electronics

MDPI

Article

YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera

Qiuli Liu , Haixiong Ye *, Shiming Wang and Zhe Xu

School of Engineering, Shanghai Ocean University, Shanghai 201306, China; liuqiuli970@gmail.com (Q.L.); smwang@shou.edu.cn (S.W.); xuzhe@shou.edu.cn (Z.X.)
* Correspondence: hxye@shou.edu.cn

Abstract: Recently, the field of vehicle-mounted visual intelligence technology has witnessed a surge of interest in pedestrian detection. Existing algorithms for dense pedestrian detection at intersections face challenges such as high computational weight, complex models that are difficult to deploy, and suboptimal detection accuracy for small targets and highly occluded pedestrians. To address these issues, this paper proposes an improved lightweight multi-scale pedestrian detection algorithm, YOLOv8-CB. The algorithm introduces a lightweight cascade fusion network, CFNet (cascade fusion network), and a CBAM attention module to improve the characterization of multi-scale feature semantics and location information, and it superimposes a bidirectional weighted feature fusion path BIFPN structure to fuse more effective features and improve pedestrian detection performance. It is experimentally verified that compared with the YOLOv8n algorithm, the accuracy of the improved model is increased by 2.4%, the number of model parameters is reduced by 6.45%, and the computational load is reduced by 6.74%. The inference time for a single image is 10.8 ms. The cascade fusion algorithm YOLOv8-CB has higher detection accuracy and is a lighter model for multi-scale pedestrian detection in complex scenes such as streets or intersections. This proposed algorithm presents a valuable approach for device-side pedestrian detection with limited computational resources.

Keywords: improved YOLOv8n; multi-scale feature fusion; attention mechanism; pedestrian detection

check for updates

Citation: Liu, Q.; Ye, H.; Wang, S.; Xu, Z. YOLOv8-CB: Dense Pedestrian Detection Algorithm Based on In-Vehicle Camera. *Electronics* **2024**, *13*, 236. https://doi.org/10.3390/electronics13010236

Academic Editor: Ping-Feng Pai

Received: 23 November 2023
Revised: 30 December 2023
Accepted: 2 January 2024
Published: 4 January 2024

1. Introduction

In recent years, pedestrian detection has emerged as a pivotal component in a variety of applications [1–23], including driver assistance systems, vehicle surveillance, and proactive safety mechanisms. This has established it as a fundamental and imperative area of study within the realm of object detection. Coinciding with the brisk advancement of the AI sector and the augmentation of computational power in computer hardware, pedestrian detection methodologies grounded in computer vision have seen extensive implementation in the vehicular camera systems of autonomous vehicles [4,5]. These innovative technologies facilitate the precise localization of pedestrians, substantially mitigating accident risks. Nonetheless, in spite of these technological strides, current pedestrian detection systems encounter substantial challenges in maintaining both high accuracy and rapid processing speeds. This is particularly evident in complex environments characterized by dense pedestrian populations, obstructive elements, and constricted urban intersections.

In the rapid development of visual intelligence technology, a large number of scholars have dabbled in pedestrian detection research using deep learning and achieved impressive results [6]. Notably, deep learning methods using convolutional neural networks (CNNs) have played a key role in addressing the complex challenges posed by multi-scale pedestrian detection [7–12]. Currently, two main approaches to deep learning for object detection are of interest. The first approach, represented by Girshick's R-CNN [13], employs a two-stage principle that uses region suggestion to generate candidate regions for subsequent classification and regression tasks. Building on this example, Ren and colleagues enhanced this paradigm by introducing the Faster R-CNN [14], which integrates an intrinsic deep network for candidate region replacement, thereby improving detection accuracy. Despite their effectiveness, these methods have limitations in terms of processing speed and their inherent complexity prevents effective deployment on mobile platforms. In contrast, single-stage methods, represented by Redmon's YOLO family [15], provide comprehensive end-to-end detection capabilities, proficiently regressing images to classify object classes and spatial coordinates. The SSD algorithm [16], pioneered by Liu et al. and referred to as a "single-shot, multi-box detector", provides excellent fast detection capabilities but can be limited in recognizing smaller objects. Currently, the YOLO family is the most complete and commonly used device-side solution due to its superior accuracy and practical deployability. However, in the realm of complex detection tasks, end-to-end approaches may require compromises in detection accuracy in order to increase speed. In order to reconcile the tension between detection speed and accuracy, researchers have started to further refine the models based on the YOLO family [17–20].

At urban traffic intersections, pedestrians often have different scale characteristics, which poses a challenge to the detection system. To address this problem, several researchers have proposed methods for balancing scales in multi-scale pedestrian detection. Zhang and colleagues [21] combined a deep residual contraction network with an attentional mechanism to significantly enhance YOLOv5s, thereby enriching the feature channel with valuable information. This approach captures a large number of multi-scale pedestrian features by extending the spatial pyramid pooling module, which significantly improves the detection accuracy, especially in underground pedestrian detection scenarios. Ding et al. [22], on the other hand, developed the Cascaded Cross-Layer Fusion Network (CCFNet). The CCF modules in the backbone of this network can collaborate features from different layers to facilitate more detailed semantic interpretation. It is further complemented by a global smoothing map in the detection header, which aims to assimilate global feature data. In addition, Tan et al. [23] introduced the concept of the Weighted Bidirectional Feature Pyramid Network (BiFPN). This innovative network cleverly fuses multi-scale neural network features by composite scaling of feature maps of different resolutions, thus simplifying the fusion process and improving its effectiveness and efficiency.

At urban traffic intersections and similar densely populated areas, the problem of overlapping pedestrians and heavy occlusion often leads to lost tracking targets and reduced detection rates. To address this challenge, Lv and his team [24] developed an innovative multi-branch, anchorless framework network with a special focus on the differential learning of pedestrians' local positions. This approach greatly improves the detection rate of hidden pedestrians in crowded environments. Based on Retina Net, Zhou et al. [25] introduced an occlusion-aware pedestrian detection algorithm. The algorithm cleverly combines a dual attention mechanism by integrating spatial and channel attention sub-networks into the regression and classification branches, which significantly improves the performance under severe occlusion conditions. However, it is worth noting that the introduction of these sub-networks increases the computational load and thus affects the frame rate.

In scenarios where pedestrian targets are small in size, such as roads, traditional methods have a high false detection rate for these small targets. To address this problem, Gu et al. [26] made significant improvements to the YOLOv5 algorithm. They improved the feature pyramid network structure (i.e., IM-FPN) to facilitate multi-scale feature fusion for dense objects and synergized it with a layer dedicated to detecting small targets, effectively reducing the missed detection rate of small outdoor targets. Lou et al. [27] proposed innovative adaptations to the YOLOv8 backbone network, advocating the use of cascaded depth-separable convolutions instead of the traditional C2F feature fusion module. This adaptation helps to retain a wider range of multi-scale pedestrian information, thus improving the network's ability to detect small pedestrian targets in sensor devices. To further enrich the feature fusion process of YOLOv5, Niu and his team also integrated bilinear interpolation up-sampling and five CBAM attention mechanisms [28], which play an important role in significantly reducing the miss detection rate of small road targets.

