symmetry

MDPI

Article

An Improved Pedestrian Detection Model Based on YOLOv8 for Dense Scenes

Yuchao Fang and Huanli Pang *

School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China; fangyc392@gmail.com
* Correspondence: panghuanli@ccut.edu.cn

Abstract: In dense scenes, pedestrians often exhibit a variety of symmetrical features, such as symmetry in body contour, posture, clothing, and appearance. However, pedestrian detection poses challenges due to the mutual occlusion of pedestrians and the small scale of distant pedestrians in the image. To address these challenges, we propose a pedestrian detection algorithm tailored for dense scenarios called YOLO-RAD. In this algorithm, we integrate the concept of receiving field attention (RFA) into the Conv and C2f modules to enhance the feature extraction capability of the network. A self-designed four-layer adaptive spatial feature fusion (ASFF) module is introduced, and shallow pedestrian feature information is added to enhance the multi-scale feature fusion capability. Finally, we introduce a small-target dynamic head structure (DyHead-S) to enhance the capability of detecting small-scale pedestrians. Experimental results on WiderPerson and CrowdHuman, two challenging dense pedestrian datasets, show that compared with YOLOv8n, our YOLO-RAD algorithm has achieved significant improvement in detection performance, and the detection performance of mAP@0.5 has increased by 2.5% and 6%, respectively. The detection performance of mAP@0.5:0.95 was improved by 2.7% and 6.8%, respectively. Therefore, the algorithm can effectively improve the performance of pedestrian detection in dense scenes.

Keywords: YOLOv8; dense pedestrian; receptive field attention; adaptive spatial feature fusion; dynamic head

check for updates

Citation: Fang, Y.; Pang, H. An Improved Pedestrian Detection Model Based on YOLOv8 for Dense Scenes. *Symmetry* **2024**, *16*, 716. https://doi.org/10.3390/sym16060716

Academic Editors: Changxin Gao, Yuanjie Shao and Jie Yang

Received: 2 May 2024
Revised: 29 May 2024
Accepted: 6 June 2024
Published: 9 June 2024

1. Introduction

In today's society, urbanization is advancing rapidly, leading to a continual increase in population density within cities. The high mobility of crowds in these dense urban scenes poses significant challenges for traffic management, security monitoring, and various other fields. Consequently, the importance of object detection technology, particularly pedestrian detection, has become more pronounced. Despite significant progress in the field of object detection, accurately identifying pedestrians in dense scenes remains a formidable task. Traditional algorithms rely on manually engineered feature representations. For example, Haar wavelet features combine human motion and appearance features [1], HOG features analyze edge direction information for pedestrian contour outlining [2], and LBP features [3] provide grayscale and rotation invariance, along with scale invariance similar to the SIFT feature [4]. However, these traditional methods often suffer from drawbacks such as slow processing speeds and lower accuracy levels.

In recent years, deep learning has revolutionized pedestrian detection, with algorithms broadly categorized into two types: two-stage and one-stage algorithms. Two-stage algorithms, like the R-CNN series [5–8] and SPPNet [9], first identify regions of interest within an image and then use a classification network to detect objects in these regions. In contrast, one-stage algorithms, such as SSD [10], YOLO series [11–17], RetinaNet [18], and CenterNet [19], directly predict object locations and categories without the need for region proposal extraction. These algorithms have shown significant improvements in detection performance by eliminating the region proposal step.

Nevertheless, pedestrian detection in dense scenes remains challenging due to occlusion between pedestrians and difficulties in detecting small-scale pedestrians. To address these challenges, this paper proposes the YOLO-RAD pedestrian detection algorithm, with the following key contributions:

1. In this study, receptive field attention (RFA) [20], combined with Conv and C2f modules in the model, solves the parameter sharing problem of the convolutional kernel and significantly improves the ability of the network to extract pedestrian features.
2. In this study, the neck of YOLOv8 was improved, and a four-layer adaptive spatial feature fusion (ASFF) [21] module was designed to reduce the feature information conflicts between different feature layers in the fusion process. Experiments show that the proposed method can effectively enhance the fusion ability of pedestrian feature information between different feature layers.
3. In this study, a small-target dynamic head structure (DyHead-S) based on the dynamic head (DyHead) framework [22] is proposed and used to improve the ability to detect small-scale pedestrians.

