

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \|\Sigma_1^{1/2} - \Sigma_2^{1/2}\|_F^2 \quad (1)$$

In the given equation, $\|\cdot\|_F$ is the Frobenius norm, and m_n and $\sum n$ represent the mean and covariance matrices of the n^{th} 2D Gaussian distribution, respectively. Furthermore, when considering Gaussian distributions N_a and N_b modeled by the bounding boxes $A = (cx_a, cy_a, w_a, h_a)$ and $B = (cx_b, cy_b, w_b, h_b)$, the equation can be simplified, as shown in Equation (2).

$$W_2^2(N_a, N_b) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2} \right]^T, \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2} \right]^T \right) \right\|_2^2 \quad (2)$$

In the equation, (cx_a, cy_a), (cx_b, cy_b), (w_a, h_a) and (w_b, h_b) represent the center point coordinates, width, and height of bounding boxes A and B, respectively. $W_2^2(N_a, N_b)$ is a distance measure and cannot be used as a similarity measure. To solve this problem, the exponential form of the Wasserstein distance is normalized to derive a new metric called the Normalized Wasserstein Distance (NWD), as shown in Equation (3).

$$NWD(N_a, N_b) = exp\left(-\frac{\sqrt{W_2^2(N_a, N_b)}}{C} \right) \quad (3)$$

In Equation (3), C represents a constant closely related to the dataset, and is used to control the penalty when there is a poor overlap between the predicted and the actual bounding boxes. A higher value of C corresponds to a smaller penalty, while a lower value of C corresponds to a larger penalty. In this paper, the value of C is set to 3. Figures 6 and 7 show the comparison of IoU deviation curves [11] and NWD deviation curves in two different scenarios. The boxSize parameter represents the scale configuration of the bounding box. The horizontal axis represents the pixel deviation between the center points of the predicted box B and the actual box A. The vertical axis represents the corresponding IoU and NWD values for a given deviation. The curves of different colors represent the IoU Deviation or NWD Deviation curves for the respective scale settings. Since the position of the bounding box can only be changed discretely, the numerical deviation curves are presented as scatter plots. By examining the deviation curves, it is evident that NWD is more sensitive to changes in pixel deviation and better reflects the accuracy and stability of the target detection algorithm.

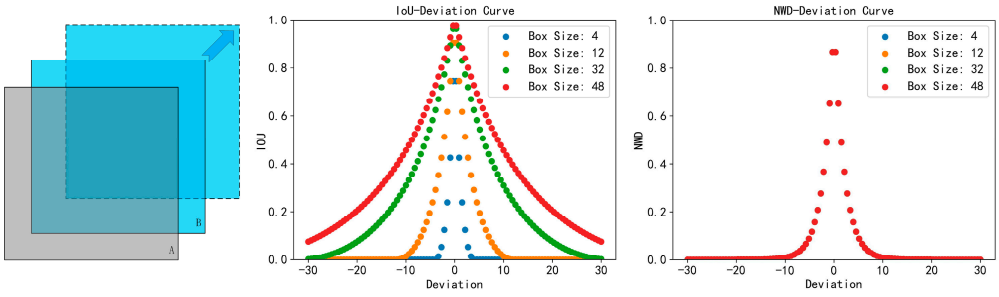

Figure 6. Deviation curve when the proportion of predicted box B equals the true A.

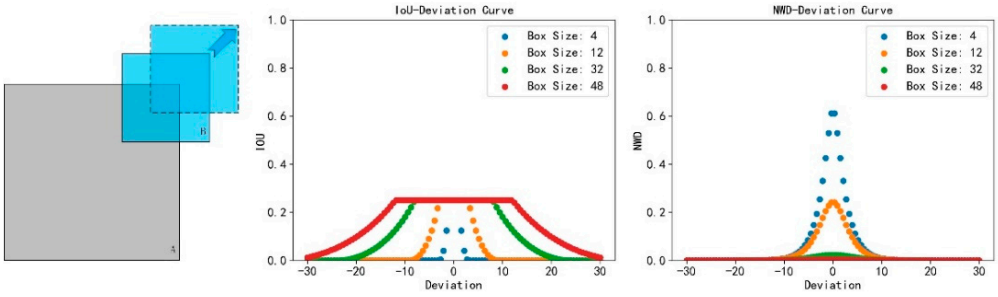

Figure 7. Deviation curve when the proportion of predicted box B is half of the true A.

Compared to the Intersection over Union (IoU), the Normalized Wasserstein Distance (NWD) has several obvious advantages:

1. IoU only quantifies the overlap and union of two bounding boxes and does not provide additional detailed information. On the contrary, NWD converts bounding boxes into probability distributions using the Gaussian distribution, enabling the measurement of the distance between these distributions. Due to the continuity of the Gaussian distribution, NWD provides enhanced granularity in identifying similarity between bounding boxes. Therefore, NWD provides richer information and a more accurate assessment of similarity.
2. IoU requires the bounding boxes to have the same scale and orientation, and its accuracy has decreased. On the contrary, NWD can evaluate the similarity between bounding boxes with differing scales and orientations as it adopts a Gaussian distribution that exhibits scale and direction invariance. Therefore, NWD excels in session scenarios involving mismatched directions and proportions in bounding boxes, which IoU cannot accomplish.
3. IoU tends to underestimate similarity when facing irregularly shaped objects. On the contrary, NWD evaluates similarity by converting bounding boxes into probability distributions, thus proficiently handling irregular shapes.

NWD emerges as a more accurate, continuous, and robust metric for measuring similarity between bounding boxes. It performs well in scenes with irregular object shapes and different scales, thus outperforming IoU.