Despite the continuous advances in pedestrian detection techniques, contemporary algorithms mainly rely on large amounts of computational resources, especially graphics card acceleration, to achieve optimal accuracy and processing speed. These algorithms usually emphasize multi-scale feature fusion, focusing on easily distinguishable global features, but tend to ignore intricate local details, which affects the detection of pedestrians in dense or small-target scenes. In order to overcome these obstacles and strike a balance between detection accuracy and speed, especially in complex environments such as street intersections, this paper introduces a new lightweight dense pedestrian detection algorithm based on YOLOv8n, namely YOLOv8-CB.

The YOLOv8-CB algorithm employs a lightweight cascaded feature fusion network (CFNet), which replaces the traditional C2F block in the backbone network with an advanced Focal-NeXtF Block. This innovation not only simplifies the complexity of the model, but also improves the efficiency of extracting and merging multi-scale features. As the core of the detection header, the algorithm integrates a four-layer CBAM channel space focusing mechanism, which enables the model to pay more attention to the relevant feature channels and enhances the feature representation associated with small target pedestrians. In addition, the algorithm integrates a bidirectional weighted feature fusion structure (BIFPN) in its feature fusion component, which is capable of weighting and fusing multi-scale features extracted from the backbone network, thereby substantially improving the detection of pedestrians obscured by dense occlusions. Empirical tests confirm that CB-YOLOv8n excels in extracting multi-scale features accurately and comprehensively, outperforming other algorithms in terms of efficiency and presenting a more streamlined model. This makes it particularly suitable for dense pedestrian detection applications in urban street and intersection scenarios. The following section describes in detail the fundamentals and innovations of the YOLOv8-CB algorithm.

2. Materials and Methods

2.1. YOLOv8 Algorithm

The YOLOv8 network architecture consists of four main components: the input, the backbone network, feature enhancement (Neck), and the decoupling head (Head). On the input side, key enhancements include Mosaic data augmentation, adaptive anchor frame computation, and adaptive grayscale filling. The backbone network of YOLOv8 departs from the conventional C3 module, opting for the CSP (Cross Stage Partial) concept with the lightweight CSPLayer_2Conv module. The backbone network concludes with the widely adopted SPPF (Spatial Pyramid Pooling with Factorized convolutions) module, contributing to its robust feature extraction capabilities. In the feature enhancement section, a bidirectional pathway known as PAN-FPN (Path Aggregation Network–Feature Pyramid Network) is employed [29,30]. This feature pyramid network integrates three down-sampled inputs through channel fusion via up-sampling. Ultimately, the output is fed into three branches, directing the flow towards the decoupling head. The decoupling head is responsible for segregating the regression and prediction branches. In the regression branch, loss calculation involves both category and localization components. For category loss, VFL Loss (Varifocal Loss) is adopted, utilizing the BCE (Binary Cross Entropy) loss function. Localization loss includes the DFL (Distribution Focal Loss) and CIOU (Complete IOU) loss components. The overall YOLOv8 network architecture is visually depicted in Figure 1.

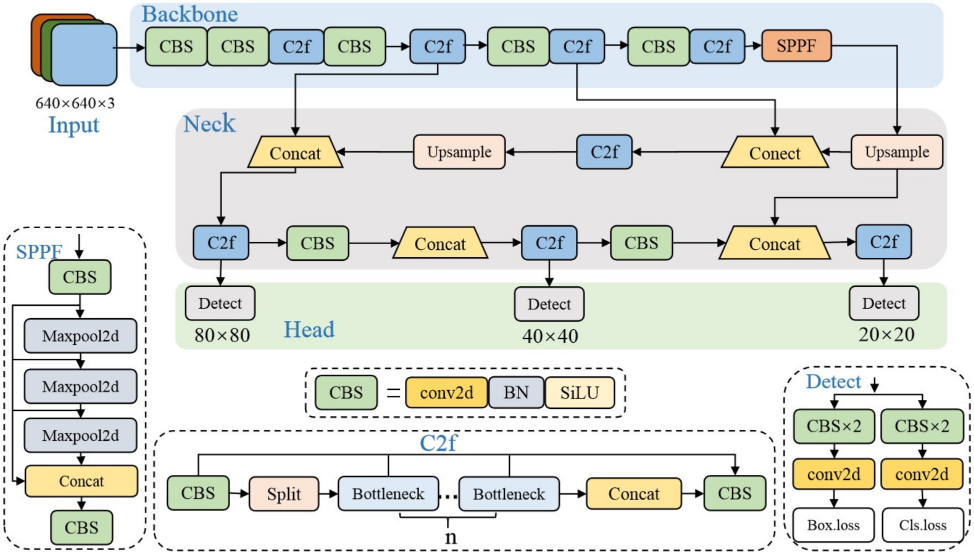


Figure 1. The network structure of YOLOv8.

DFL aims to model the target detection frame's position as a global distribution using cross-entropy. This optimization approach is employed to enhance the probability of the position being close to the label. Consequently, this enables the network to swiftly concentrate on the target position, as shown in Equation (1). S_i and S_{i+1} represent the output of the sigmoid function for the network, y_i and y_{i+1} denote interval orders, and y is a label.

$$DFL(\mathcal{S}_i, \mathcal{S}_{i+1}) = -((y_{i+1} - y)\log(\mathcal{S}_i) + (y - y_i)\log(\mathcal{S}_{i+1})) \tag{1}$$

CIOU quantifies the distance between the actual frame and the predicted frame, while DIOU measures the Euclidean distance between the centers of the two detection frames. CIOU builds upon DIOU by incorporating considerations for the aspect ratio of the detection frames, as shown in Equations (2)–(4). α represents the weight function; v is used to measure the similarity of the aspect ratio; IOU is the intersection ratio between the real frame and the predicted frame; ρ is the Euclidean distance between the centers of the predicted frame and the real frame; b and b^{gt} represent the centers of the predicted frame and the real frame; c is the diagonal distance of the smallest enclosing region that can contain both the predicted frame and the real frame.

$$CIOU = \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \tag{2}$$

$$\mathrm{v} = \frac{4}{\pi^2}(\arctan\frac{\mathrm{w^{gt}}}{\mathrm{h^{gt}}} - \arctan\frac{\mathrm{w}}{\mathrm{h}})^2 \tag{3}$$

$$\alpha = \frac{v}{(1 - IOU) + v} \tag{4}$$

2.2. The Proposed YOLOv8-CB Algorithm

In this study, the YOLOv8n algorithm is utilized as the baseline for optimization and enhancement. The proposed YOLOv8-CB model introduces a lightweight Cascaded Fusion Network (CFNet) to replace the C2F module in the backbone network. This modification enables the model to simultaneously highlight global features and local details, enriching the spectrum of multi-scale features available for subsequent feature fusion. Additionally, the detection header incorporates five Convolutional Block Attention Module (CBAM) attention mechanisms, directing the model's focus toward semantic and positional information of pedestrians at both channel and spatial levels. Furthermore, an enhanced Bidirectional Weighted Feature Fusion structure (BIFPN) is introduced at the feature enhancement stage to bolster detection performance, particularly in scenarios involving occlusion and pedestrians. The refined YOLOv8n network structure is illustrated in Figure 2.