(2) Convergence

Convergence refers to the behavior of a deep learning model during the training process, especially how the performance of the model improves or stabilizes during iterations or epochs. It is an important characteristic for assessing the learning ability and effectiveness of the model.

In deep learning, convergence is usually measured by monitoring the loss function or objective function. The loss function quantifies the difference between the model's predictions and the truth labels in the training data. The goal of training is to minimize this loss function and improve performance.

In the training process, gradient descent and other optimization algorithms are used to iteratively adjust the parameters of the model to minimize the loss function. As the training progresses, the performance of the model improves and the losses are reduced. Convergence occurs when the loss function reaches a sufficiently low or stable value, indicating that the model has learned to generalize well and make accurate predictions.

The convergence of a deep learning model can be evaluated by monitoring changes in training and validation loss. Different scenarios indicate different convergence states. If both training and validation losses are decreasing, it indicates that the network is still learning and improving its performance. On the contrary, if the training loss continues to decrease and the validation loss remains stable, the model may overfit the training data. When the training loss stabilizes and the validation loss decreases, it may indicate a potential problem with the dataset. If the training loss and validation loss are stable, it

indicates that the model has converged or encountered a learning bottleneck, and adjusting the learning rate could be attempted. On the other hand, an increase in training and validation loss indicates a problem with the network architecture or improper setting of training parameters, in which case the training should be immediately stopped for code adjustment. In Figure 8, the change curves of the loss for the model before and after adding the NWD loss function are shown. The blue curve represents the loss change before improvement, while the red curve represents the loss change after modifying the NWD loss function. These curves provide insights into the impact of the modification on the convergence behavior of the model.

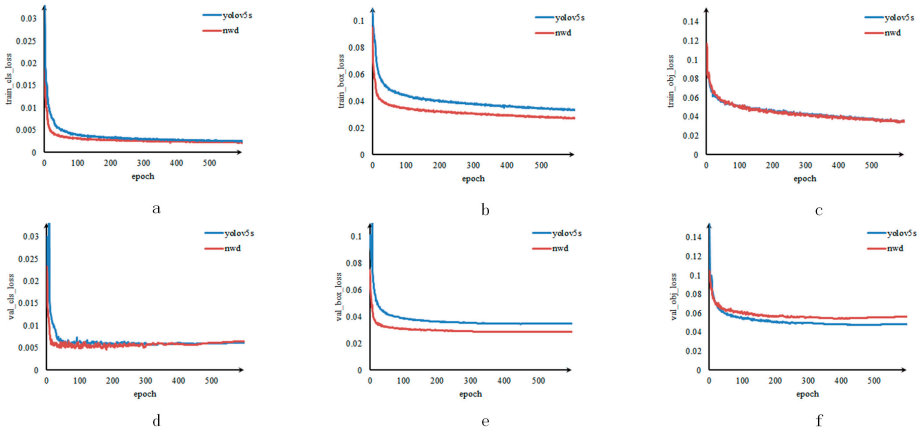

Figure 8. Loss change curve. (**a**) tran_cls_loss (**b**) train_box_loss (**c**) train_obj_loss (**d**) val_cls_loss (**e**) val_box_loss (**f**) val_obj_loss.

The cls_loss (Classification Loss) measures the accuracy of the model in classifying object categories, while the box_loss (Bounding Box Loss) measures the accuracy of the model in locating object bounding boxes. The obj_loss (Objectness Loss) measures the accuracy of the model in detecting the presence of objects.

According to Figure 8, it can be observed that train_loss and val_loss converge to stability, indicating that the model has converged. However, for cls_loss and box_loss, the NWD modification leads to faster convergence. In contrast, train_obj_loss shows similar behavior before and after the improvement, with overlapping curves. The val_obj_loss for YOLOv5s converges slightly faster. These observations suggest that the model, after improvement, learns the class information and precise bounding box locations of the target faster and achieves more accurate detection and localization. However, compared to before the improvement, the model has slightly lower prediction accuracy for the presence or absence of targets.

4.2.4. Introduction of the CSPCM model

(1) DFC attention

The Decoupled Fully Connected (DFC) [36] attention mechanism is a hardware-friendly technology specifically designed to capture the wide relationship between pixels while ensuring the computational efficiency of lightweight convolutional neural networks. Its design involves the utilization of fully connected layers that can be efficiently executed on standard hardware.

The DFC attention mechanism decomposes a fully connected layer into two different components: a horizontal fully connected layer and a vertical fully connected layer. These two layers are responsible for gathering the information about pixels in the 2D feature map from the convolutional neural network. By combining these separate, fully connected layers, the DFC attention mechanism effectively captures long-range dependencies along the horizontal and vertical directions, resulting in a comprehensive perception field.

A significant advantage of the DFC attention mechanism is its compatibility with hardware implementations, as it can efficiently execute on a common hardware platform. By using the computationally efficient full connection layer, the DFC attention mechanism achieves a balance between capturing long-range dependencies and maintaining the overall execution efficiency of lightweight convolutional neural networks.

For a given feature map $Z \in R^{H\times W\times C}$, it can be considered a set of HW tokens, denoted as $Z = z_{11}, z_{12}, \cdots, z_{HW}$. The direct approach to constructing the attention graph through a fully connected layer can be formulated by the following equation:

$$a_{hw} = \sum_{h',w'} F_{hw,h'w'} \odot z_{h'w'} \tag{4}$$

In Equation (4), the symbol $\odot$ denotes element-wise multiplication, F represents the learnable weights within the fully connected layer, H and W denote the height and width of the feature map, and C represents the number of channels in the feature map. The resulting attention map is represented by $A = a_{11}, a_{12}, \cdots, a_{HW}$. Equation (4) effectively captures global information by aggregating all the patches together using the learnable weights, which is simpler than traditional self-attention methods [37]. To further simplify Equation (4), it can be decomposed into two fully connected (FC) layers, where features are aggregated separately along the horizontal and vertical directions. This decomposition can be expressed by Formula as follows:

$$a'_{hw} = \sum_{h'=1}^{H} F^{H}_{h,h'w} \odot z_{h'w}; h = 1, 2, \cdots, H; w = 1, 2, \cdots, W \tag{5}$$

$$a_{hw} = \sum_{w'=1}^{W} F^{W}_{w,h'w} \odot a'_{h'w}; h = 1, 2, \cdots, H; w = 1, 2, \cdots, W \tag{6}$$

In Equation (5), F^H represents the transformation weight used to capture remote dependencies along the horizontal direction, while in Equation (6), F^W represents the transformation weight used to capture remote dependencies along the vertical direction. These transformation weights are applied to the original feature Z to capture long-range dependencies in each direction. By continuously applying Equations (5) and (6) to features, decoupled fully connected (DFC) attention operations are performed. The information flow of the DFC attention is illustrated in Figure 9. The DFC's attention module reduces the computational complexity to $O(H^2W + HW^2)$ by decoupling the horizontal and vertical transformations. In Equation (4), all patches within the square region directly participate in the calculation of the focus patch. However, in the DFC, a patch is directly aggregated by the patches on its vertical/horizontal line, while other patches are indirectly related to the focus patch through the generation of the patches on the vertical/horizontal line. Therefore, the calculation of patches also involves all patches in the square region.

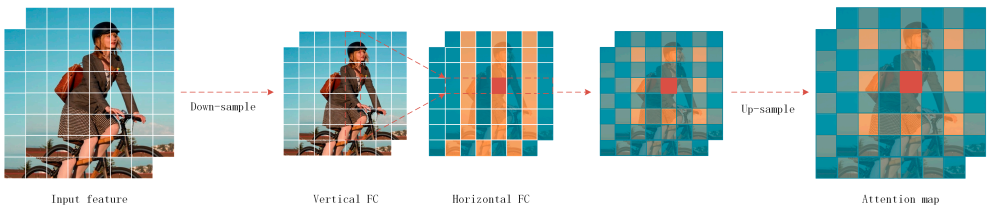

Figure 9. DFC attention.

Equations (5) and (6) represent the general form of DFC attention for aggregating pixels along the horizontal and vertical directions, respectively. By sharing some transformation weights, the DFC attention can be easily implemented using convolution operations, which eliminates the need for time-consuming tensor shaping and transposition operations that may affect the actual inference speed. To process input images of different resolutions,

the filter size can be decoupled from the size of the feature map. This means that deep convolutions with two kernel sizes, $1 \times K_H$ and $K_W \times 1$, are sequentially applied to the input features.

When using convolutional implementation, the theoretical complexity of DFC attention is expressed as $O(K_H HW + K_W HW)$. This strategy is well supported by tools such as TFLite and ONNX, enabling fast inference on mobile devices.

(2) ConvMix Module

In the previous presentation, the model's ability to detect small targets and computational efficiency were improved by introducing a small target detection layer and modifying the CARAFE upsampling method. Although both modifications have been proven effective, they also increase the model's complexity, resulting in higher computational requirements and lower inference speeds. Therefore, an improved Cross-Stage Partial Connection Module (CSPCM) based on ConvMix [38] is proposed to reduce computational complexity while maintaining module performance.

The ConvMix module, as part of the CSPCM module, plays a crucial role in feature extraction and input convolution operations. The module structure, shown in Figure 10, combines Resnet and Conv_1 $\times$ 1 for hybrid convolution operations, aiming to enhance the expressiveness and performance of the input features.

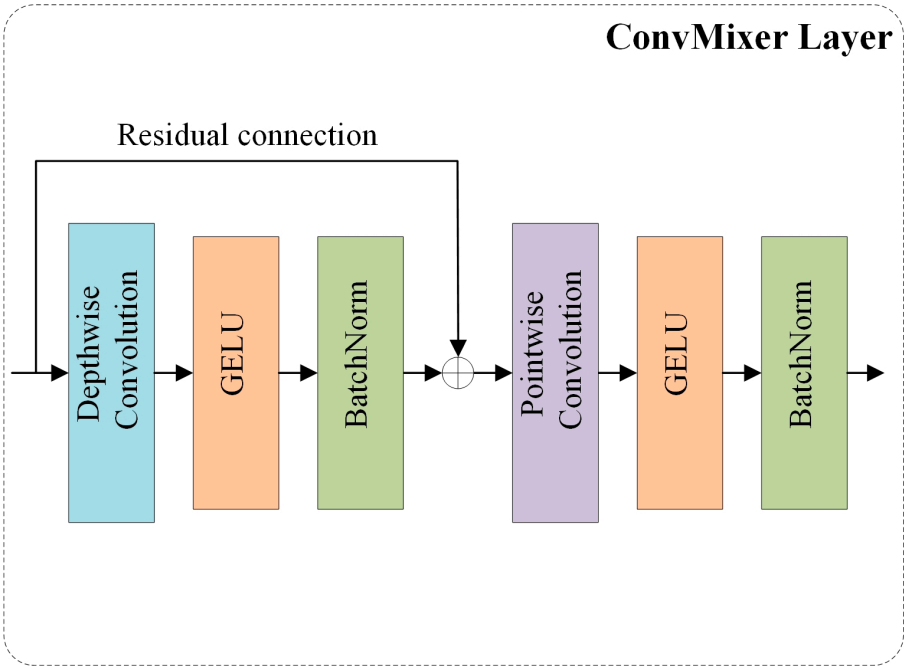

Figure 10. ConvMix module.