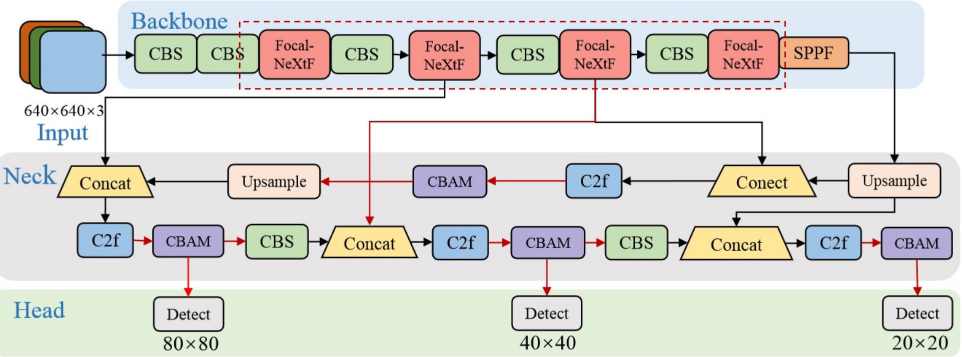


Figure 2. The network structure of YOLOv8-CB. The dashed area is an improvement to the backbone section and the red module is the replacement of the C2F module with a Focal-NeXtF module. The red line connection between the backbone network and the Neck is the BIFPN connection layer. The purple module is the new CBAM Attention Mechanism module. The red lines are the connections of the improved parts.

2.2.1. Cascading Fusion Network CFNet

The enhanced cascade fusion network introduced in this paper implements M cascades of stages within the network's backbone to generate multi-scale features. Each stage comprises a sub-backbone for feature extraction and an exceptionally lightweight transformation block for feature integration. Diverging from the feature pyramid structure (FPN), where features from neighboring layers are summed up and processed by 3×3 convolutional layers for transformation, the cascade fusion network (CFNet) adopts a different approach [31]. The CFNet involves stacking additional convolutional layers to transform integrated features and embeds the feature integration operation into the sub-backbone within the network. This design choice enables the network to more deeply and efficiently fuse features, involving most of the parameters in the entire backbone network. The CFNet fusion module is illustrated in Figure 3.

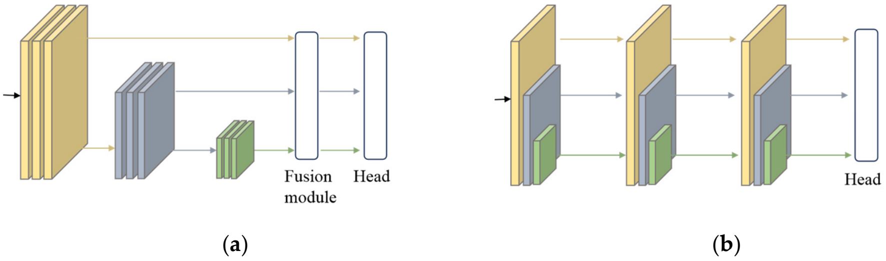


Figure 3. (**a**) FPN and its variants. (**b**) Fusion module for CFNet. By constructing a top-down feature pyramid and fusing multi-scale features. Arrows represent lateral connections of feature maps, and modules of different colors represent feature maps of different scale levels.

Figure 4 illustrates the CFNet network architecture. The input RGB image of size $H \times W$ undergoes processing by a Stem convolutional layer and N consecutive Block feature extraction block to extract high-resolution features of size $H/4 \times W/4$. The Stem convolutional layer comprises two 3×3 convolutional layers with a step size of 2, where each convolutional layer is succeeded by a normalization layer and a GELU activation function. Subsequently, these features undergo downsampling by an 2×2 convolutional layer with a step size of 2 and are directed to M cascade stages to extract multiscale features. The output of each stage consists of P3, P4, and P5 neural network layers with steps of 8, 16, and 32, respectively. Only the P3 layer is forwarded to the next stage, and the fused features (P3, P4, and P5) output in the last stage are utilized for dense detection.

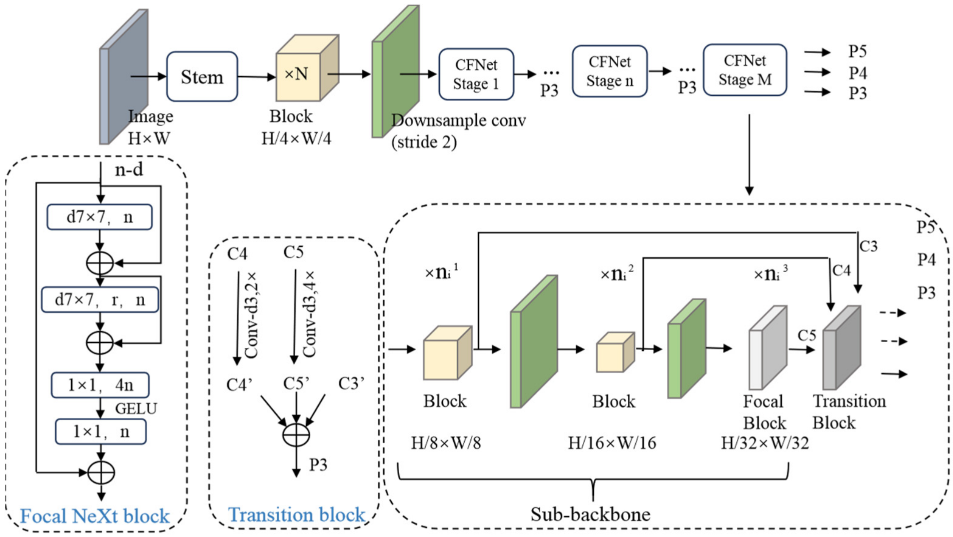


Figure 4. The network structure of CFNet.

Transformation blocks are employed to integrate features at different scales in each stage. Initially, the 1 × 1 convolutional layer is utilized to reduce the number of channels in C4 and C5 to align with C3. The spatial size of the features is standardized using a bilinear interpolation operation before conducting element-by-element summation.

In the focusing block FocalNeXt, expansion depth convolution and two jump connections are introduced to generate features with distinct resolutions. This design is characterized by its ability to simultaneously extract fine-grained local features and coarse-grained global features, thereby enlarging the receptive fields of the neurons in the last block group of each stage without introducing a substantial number of parameters. Here, N represents the number of channels in the output features, $d7 \times 7$ denotes the deep convolution of 7×7, and R is the expansion rate of the convolution. Each convolution of $d7 \times 7$ is succeeded by a normalization layer and a GELU unit. While the utilization of global attention or large convolutional kernels to expand the sensory field has been extensively studied in recent years [32–35], these approaches often introduce significant computational cost and memory overhead, especially in dense prediction tasks, due to the large size of the input image. In contrast, the focusing block proposed in this paper introduces only a minimal additional cost.

2.2.2. Bidirectional Weighted Feature Fusion Method BIFPN

Small-sized targets, such as pedestrian location information, predominantly exist in the shallower layers of the feature extraction network. Throughout the feature extraction process, the shallowest layer may discard significant pedestrian-related information. However, the Weighted Bidirectional Feature Fusion (BIFPN) is designed to understand the importance of various input features and adaptively fuse contextual multi-scale features. The architecture of the BIFPN network is depicted in Figure 5.

In Figure 5, the top-down pathway conveys high-level semantic information, while the low-up pathway carries location information from the underlying features. The same-level cross-node connection pathway is a newly added connection for input and output. The concept behind the bi-directional weighted feature fusion method involves treating each bi-directional path as a feature network layer and repeating the same layer multiple times to efficiently leverage positional and semantic information for feature fusion.