The Resnet module consists of a 2D convolution layer, a GELU activation function, and a batch normalization layer. It helps capture and extract meaningful features from the input.

The Conv_1 $\times$ 1 module includes a 1 $\times$ 1 2D convolution layer, a GELU activation function, and a batch normalization layer. It is designed for dimensionality reduction and further enhances the feature representation capabilities embedded in each patch.

By incorporating the ConvMix module after embedding each patch, the model can effectively improve the feature extraction process and enhance the representation ability of input features.

(3) CSPCM Module

The CSPCM module, short for Cross-Stage Partial Channel Mixing, is a commonly used module for enhancing feature representation in deep neural networks. Its main

purpose is to improve the performance of the network while reducing the number of parameters and computational requirements. The structure of the CSPCM module is illustrated in Figure 11a.

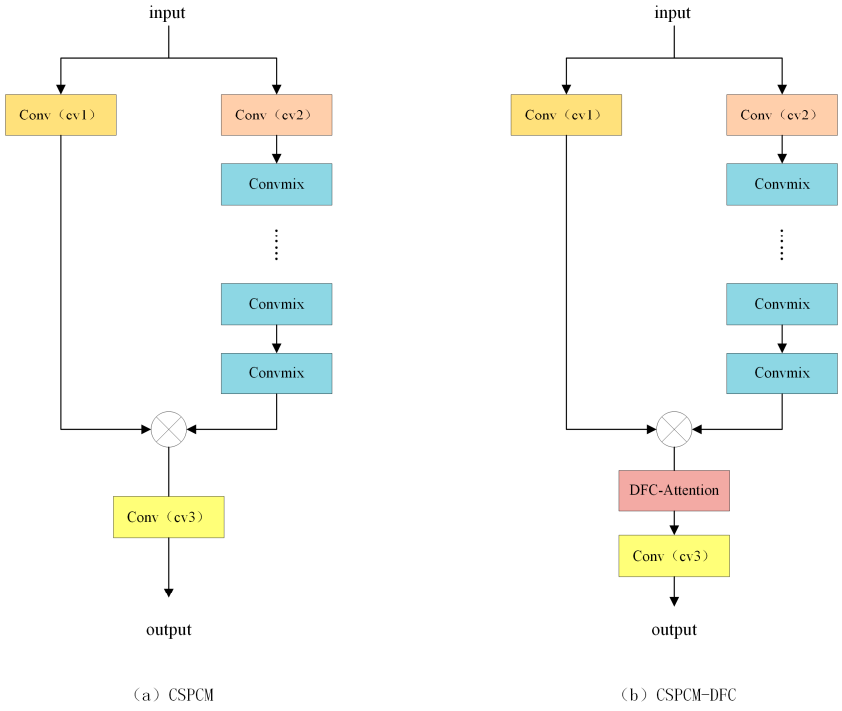

Figure 11. CSPCM module.

The CSPCM module operates by dividing the input feature map into two parts. Part of it undergoes convolution operations and is enhanced through multiple ConvMix modules. The enhanced features are then merged with the other parts, and then a 1×1 convolution is applied to perform feature fusion. The ConvMix module consists of two convolutional layers and includes a residual connection, which helps improve the network's ability to represent nonlinear features. In addition, the CSPCM module introduces a dilation factor (e) to control the module's width, which refers to the number of channels in the intermediate hidden layer. By adjusting this parameter, a balance can be achieved between module performance and computational complexity.

(4) CSPCM-DFC

In a previous discussion, DFC was introduced, highlighting its ability to effectively capture a broader range of pixel relationships while maintaining computational efficiency. In addition, DFC is designed to run efficiently on standard hardware without requiring more complex operations. These characteristics make DFC attention very suitable for resource-constrained environments, such as lightweight convolutional neural networks and mobile devices.

In this section, the DFC and CSPCM modules are combined to enhance the expressiveness of the lightweight model and capture long-range dependencies between spatial pixels. The structure diagram of the combined module, referred to as CSPCM-DFC, is shown in Figure 11b.

The CSPCM module in CSPCM-DFC includes the DFC attention mechanism, similar to its combination with GhostNetV2 [36]. The input features are first divided into two branches, each of which undergoes different convolution operations to obtain different feature representations. The ConvMix module is similar to the Ghost module in GhostNetV2, used for feature fusion. It extracts different perspectives of the features to further enhance the feature representations stored in variable $\times 1$. The output of the other branch is then

merged with ×1 to generate a feature map with double the channel count (2 * c_), where c_ represents the number of channels in the middle hidden layer of the CSPCM module.

The generated feature map with enhanced representations is then passed through the DFCAttention module. This module utilizes the DFC attention mechanism to capture long-range dependencies between spatial pixels by assigning weights to the features. Finally, the DFCAttention module processes the features through another convolution operation to obtain the final feature representation.

To validate the practical effects of Convmix, CSPCM, and DFCAttention, a series of experiments were conducted, and the corresponding experimental results are presented in Table 2.

Table 2. Convmix, CSPCM, and DFC experimental results.