The weighting process introduces a learnable weight parameter, which, if not appropriately constrained, can lead to unstable training. The fusion mechanism of weighted features scales the range to between [0, 1], ensuring fast and efficient training, as depicted in Equation (5). w represents the weight parameters; I_i represents features with different resolutions; ε represents random numbers; O represents the fused feature outputs.

$$\mathrm{O} = \sum\nolimits_i \frac{w_i \times I_i}{\varepsilon + \sum_j w_j} \tag{5}$$

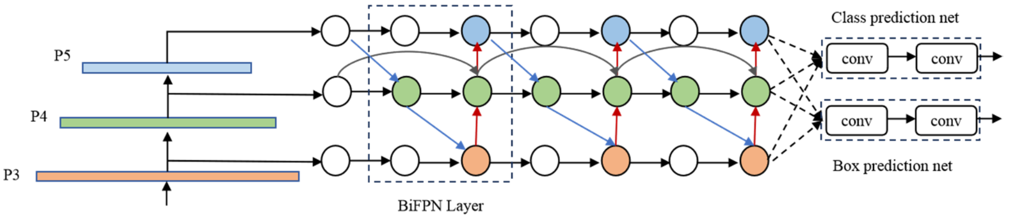


Figure 5. BIFPN module. Extract more multi-scale features of the backbone network by adding paths between the encoder and decoder to aggregate features of different scales and resolutions. Blue color indicates small target detectors. Green indicates medium-sized target detectors and orange indicates large target detectors.

The final feature map, incorporating bi-directional scale linking and fast normalized fusion, is represented by Equations (6) and (7). P_4^{out} denotes the bi-directionally weighted output features of the P_4 convolutional layer; p_4^{td} represents the inverse output features of the P_4 layer; $Resize(\cdot)$ typically denotes an up-sampling or down-sampling stage.

$$P_4^{td} = Conv\left(\frac{w_1 \cdot P_4^{in} + w_2 \cdot Resize(P_5^{in})}{w_1 + w_2 + \varepsilon}\right) \tag{6}$$

$$P_4^{out} = Conv\left(\frac{w_1' \cdot P_4^{in} + w_2' \cdot P_4^{td} + w_3' \cdot Resize(P_3^{out})}{w_1' + w_2' + w_3' + \varepsilon}\right) \tag{7}$$

2.2.3. Channel and Spatial Attention Mechanism CBAM

CBAM (Convolutional Block Attention Module) is a lightweight attention mechanism module [16,36], depicted in Figure 6, encompassing a channel attention module and a spatial attention module. These modules sequentially perform mapping in two dimensions: channel and spatial. The formula for CBAM, presented in Equation (8), involves F as the input feature map, $F \in R^{C\times H\times W}$; M_c as the generated one-dimensional channel attention map, $M_c \in R^{C\times 1\times 1}$; and M_s as the generated two-dimensional spatial attention map, $M_s \in R^{1\times H\times W}$.

$$\begin{aligned} F' &= M_c(F) \otimes F \\ F'' &= M_s(F') \otimes F' \end{aligned} \tag{8}$$

The channel attention module focuses more on channels with robust feature information. The input feature map undergoes global maximum pooling and global average pooling to, respectively, derive one-dimensional vectors of channel attention. This process achieves the compression of spatial dimensions, aggregates the spatial information of the feature mapping, and subsequently, through the shared perceptron MLP module, sums the outputs of the two branches. An activation function is applied to obtain the weighting coefficients. The formula for the channel attention mechanism, as shown in Equation (9), includes the sigmoid function of σ; the shared weights of the MLP denoted by W_0 and W_1, and preceding the ReLU activation functions, $W_0 \in R^{C\times C/r}$ and $W_1 \in R^{C\times C/r}$; global maximum pooling, $\mathrm{F}_{avg}^c \in \mathrm{R}^{1\times H\times W}$; and average maximum pooling, $\mathrm{F}_{\max}^c \in \mathrm{R}^{1\times H\times W}$.

$$M_c(F) = \sigma(MLP(AvgPool(F)) + MLP(\max Pool(F))) = \sigma(W_1(W_0(F_{avg}^c)) + W_1(W_0(F_{\max}^c))) \tag{9}$$

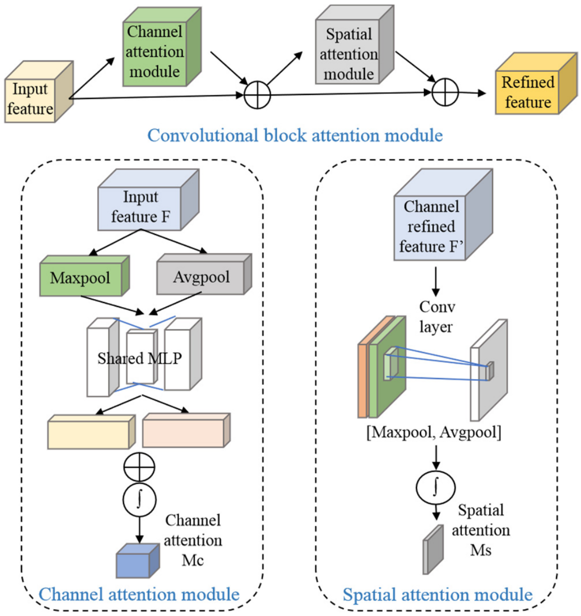


Figure 6. Convolutional block attention module.

The spatial attention module is designed to prioritize the more crucial information within the input image. Initially, global maximum pooling and global average pooling operations are employed to generate a comprehensive feature description, which is then concatenated by the channel. Finally, weight coefficients are obtained through a convolutional layer and a sigmoid function. The formula for the Spatial Attention Mechanism, presented in Equation (10), involves $f^{7\times7}$ as a 7 × 7 convolutional kernel; and the feature description comprises both global maximum pooling, $\mathrm{F}^{s}_{avg} \in \mathrm{R}^{1\times H\times W}$, and global average pooling, $\mathrm{F}^{s}_{\max} \in \mathrm{R}^{1\times H\times W}$.

$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)])) = \sigma(f^{7\times7}([F^{s}_{avg}; F^{s}_{\max}])) \tag{10}$$

In this study, five CBAM attention mechanism modules are incorporated into the Head section to achieve feature refinement and extraction in two different dimensions: channel and space. This specifically targets local pedestrian information in the image, aiming to reduce the loss of feature extraction in scenarios involving background complexity and occlusion. The overall goal is to enhance the network's expressive ability and improve the average accuracy of pedestrian detection at intersections.

3. Results

3.1. Experimental Setup and Dataset Preparation

The hardware platform utilized for this study featured a high-performance NVIDIA GeForce RTX 4090 GPU with 24 GB VRAM. The software environment included the Ubuntu 20.04 operating system, the PyTorch 1.10.0 deep learning framework, Torchvision 1.13.1+python3.8_cu113, Anaconda, and CUDA 11.3 with CUDNN 8.0.

For the dataset, the WiderPerson public dataset was chosen [37], supplemented by real street intersection images, creating a pedestrian detection dataset suitable for dense scenes, accounting for various pedestrian occlusion scenarios. To streamline the dataset, we retained labels for ordinary pedestrians, bicyclists, and individuals with occluded bodies, assigning the new indexes "0, 1, 2", respectively. The image paths for the training and test sets were generated, and the data were transformed into a dictionary, with a JSON file serving as the label file. The corresponding dataset configuration file was created to align with the model requirements. The processed dataset comprised 13,381 images, with 7999 allocated to the training set, 1000 to the test set, and 4382 to the validation set. It included approximately 400,000 labeled frames.