Algorithm	P	R	mAP.5	mAP.95	Parameters
YOLOv5s	95	87.2	91.9	68.9	7,018,216
Convmix	93.9	86.5	91.7	67.9	5,334,760
CSPCM	94.9	86.8	91.7	68.3	5,949,672
CSPCM-DFC	93.8	88.4	92.1	69.1	5,927,114

From the experimental results, it can be observed that combining Convmix significantly reduces the number of model parameters. However, the reduction of this parameter is accompanied by a decrease in accuracy. On the other hand, when using CSPCM, the number of parameters increases, but there is a notable improvement in P as well as other performance indices. Furthermore, when DFC is added to CSPCM, there is a slight decrease in the number of parameters while the accuracy is improved.

The ConvMix module's feature fusion and the DFCAttention module's attention mechanism contribute to enhancing the model's representation. The ConvMix module provides feature representations from multiple perspectives, enriching the model's understanding of images. The DFCAttention module, with its DFC attention mechanism, limits the long-range dependence between spatial pixels, enabling the model to better comprehend global information within an image. By combining DFC, the performance of the CSPCM module is improved, resulting in better performance and feature capture for lightweight models.

5. Experiment

5.1. Experimental Setup

The experiment was conducted using the deep learning environment and framework mentioned in Table 3. The environment and framework were applied to the Faster RCNN, unimproved YOLOv5, YOLOx [39], and YOLOv3 networks with the same configurations.

Table 3. Experimental environment configuration table.

Configuration Name	Configuration Parameters
Operating System	Ubuntu 18.04.6 LTS
CPU	Intel Xeon(R) Silver 4210 CPU @ 2.20 GHz × 40
GPU	NVIDIA GeForce RTX 2080 Ti/PCIe/SSE2
Memory	125.6 GiB
Software	Anaconda3-5.3., Pycharm2021
Deep Learning Framework	Pytorch 1.13.1
GPU Acceleration Library	CUDA 11.6

5.2. Evaluation Index and Ablation Experiment

In this paper, the performance of the improved YOLOv5 target detection model is evaluated using several indicators, including precision, recall, mAP.5, mAP.95, and

parameters. These indicators provide insights into the model's effectiveness in detecting targets accurately.

Precision measures the ratio of correctly identified positive samples to all identified positive samples. Recall represents the ratio of correctly identified positive samples to all actual positive samples.

The formulas for these indicators are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$AP = \int_0^1 p(r)dr \tag{9}$$

$$map.5 = \frac{1}{N}\sum_{i=1}^{N} AP_i^{IoU=0.5} \tag{10}$$

$$map.95 = \frac{1}{N}\sum_{i=1}^{N} AP_i^{IoU=0.95} \tag{11}$$

In the equations, TP denotes the number of correctly classified positive samples, FP denotes the number of incorrectly classified positive samples, FN denotes the number of incorrectly classified negative samples, and N denotes the total number of categories. The AP (Average Precision) is calculated separately for each category and then averaged over all categories. $p(r)$ denotes the AP under a given recall rate r and the AP of the ith category under IoU thresholds of 0.5 and 0.95.

A series of ablation experiments were conducted on the validation and test sets to investigate the effects of different conditions on the model's performance. The results of these experiments are presented in Table 4, where H represents the small target detection layer, CD represents the CSPCM-DFC, CA represents the CARAFE upsampling module, and NWD represents the NWD loss function.

Table 4. Ablation experiments.

Algorithm	H	CD	CA	NWD	Val				Test				Parameters
					P	R	mAP.5	mAP.95	P	R	mAP.5	mAP.95	
YOLOv5s					95	87.2	91.9	68.9	94.2	86.3	91	68.7	7,018,216
A	√				96.8	93.3	96.4	75.4	96	93.2	96.2	75.7	7,185,888
B		√			93.8	88.4	92.1	69.1	93.2	87.6	91.7	68.9	5,927,114
C			√		92.2	89	92.5	69.3	91.7	88.3	91.8	69	7,152,272
D				√	94.7	89.4	93.2	70.7	93.5	89.4	92.6	70.7	7,018,216
E	√	√			96.1	93.3	96.1	75.6	95.4	93.3	96.1	75.5	6,094,816
F	√	√	√		94.9	93.8	96.6	76.3	94	93.4	95.4	76.3	6,239,292
Last	√	√	√	√	96	95	96.3	77.5	95.3	94.7	96.2	77.3	6,239,292

Experiment A involved adding a small target detection layer to YOLOv5s. The experimental results showed significant improvements in indicators such as P, R, mAP.5, and mAP.95, but this occurred as the number of parameters increased.

Experiment B replaced layers 8 and 23 of the original model with the CSPCM-DFC module. The results showed that although mAP.5 and mAP.95 did not improve significantly, the number of parameters decreased by 1,091,102, reducing the complexity of the model without affecting performance.

Experiment C utilized the CARAFE module as a replacement for the original Nearest Upsample module. The results indicated an increase of 0.8 in mAP.5, 0.3 in mAP.95, and a slight increase in the number of parameters, indicating an improvement in the performance of the model.

Experiment D introduced the NWD loss function to replace the original CIOU loss function. After improvement, P slightly decreased, R increased by 2, mAP.5 increased by 1.6, and mAP.95 increased by 2, indicating that the improved model captures targets more comprehensively, although the possibility of misclassification slightly increased.

Experiments E, F, and Last involved sequentially adding the CSPCM-DFC, CARAFE, and NWD components to Experiment A, respectively. The results showed a significant decrease in the number of model parameters after adding the CSPCM-DFC module, while the model's performance remained basically unchanged. The addition of CARAFE resulted in a decrease in mAP.5 and an increase in mAP.95, indicating improvements in target localization and high-confidence prediction. Finally, the addition of NWD resulted in an increase in mAP.5 and mAP.95, indicating an improvement in the detection performance of the model.

In summary, the ablation experiments demonstrated that the performance of EV helmet recognition methods can be significantly improved by combining different components. The final experiments achieved significant improvements in precision, recall, mAP.5, and mAP.95 indicators. These findings provide strong support for our study and offer valuable insights for further development and applications of the EV helmet recognition method.

5.3. Comparative Experiments

To further validate the effectiveness of the algorithm, experiments were conducted using the same dataset, equipment, and training strategies under the same conditions. Mean Average Precision at IoU 0.5 (mAP.5), model size, and Frames Per Second (FPS) were used as evaluation indicators. These indicators were compared with the Faster-RCNN, YOLOv3, and YOLOx algorithms, and the results are presented in Table 5.

Table 5. Comparison experiments.

Algorithm	Val_mAP.5/%				Test_mAP.5/%				FPS	Volume/mb
	E-Bike	With	Without	All	E-Bike	With	Without	All		
Faster-RCNN	94.731	12.82	14.37	40.62	94.49	12.92	13.48	40.29	15.257	113.5
YOLOv3	85.7	81	73.9	80.2	85.4	80.7	73.7	79.9	29.4	123.5
YOLOx	90.85	90.58	89.12	90.18	90.91	90.86	90.16	90.64	32.072	71.8
Last	99.1	97.5	92.2	96.3	99	97.6	92.1	96.2	35.707	13.8

Based on the comparative experimental results in the table, the performance of the improved electric bicycle helmet recognition method based on the YOLOv5s algorithm compared to other algorithms was analyzed and discussed. The following is an analysis and discussion of the results:

Firstly, the proposed method was compared with Faster-RCNN, YOLOv3, and YOLOx algorithms in terms of target categories (electric bicycle, helmeted, and non-helmeted) and overall mAP.5.

From Table 5, it can be observed that Faster-RCNN obtained mAP.5 values of 94.49, 12.92, and 13.29 in the categories E-bike, helmeted, and non-helmeted, respectively, with a total mAP.5 of 40.29. YOLOv3 obtained mAP.5 values of 85.4, 80.7, and 73.7, with a total mAP.5 of 79.9. YOLOx obtained mAP.5 values of 90.91, 90.86, and 90.16, with a total mAP.5 of 90.64. In comparison, the method proposed in this paper performed very well in these categories, generating values of 99, 97.6, and 92.1, with a total mAP.5 of 96.2. These results highlight the significant advantages of the proposed algorithm for electric bicycle helmet recognition.

Furthermore, each algorithm was evaluated in terms of Frames Per Second (FPS) and model size (storage space). The proposed method achieved an FPS of 35.707, significantly surpassing Faster-RCNN, YOLOv3, and YOLOx. Therefore, the proposed method demonstrates significant advantages in processing speed.

In terms of model size, the proposed method only occupies 13.8 MB, while Faster-RCNN, YOLOv3, and YOLOx occupy 113.5 MB, 123.5 MB, and 71.8 MB of capacity, respectively. This result indicates that the proposed method has lower storage space requirements.

In conclusion, the YOLOv5s-based electric bicycle helmet recognition method outperforms Faster-RCNN, YOLOv3, and YOLOx algorithms in terms of target categories and overall mAP.5. In addition, the proposed algorithm also demonstrates advantages in processing speed and storage space. These results highlight the enormous potential of our method in the field of electric bicycle helmet recognition, providing convincing evidence for its practical application.

5.4. Comparison of the Detection Effect

To evaluate the model's performance, three types of images are used as test images. This includes video clip frames captured from the simulated surveillance viewpoint of the UAV (Figure 12a,b), video clip frames of road surveillance video in the test set (Figure 12c), and network images (Figure 12d). These images are utilized to compare the actual results of the original YOLOv5s algorithm with the method proposed in this paper.

Figure 12. Original image. (**a**,**b**) Video intercepts taken by a drone simulating a surveillance viewpoint (**c**) Road surveillance video intercepts from the test set (**d**) Network images.

According to the detection results shown in Figures 12–14, it can be seen that the YOLOv5s algorithm and the improved algorithm in this paper exhibit different performance in detecting hats, helmets, and long distance ultra-small targets. In the detection of the b-map, the YOLOv5s algorithm detects helmets with difficult, while the improved algorithm in this paper has achieved greater success in detecting this target. In almost all cases, compared with the YOLOv5s algorithm, the improved algorithm shows a higher confidence level in identifying the helmet frame of an e-bike. This observation indicates that the improved algorithm has significant advantages in identifying small targets such as e-bike helmets. However, both the improved algorithm and the original YOLOv5s algorithm have instances of misidentifying traffic lights as electric bikes. These findings can provide valuable insights for future algorithm improvements.