Regarding model training parameters, an image size of 640 × 640 was set, and dataset augmentation involved scaling, rotation, and random adjustments to brightness and contrast. Training commenced with the official pre-trained YOLOv8n model, conducting 100 rounds on the dataset. The trained weight files were then employed for further training on the new model. The gradient descent optimization algorithm was applied to dynamically adjust the learning rate and weight decay coefficient, mitigating overfitting. After several validations, the initial learning rate was set to 0.004, the weight decay factor to 0.0005, the batch size to 64, and the number of iterations (epochs) to 100.

3.2. Experimental Evaluation Index

In this study, we used mean average precision (mAP), frame rate (Frames Per Second or FPS), the number of parameters (params), and Floating-Point Operations per Second (FLOPs) as key indicators to assess the detection accuracy, speed, model size, and complexity. Average precision (AP) is a measure of the detection capability of a trained model to detect specific categories of interest. It is calculated as the area of the region enclosed by the precision–recall (P-R) curve and the axes. As defined in Equations (11) and (12), TP is the number of correctly predicted bounding boxes; FP is the number of incorrectly predicted positive samples; FN is the number of non-detected positive samples; AP is the average precision per category; mAP is the average precision across all categories; k is the number of categories.

$$P = \frac{TP}{(TP+FP)} \quad R = \frac{TP}{(TP+FN)} \tag{11}$$

$$AP = \int_0^1 P{\cdot}RdR \quad mAP = \frac{1}{k}\sum_{k}^{i=1} AP_i \tag{12}$$

A higher mAP@0.5 signifies improved average precision when the Intersection over Union (IOU) threshold exceeds 0.5, indicative of enhanced detection effectiveness. The frame rate represents the number of images detected by the model per second, denoted in frames per second (frame/s). The number of parameters quantifies the memory footprint of the model in megabytes (MB), while FLOPs measures the computational complexity in gigaflops (G), or one billion floating-point operations. The loss function serves as a crucial metric reflecting the model's convergence speed during training, encompassing localization loss, classification loss, and confidence loss.

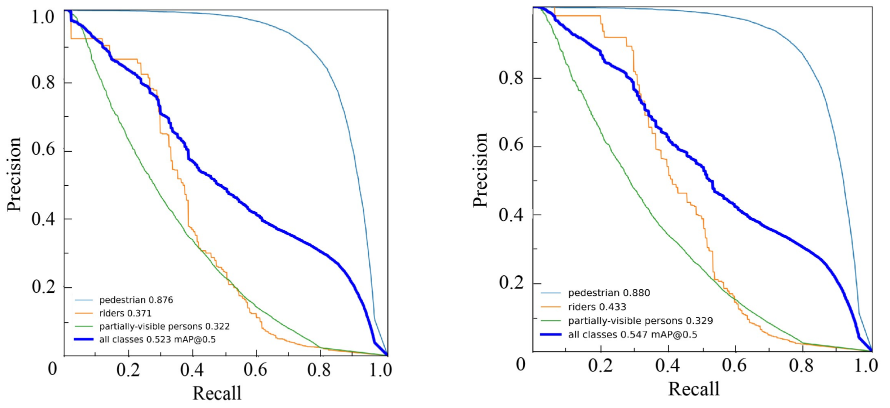
3.3. Data Analysis

To validate the efficacy of the improved algorithm, this paper conducts training and testing of both the YOLOv8n and YOLOv8-CB models on the respective dataset. The detection results for each category are presented in Table 1.

Table 1. Comparison of detection results before and after improvement of the YOLOv8n algorithm for WiderPerson datasets.

Method	Pedestrian	Riders	Partially Visible-Persons	P	R	mAP0.5
YOLOv8n	87.6	37.1	32.2	59.2	49.4	52.3
YOLOv8-CB	88.0	43.3	32.9	59.8	51.4	54.7

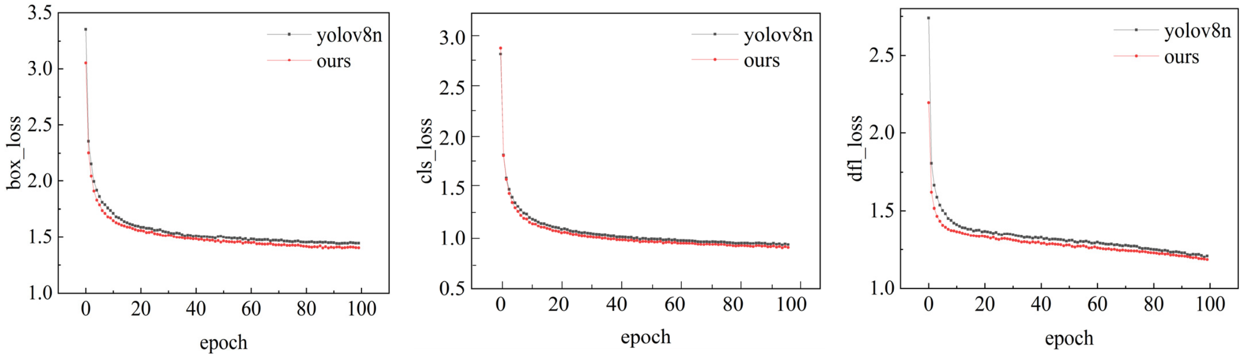
From Table 1, the YOLOv8-CB algorithm demonstrates a 2.4% improvement in the mAP@0.5 index compared to the original algorithm. The detection accuracy for each category in the dataset is enhanced. The P-R curves are shown in Figure 7. Notably, in categories with fewer samples, such as riders, where the recognition difficulty is high, YOLOv8-CB exhibits more pronounced improvements in detection accuracy, the accuracy of the behavioral categories is up to 88%.



(**a**) (**b**)

Figure 7. Comparison of PR curves of YOLOv8n and YOLOv8-CB, detection accuracy in each category and mAP@0.5. (**a**) The detection result of YOLOv8n. (**b**) The detection result of YOLOv8-CB.

As depicted in Figure 8, during network model convergence, the YOLOv8-CB algorithm exhibits lower loss values across all three categories compared to the YOLOv8n algorithm, indicating faster model convergence.



(**a**) (**b**) (**c**)

Figure 8. Comparison before and after improvement of YOLOv8n. (**a**) The decay curve of localization loss. (**b**) The decay curve of classification loss. (**c**) The decay curve of confidence loss.

In this study, ablation experiments are conducted to validate the effectiveness of each improved module. The YOLOv8n network serves as the baseline, with the four C2F modules in the backbone network replaced by FocalNeXtF modules, denoted as FN modules. This replacement, while increasing the network's layer count, significantly reduces the overall parameter count. Subsequently, we assess the impact of various combinations of the FN module in the backbone network, the enhanced multilayer feature fusion BIFPN connectivity layer, and the attention mechanism module CBAM on the holistic model performance. The experimental results are presented in Table 2.

From Table 2, it is evident that, based on the YOLOv8n model, increasing the FN module in the backbone network results in a computation of 7.4 GFLOPs, a reduction of 0.6 MB in the number of parameters, and a decrease of 8.37 frames/s in frame rate (FPS). When only the FN module and the four-layer CBAM attention mechanism module are added, the computation increases to 7.5 GFLOPs. The average detection accuracy mAP@0.5 improves to 54.2%, a 1.9% enhancement. Introducing only the FN module and the BIFPN connectivity layer results in a computation of 7.4 GFLOPs, with a frame rate FPS improvement of 7.1 frames/s. In comparison to the YOLOv8n model, the YOLOv8-CB model in this paper reduces computation by 0.7 GFLOPs, achieving an average detection accuracy mAP@0.5 of 54.7%, a 2.4% improvement. The number of parameters is reduced by 2.7 M, and the single-image inference time is 10.8 ms, with a slight decrease in frame rate FPS.