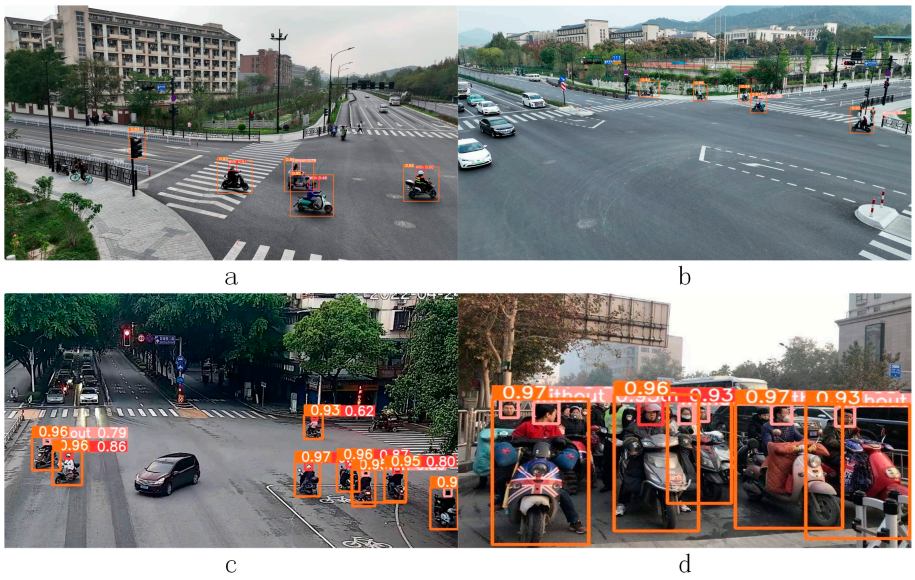

Figure 13. YOLOv5s detection results. (**a**,**b**) Video intercepts taken by a drone simulating a surveillance viewpoint (**c**) Road surveillance video intercepts from the test set (**d**) Network images.

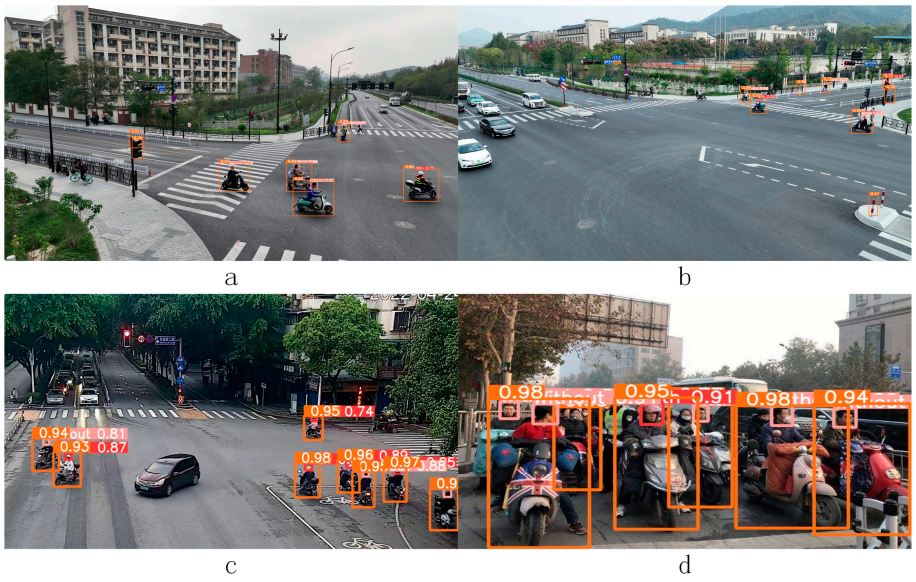

Figure 14. Improved YOLOv5s detection results. (**a**,**b**) Video intercepts taken by a drone simulating a surveillance viewpoint (**c**) Road surveillance video intercepts from the test set (**d**) Network images.

6. Conclusions

In order to address the challenges faced by existing target detection algorithms in electric bike helmet detection, an improved YOLOv5s model is proposed in this paper. Compared to the original YOLOv5s, the improved algorithm includes several enhanced features. These include the addition of an extra-small target layer, the introduction of the NWD loss function, the design of the CSPCM module based on Convmix, the replacement of Nearest Upsample with CARAFE, and the integration of the DFC attention mechanism. These modifications collectively aim to reduce model size while simultaneously improving model accuracy. Experimental results demonstrate that the proposed model achieves significant improvements in accuracy, recall, mAP.5, and mAP.95 on customized datasets. Furthermore, compared to the original YOLOv5s, the number of parameters and model size have been reduced.

This paper presents the architecture and key technologies of the improved model and provides a comparative analysis of experimental results with the original YOLOv5s. The experimental results validate the effectiveness of the proposed algorithm. The significance of this paper is to provide an effective improvement scheme that can have an impact on other target detection algorithms, especially in small target detection scenarios.

Future research can focus on exploring the application of our models in different datasets and scenarios, optimizing algorithm performance and efficiency, and concurrently delving deeper into the realm of reinforcement learning. By combining image recognition with reinforcement learning, we hope to develop more intelligent and adaptive systems.

Author Contributions: Software, S.W. and H.H.; Investigation, Y.W.; Resources, X.X.; Writing—original draft, S.W.; Writing—review & editing, B.H.; Supervision, X.X.; Project administration, B.H.; Funding acquisition, Z.F. and S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Open Foundation of the Key Laboratory of Intelligent Robot for Operation and Maintenance of Zhejiang Province (SZKF-2022-R06), Open Foundation of the Key Laboratory of Intelligent Robot for Operation and Maintenance of Zhejiang Province (SZKF-2022-R04), Zhejiang University of Science and Technology 2022 postgraduate research innovation fund projects (2022yjskc06), Zhejiang Provincial Natural Science Foundation (LY19F030004), Zhejiang Provincial Department of Transportation Science and Technology Plan Project (202206).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