Table 2. Experiment results of ablation experiment.

Method	FN	CBAM	BIFPN	mAP/%	GFLOPs	Params(M)	FPS (Frame/s)
Yolov8n				52.3	8.2	3.2	113.63
Yolov8n + FN	√ [1]			53.5	7.4	2.6	105.26
Yolov8n + FN + CBAM	√	√		54.2	7.5	2.7	91.74
Yolov8n + FN + BIFPN	√		√	54.3	7.4	2.6	112.36
Yolov8-CB	√	√	√	54.7	7.5	2.7	92.59

[1] √ indicates the module used.

The model testing results reveal that in the pre-processing and post-processing stages, YOLOv8-CB takes the same time as YOLOv8n, at 0.2 ms and 0.8 ms, respectively. However, in the inference stage, YOLOv8-CB incurs an additional 2 ms compared to YOLOv8n. This increase is attributed to the improvement of the YOLOv8n algorithm, which introduced a series of modules, leading to an increase in the number of layers in the network and, consequently, an increase in inference complexity.

To offer a more intuitive comparison between the YOLOv8-CB algorithm and the baseline model, Figure 9 illustrates complex scenarios depicting intersections in life. Upon examining the visualization results, it is evident that YOLOv8-CB excels in detecting pedestrians at a distance on the road, as well as in densely populated areas with overlapping objects.

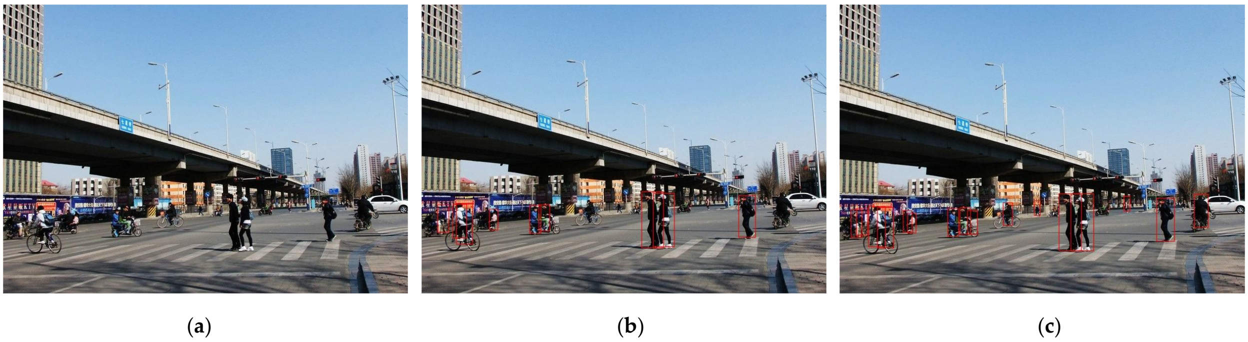

(**a**) (**b**) (**c**)

Figure 9. Selected detection results of YOLOv8n and YOLOv8-CB in the traffic intersection life scene. It can be seen from the figure that the detection results of YOLOv8n are not as good as those of YOLOv8-CB. (**a**) Input image. (**b**) Detection results of YOLOv8n with missed detection of small targets at long range in the figure. (**c**) Detection result of YOLOv8-CB, which detects small targets at a long distance.

3.4. Comparative Analysis

In our pursuit to ascertain the efficacy of our enhanced algorithm for navigating challenges posed by occlusions and small targets, we meticulously selected two particularly demanding datasets: VisDrone and CrowdHuman [38,39]. The VisDrone dataset, a comprehensive collection curated by the AISKYEYE team at Tianjin University's Machine Learning and Data Mining Lab, offers an extensive assortment of scenes captured through a variety of drone cameras under diverse weather and lighting conditions. It boasts detailed annotations concerning scene visibility, object classifications, and occlusions, with a specific emphasis on a multitude of small-target pedestrians and occluded individuals, thus positioning it as one of the most authoritative and challenging datasets available. Additionally, the CrowdHuman dataset, introduced by Megvii Technology, is tailor-made for the detection of densely occluded pedestrians. This dataset is characterized by its considerable volume, averaging about 23 individuals per image, and covers a wide spectrum of occlusions. Each human instance within this dataset is meticulously annotated with head bounding boxes, visible body area bounding boxes, and full-body bounding boxes, making it an invaluable resource for research in dense pedestrian detection. In this paper, the improved algorithms are comprehensively compared and tested against existing mainstream object detection algorithms on these selected datasets, including SSD (VGG), YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, YOLOv7-tiny, YOLOv8n, and YOLOv8-CB. Each algorithm is configured with 100 epochs focusing on recording its mAP0.5 and mAP0.5:0.95 metrics. The detection results of the different algorithms on the three datasets are presented in Table 3.

Table 3. Comparison of detection results of algorithms for different datasets.

Datasets	Result	SSD(VGG)	YOLOv3-tiny	YOLOv4-tiny	YOLOv5s	YOLOv7-tiny	YOLOv8n	YOLOv8-CB
Visdrone	mAP0.5	23.2	21.0	18.2	27.4	25.6	28.4	30.6
	mAP0.5:0.95	11.6	11.3	10.3	15.2	12.3	15.7	17.7
Crowdhuman	mAP0.5	70.2	68.6	51.5	77.8	75.2	78.2	80.1
	mAP0.5:0.95	43.5	40.2	32.4	46.2	43.9	46.7	48.5
WiderPerson	mAP0.5	49.5	49.0	40.8	51.2	50.2	52.3	54.7
	mAP0.5:0.95	28.3	27.9	21.2	29.8	28.9	31.4	32.6

The data in Table 3 clearly show that YOLOv8-CB performs excellently and outperforms other popular target detection algorithms in terms of experimental performance, thus highlighting its comprehensive ability to detect pedestrians in challenging environments.

To assess the effectiveness of the improved algorithms, this study conducts a quantitative analysis and comparison of the results against current mainstream target detection algorithms. The evaluated models include the one-stage anchor-based frame detection model SSD, as well as various models from the YOLO family, namely YOLOv3-tiny, YOLOv4-tiny, YOLOv5s, and YOLOv7-tiny. The models undergo training and validation on the WiderPerson dataset, and the results are summarized in Table 4.

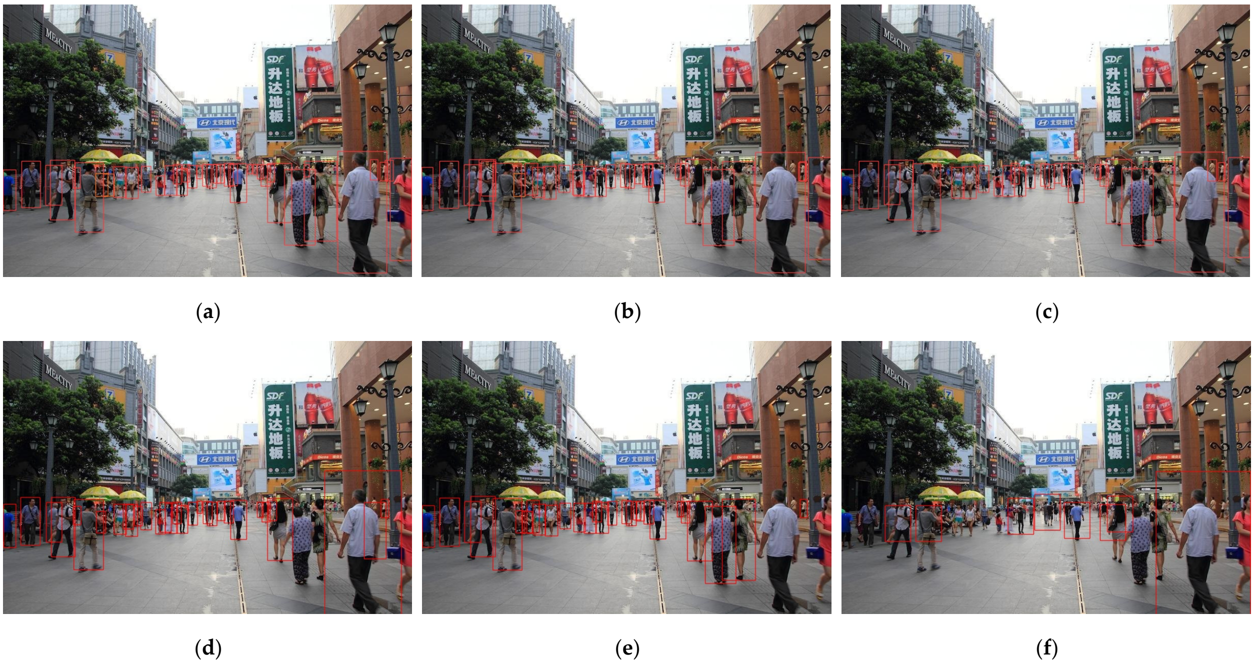
Table 4. Experiment results of contrast experiment.

Method	mAP/%	FLOPs (G)	Params (M)	FPS (Frame/s)
SSD(VGG)	49.5	62.7	26.3	35
YOLOv3-tiny	49.0	19.1	12.1	182.01
YOLOv4-tiny	40.8	16.5	6.1	112.36
YOLOv5s	51.2	15.8	7.1	70.42
YOLOv7-tiny	50.2	13.2	6.07	109.24
YOLOv8n	52.3	8.2	3.2	113.63
YOLOv8-CB	54.7	7.5	2.7	92.59

Examining Table 4 reveals that, when compared with other algorithms, the SSD algorithm demonstrates less favorable performance in pedestrian detection at intersections. This can be attributed to its higher model complexity, fewer layers in the low-level feature convolutional network, inadequate feature extraction for small target pedestrians, and a low frame rate, failing to meet the requirements for both detection accuracy and real-time applications. YOLOv3 adopts the feature pyramid concept to introduce a multi-scale detection mechanism, resulting in improved detection accuracy. However, due to residuals, the performance of YOLOv7-tiny surpasses that of YOLOv4-tiny, which adopts only one feature pyramid based on the YOLOv4 algorithm and has a lower frame rate. Although it achieves faster and lighter models, the insufficient feature extraction, as only two layers of multi-scale features are selected for the feature pyramid, leads to lower accuracy in detecting occlusions and pedestrians. YOLOv5s, a smaller training model to YOLOv5, exhibits reduced channel numbers and depth in the middle layer. YOLOv7-tiny reduces the number of channels and depth in ELAN based on the yolov7 network, resulting in reduced model complexity but with a more significant impact on pedestrian detection. The network achieves speed and parameter reduction by decreasing the stacking of convolutional layers in ELAN and altering the number of activations. However, this comes at the cost of less fusion of multi-scale features in the backbone network, resulting in poor robustness for detecting pedestrians at different scales.

The improved YOLOv8-CB algorithm presented in this paper achieves a mAP@0.5 of 54.7%, a frame rate (FPS) of 92.59 frames/s, with a model parameter count of only 2.7 M, and computational load of 7.5 GFLOPs, bringing it closer to the FPS achieved by YOLOv8n. Experimental results demonstrate that the enhanced algorithm outperforms other algorithms in terms of computation, detection accuracy, and frame rate. It is particularly well-suited for pedestrian detection at intersections.

To provide a more intuitive observation of each algorithm's detection effect, the YOLOv8-CB, YOLOv8n, YOLOv5s, YOLOv7-tiny, YOLOv3-tiny, and YOLOv4-tiny algorithms are applied to a real-life scene: a street intersection with multi-masking and multi-scale features. For ease of observation, only the output of the detection frames is retained in this paper, as depicted in Figure 10. In Figure 10a–f, it is evident that YOLOv8-CB outperforms YOLOv8n in detecting large and medium-sized pedestrians in close proximity and in scenes with moderate occlusion. YOLOv5s and YOLOv3-tiny algorithms excel in detecting large targets but exhibit duplication of detection frames and omission in cases of occluded targets. YOLOv7-tiny and YOLOv4-tiny algorithms display inaccuracies in detecting near large-size pedestrian targets, leading to high false positives and missed detection rates. In the detection of distant, small-sized pedestrians in heavily occluded scenes, the YOLOv5s and YOLOv8n algorithms are significantly less effective in detecting pedestrians, especially for small distant targets. The YOLOv7-tiny, YOLOv3-tiny, and YOLOv4-tiny algorithms are less efficient in detecting highly occluded targets, resulting in a high rate of missed detection.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 10. Pedestrian detection results of various algorithms in complex intersection scenarios. Each sub-figure illustrates the performance of the algorithms in detecting pedestrians in complex scenes. From the graphs, it can be concluded that YOLOv8-CB has the best detection results and can detect both near and far pedestrian targets. YOLOv4-tiny has the worst detection results. (**a**) The detection results of YOLOv8-CB; (**b**) The detection results of YOLOv8n, which has repeated detection and omission phenomenon for overlapping and small target pedestrian detection; (**c**) The detection results of YOLOv5s, which can detect near unobstructed pedestrians; (**d**) The detection results of YOLOv7-tiny, which has a high misdetection rate of near large targets; (**e**) The detection results of YOLOv3-tiny, which does not detect near and far pedestrian targets; (**f**) YOLOv4-tiny's detection results, which have poor detection results and slightly lower detection accuracy.

4. Conclusions

In this study, we propose a cascade fusion algorithm based on the improved YOLOv8n. The algorithm introduces a novel architecture, the cascade fusion network (CFNet), to replace the C2F module in the backbone network, enhancing the extraction of multi-scale features. Additionally, a five-layer CBAM attention mechanism is integrated into the decoupled head section to augment the semantic and positional information of small targets, addressing the challenge of inaccurate localization that can lead to detection leakage. The model incorporates a multi-layer two-way weighted feature fusion structure in the feature fusion stage, enabling efficient utilization of deep and effective information to further mitigate the leakage detection issues associated with small targets and occlusions. Through rigorous analysis and experiments, the improved algorithm demonstrates a 2.4% enhancement in detection accuracy, a reduction of 0.5 MB in the number of parameters, a decrease of 0.7 GFLOPs in computational load, and a slight reduction in the frame rate (FPS). The inference time for a single image is 10.8 ms. The results illustrate that YOLOv8-CB outperforms other detectors, excelling in comprehensive performance metrics such as accuracy and model efficiency. The algorithm demonstrates superior detection performance, particularly in dense areas with high pedestrian flow, such as street intersections. While the model exhibits commendable detection performance in dense environments, improvements are needed for detecting severely occluded pedestrians with small targets. Future work will focus on optimizing the network model to further enhance overall performance.