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

References

1. Zhang, Q.; Zhang, H.; Lu, X. Adaptive Feature Fusion for Small Object Detection. *Appl. Sci.* **2022**, *12*, 11854. [CrossRef]
2. Cui, M.; Gong, G.; Chen, G.; Wang, H.; Jin, M.; Mao, W.; Lu, H. LC-YOLO: A Lightweight Model with Efficient Utilization of Limited Detail Features for Small Object Detection. *Appl. Sci.* **2023**, *13*, 3174. [CrossRef]
3. Mirri, S.; Delnevo, G.; Roccetti, M. Is a COVID-19 second wave possible in Emilia-Romagna (Italy)? Forecasting a future outbreak with particulate pollution and machine learning. *Computation* **2020**, *8*, 74. [CrossRef]
4. Fan, D.P.; Ji, G.P.; Cheng, M.M.; Shao, L. Concealed object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 6024–6042. [CrossRef]
5. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: Pitcataway, NJ, USA, 2005; Volume 1, pp. 886–893.
6. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 1999; IEEE: Pitcataway, NJ, USA, 1999; Volume 2, pp. 246–252.
7. Chen, L.; Dai, S.-L.; Dong, C. Adaptive Optimal Tracking Control of an Underactuated Surface Vessel Using Actor–Critic Reinforcement Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, 1–14. [CrossRef]
8. Pham, T.L.; Dao, P.N. Disturbance observer-based adaptive reinforcement learning for perturbed uncertain surface vessels. *ISA Trans.* **2022**, *130*, 277–292.
9. Dao, P.N.; Liu, Y.C. Adaptive reinforcement learning in control design for cooperating manipulator systems. *Asian J. Control.* **2022**, *24*, 1088–1103. [CrossRef]
10. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Washington, DC, USA, 7–13 December 2015; pp. 1440–1448.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, 28. [CrossRef]
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
13. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
14. Zheng, N.; Loizou, G.; Jiang, X.; Lan, X.; Li, X. Computer vision and pattern recognition. *Int. J. Comput. Math.* **2007**, *84*, 1265–1266. [CrossRef]

15. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
16. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
17. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
18. Lin, H.; Deng, J.D.; Albers, D.; Siebert, F.W. Helmet use detection of tracked motorcycles using cnn-based multi-task learning. *IEEE Access* **2020**, *8*, 162073–162084. [CrossRef]
19. Hayat, A.; Morgado-Dias, F. Deep learning-based automatic safety helmet detection system for construction safety. *Appl. Sci.* **2022**, *12*, 8268. [CrossRef]
20. Li, Q.; Peng, F.; Ru, Z.; Yu, S.; Zhao, Q.; Shang, Q.; Cao, Y.; Liu, J. Research on safety helmet detection method based on convolutional neural network. In *Sixth Symposium on Novel Optoelectronic Detection Technology and Applications*; SPIE: Bellingham, WA, USA, 2020; Volume 11455, pp. 1115–1121.
21. Wang, Y. Research on a Safety Helmet Detection Method Based on Smart Construction Site. In Proceedings of the 2021 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA), Dalian, China, 27–28 August 2021; IEEE: Pitcataway, NJ, USA, 2021; pp. 341–343.
22. Sun, X.; Xu, K.; Wang, S.; Wu, C.; Zhang, W.; Wu, H. Detection and tracking of safety helmet in factory environment. *Meas. Sci. Technol.* **2021**, *32*, 105406. [CrossRef]
23. Yan, G.; Sun, Q.; Huang, J.; Chen, Y. Helmet detection based on deep learning and random forest on UAV for power construction safety. *J. Adv. Comput. Intell. Intell. Inform.* **2021**, *25*, 40–49. [CrossRef]
24. Jia, W.; Xu, S.; Liang, Z.; Zhao, Y.; Min, H.; Li, S.; Yu, Y. Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector. *IET Image Process.* **2021**, *15*, 3623–3637. [CrossRef]
25. Li, J.; Zuo, Y.; Li, Y.; Wang, Y.; Li, T.; Chen, C.P. Application of genetic algorithm for broad learning system optimization. In Proceedings of the 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS), Guangzhou, China, 13–15 November 2020; IEEE: Pitcataway, NJ, USA, 2020; pp. 783–788.
26. Cheng, R.; He, X.; Zheng, Z.; Wang, Z. Multi-scale safety helmet detection based on SAS-YOLOv3-tiny. *Appl. Sci.* **2021**, *11*, 3652. [CrossRef]
27. Shine, L.; Jiji, C.V. Automated detection of helmet on motorcyclists from traffic surveillance videos: A comparative analysis using hand-crafted features and CNN. *Multimed. Tools Appl.* **2020**, *79*, 14179–14199. [CrossRef]
28. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18 June 2023; pp. 7464–7475.
29. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
30. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8759–8768.
31. Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
32. Wang, J.; Chen, K.; Xu, R.; Liu, Z.; Loy, C.C.; Lin, D. Carafe: Content-aware reassembly of features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3007–3016.
33. Wang, J.; Xu, C.; Yang, W.; Yu, L. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.
34. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
35. Shi, W.; Caballero, J.; Huszár, F.; Totz, J.; Aitken, A.P.; Bishop, R.; Rueckert, D.; Wang, Z. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1874–1883.
36. Tang, Y.; Han, K.; Guo, J.; Xu, C.; Xu, C.; Wang, Y. GhostNetv2: Enhance cheap operation with long-range attention. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9969–9982.
37. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, 30.
38. Ng, D.; Chen, Y.; Tian, B.; Fu, Q.; Chng, E.S. Convmixer: Feature interactive convolution with curriculum learning for small footprint and noisy far-field keyword spotting. In Proceedings of the ICASSP 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; IEEE: Pitcataway, NJ, USA, 2022; pp. 3603–3607.
39. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.