Author Contributions: Q.L. was responsible for primary writing and data preparation and experimental reasoning and revision of the paper. H.Y., S.W. and Z.X. gave technical and writing method guidance as instructors. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by Shiming Wang, Dean of the College of Engineering, Shanghai Ocean University. This work is supported by the National Natural Science Foundation of China (NSFC) (No.41976194), the Soft Science Research Project of Shanghai Municipal Science and Technology Innovation Action Plan (23692102600), and the Shanghai Engineering and Technology Research Center of Shanghai Municipal Science and Technology Innovation Action Plan (19DZ2254800).

Data Availability Statement: Data is contained within the article.

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

References

1. Geng, Y.N.; Liu, S.S.; Liu, T.T.; Yan, W.Y.; Lian, Y.F. A review of pedestrian detection techniques based on computer vision. *Comput. Appl.* **2021**, *41*, 43–50.
2. Brunetti, A.; Buongiorno, D.; Trotta, G.F.; Bevilacqua, V. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing* **2018**, *300*, 17–33. [CrossRef]
3. Dollar, P.; Wojek, C.; Schiele, B.; Perona, P. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 743–761. [CrossRef] [PubMed]
4. Xiao, Y.Q.; Yang, H.M. A review of target detection algorithms in traffic scenarios. *Comput. Eng. Appl.* **2021**, *57*, 30–41.
5. Yuan, J.; Zhang, G.; Li, F.; Liu, J.; Xu, L.; Wu, S.; Jiang, T.; Guo, D.; Xie, Y. Independent moving object detection based on a vehicle mounted binocular camera. *IEEE Sens. J.* **2020**, *21*, 11522–11531. [CrossRef]
6. Wei, Y.; Xu, C.Q.; Diao, Z.F.; Li, B.Q. Multi-target pedestrian tracking algorithm based on generative adversarial network. *J. Northeast. Univ.* **2020**, *41*, 1673–1679+1720.
7. Ke, W.; Zhang, T.L.; Huang, Z.Y.; Ye, Q.X.; Liu, J.Z.; Huang, D. Multiple anchor learning for visual object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10203–10212.
8. Jain, D.K.; Zhao, X.; González-Almagro, G.; Gan, C.; Kotecha, K. Multimodal pedestrian detection using metaheuristics with deep convolutional neural network in crowded scenes. *Inf. Fusion* **2023**, *95*, 401–414. [CrossRef]
9. Ferraz, P.A.P.; de Oliveira, B.A.G.; Ferreira, F.M.F.; Martins, C.A.P.D.S. Three-stage RGBD architecture for vehicle and pedestrian detection using convolutional neural networks and stereo vision. *IET Intell. Transp. Syst.* **2020**, *14*, 1319–1327. [CrossRef]
10. Hou, Y.L.; Song, Y.; Hao, X.; Shen, Y.; Qian, M.; Chen, H. Multispectral pedestrian detection based on deep convolutional neural networks. *Infrared Phys. Technol.* **2018**, *94*, 69–77. [CrossRef]

11. Shen, C.; Zhao, X.; Fan, X.; Lian, X.; Zhang, F.; Kreidieh, A.R.; Liu, Z. Multi-receptive field graph convolutional neural networks for pedestrian detection. *IET Intell. Transp. Syst.* **2019**, *13*, 1319–1328. [CrossRef]
12. Panigrahi, S.; Raju, U.S.N. Pedestrian Detection Based on Hand-crafted Features and Multi-layer Feature Fused-ResNet Model. *Int. J. Artif. Intell. Tools* **2021**, *30*, 2150028. [CrossRef]
13. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
14. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
15. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
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, Proceedings, Part I 14*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
17. Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. *Machines* **2023**, *11*, 677. [CrossRef]
18. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef] [PubMed]
19. Zha, M.; Qian, W.; Yi, W.; Hua, J. A lightweight YOLOv4-Based forestry pest detection method using coordinate attention and feature fusion. *Entropy* **2021**, *23*, 1587. [CrossRef] [PubMed]
20. Kou, X.; Liu, S.; Cheng, K.; Qian, Y. Development of a YOLO-V3-based model for detecting defects on steel strip surface. *Measurement* **2021**, *182*, 109454. [CrossRef]
21. Zhang, X.Z.; Qiu, Y.; Zhang, C. Improved YOLOv5s algorithm for pedestrian target detection in subway scenes. *Adv. Laser Optoelectron.* **2023**, *60*, 144–153.
22. Ding, Z.; Gu, Z.; Sun, Y.; Xiang, X. Cascaded Cross-Layer Fusion Network for Pedestrian Detection. *Mathematics* **2022**, *10*, 139. [CrossRef]
23. Tan, M.X.; Pang, R.M.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
24. Lv, Z.X.; Wei, X.; Huang, D.Q. Dense pedestrian detection algorithm for multi-branch anchorless frame networks. *Opt. Precis. Eng.* **2023**, *31*, 1532–1547.
25. Zhou, D.K.; Song, R.; Yang, X. Occlusion-aware pedestrian detection combining dual attention mechanisms. *J. Harbin Inst. Technol.* **2021**, *53*, 156–163.
26. Gu, Z.C.; Zhu, K.; You, S.T. YOLO-SSFS: A Method Combining SPD-Conv/STDL/IM-FPN/SIoU for Outdoor Small Target Vehicle Detection. *Electronics* **2023**, *12*, 3744. [CrossRef]
27. Lou, H.; Duan, X.; Guo, J.; Liu, H.; Gu, J.; Bi, L.; Chen, H. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics* **2023**, *12*, 2323. [CrossRef]
28. Niu, W.H.; Yin, M.M. Road small target detection algorithm based on improved YOLOv5. *J. Sens. Technol.* **2023**, *36*, 36–44.
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, 18–23 June 2018; pp. 8759–8768.
31. Lv, X.D.; Wang, S.; Ye, D. CFNet: LiDAR-camera registration using calibration flow network. *Sensors* **2021**, *21*, 8112. [CrossRef] [PubMed]
32. Wang, Y.; Huang, R.; Song, S.; Huang, Z.; Huang, G. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 11960–11973.
33. Rao, Y.; Zhao, W.; Zhu, Z.; Lu, J.; Zhou, J. Global filter networks for image classification. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 980–993.
34. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
35. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31 × 31: Revisiting large kernel design in cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11963–11975.
36. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; Proveedings, Part VII; pp. 3–19.
37. Zhang, S.; Xie, Y.; Wan, J.; Xia, H.; Li, S.Z.; Guo, G. Widerperson: A diverse dataset for dense pedestrian detection in the wild. *IEEE Trans. Multimed.* **2019**, *22*, 380–393. [CrossRef]

38. Cao, Y.; He, Z.; Wang, L.; Wang, W.; Yuan, Y.; Zhang, D.; Zhang, J.; Zhu, P.; Van Gool, L.; Han, J.; et al. VisDrone-DET2021: The Vision Meets Drone Object Detection Challenge Results. In Proceedings of the 2021 IEEE International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, 11–17 October 2021; pp. 2847–2854.
39. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. CrowdHuman: A Benchmark for Detecting Human in a Crowd 2018. *arXiv* **2018**. [CrossRef]

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