2. Related Work

2.1. Pedestrian Detection

Despite significant advancements in pedestrian detection technology, multi-scale and occlusion issues remain major challenges. Yang et al. [23] proposed the Scale-Sensitive Feature Reorganization Network (SSNet), which utilizes multiple parallel branch sampling modules to flexibly adjust the receptive field and anchor stride to extract scale-sensitive features. Additionally, a context-enhanced fusion module was introduced to reduce information loss in mid-to-high-level features. Although SSNet performs well in detecting small-scale pedestrians, it fails to meet real-time requirements. Ma et al. [24] proposed the MSCM ANet network, which introduces multi-scale convolution modules and adds attention modules to focus the detection network on pedestrian features. Although MSCM ANet improves detection accuracy, it reduces detection speed. Yan et al. [25] proposed R-SSD based on the SSD architecture, where different scale feature maps are fused during the feature fusion process, and the fusion blocks are combined with other layers to generate six prediction layers of varying depths. Each prediction layer in SSD includes residual blocks to enhance prediction performance. This method does not require anchor configuration for different datasets but performs poorly in crowded pedestrian scenes. Yang et al. [26] proposed a pedestrian detection method based on parallel feature fusion using the Choquet integral. The integration of the Choquet integral allows for the parallel fusion of HOG and LBP features. The resulting parallel feature, HOG-HOLBP, not only retains the advantages of HOG and LBP but also avoids the dimensionality disaster inherent in traditional serial fusion. Chintakindi Balaram Murthy et al. [27] introduced an improved YOLOv2 pedestrian detection algorithm (YOLOv2PD) that employs a multi-layer feature fusion (MLFF) strategy to enhance the model's feature extraction capabilities and removes one convolutional layer in the final stage to reduce computational complexity. Additionally, the normalized improved loss function is used to enhance detection performance. YOLOv2PD can perform real-time detection; however, its performance is suboptimal for detecting small-scale and occluded pedestrians. Most pedestrian detectors perform well when visibility is high and occlusion is minimal. However, their performance may degrade when pedestrians are occluded, particularly under severe occlusion conditions. Liu et al. [28] proposed a global context-aware feature extraction module that integrates contextual information with both local and global pedestrian features. Additionally, they designed a visual feature enhancement module that incorporates unincluded upper body information into the network to enhance the representation of extracted features. However, this method shows inconsistent performance across different datasets and has weak generalization capabilities. Qin et al. [29] proposed the FE-CSP single-stage pedestrian detection algorithm, which combines GCB and attention mechanisms and uses deformable convolutions to enhance the feature extraction capability of the backbone network. Additionally, it employs a feature pyramid network to fuse low-level and high-level features, capturing more semantic information. However, FE-CSP struggles to accurately detect individual pedestrians in very crowded situations. He et al. [30] introduced the DMSFLN pedestrian detection network, which employs both a standard full-body detection branch and an additional visible-body branch for pedestrian detection. These two branches are supervised by full-body and visible-body annotations, respectively, but the performance in crowded scenes is still suboptimal.

In summary, addressing the aforementioned challenges, this study proposes the YOLO-RAD model. This model not only enhances the network's ability to extract pedestrian features but also significantly improves the recognition of small and occluded targets in dense scenes. It achieves good detection performance on the WiderPerson and CrowdHuman datasets.

2.2. YOLOv8 Network Model

YOLOv8 [31] represents an advanced object detection model, building upon previous iterations with optimizations and enhancements for higher performance, flexibility, and efficiency. It consists of three main components: the backbone network, neck network, and detection head. Notably, the backbone network is based on the Darknet53 architecture, with the C2f module replacing the commonly used C3 module, inspired by the ELAN philosophy from YOLOv7. The SPPF module enhances the Spatial Pyramid Pooling (SPP) module, improving computational speed and addressing redundant information in feature extraction. The neck network combines FPN [32] and PAN [33] structures for enhanced feature fusion, facilitating the integration of high-level features with low-level feature maps. In YOLOv8, the detection head has shifted from the original coupling head to the decoupling head and from the Anchor-Based approach of YOLOv5 to Anchor-Free. This updated design eliminates the previous Objectness branch, replacing it with decoupled classification and regression branches. The classification branch employs BCE Loss, while the regression branch utilizes Distribution Focal Loss (DFL) [34] alongside CIoU Loss.

Most contemporary object detectors emphasize positive and negative sample allocation strategies. Notable examples include simOTA in YOLOX [35], Task Aligned Assigner in TOOD [36], and Dynamic Soft Label Assigner in RTMDet [37]. These assigners primarily utilize dynamic allocation strategies, while YOLOv5 still adheres to a static allocation strategy. Recognizing the advantages of dynamic allocation strategies, the YOLOv8 algorithm directly incorporates the Task Aligned Assigner from TOOD; it involves the measurement of task alignment and the strategy of sample allocation. Specifically, it uses a combination of the classification score and the higher-power product of the IoU to measure the degree of task alignment, as illustrated in Equation (1):

$$t = s^{\alpha} \times u^{\beta}, \tag{1}$$

where s denotes the predicted score corresponding to the annotated category and u represents the intersection over union (IoU) between the predicted box and the ground truth box.

Indeed, YOLOv8 has surpassed many previous models with its excellent detection performance, making it one of the most popular object detection networks currently available. Moreover, it offers scalability by providing models of different scales to meet various usage requirements. The structure of the YOLOv8 model is shown in Figure 1.

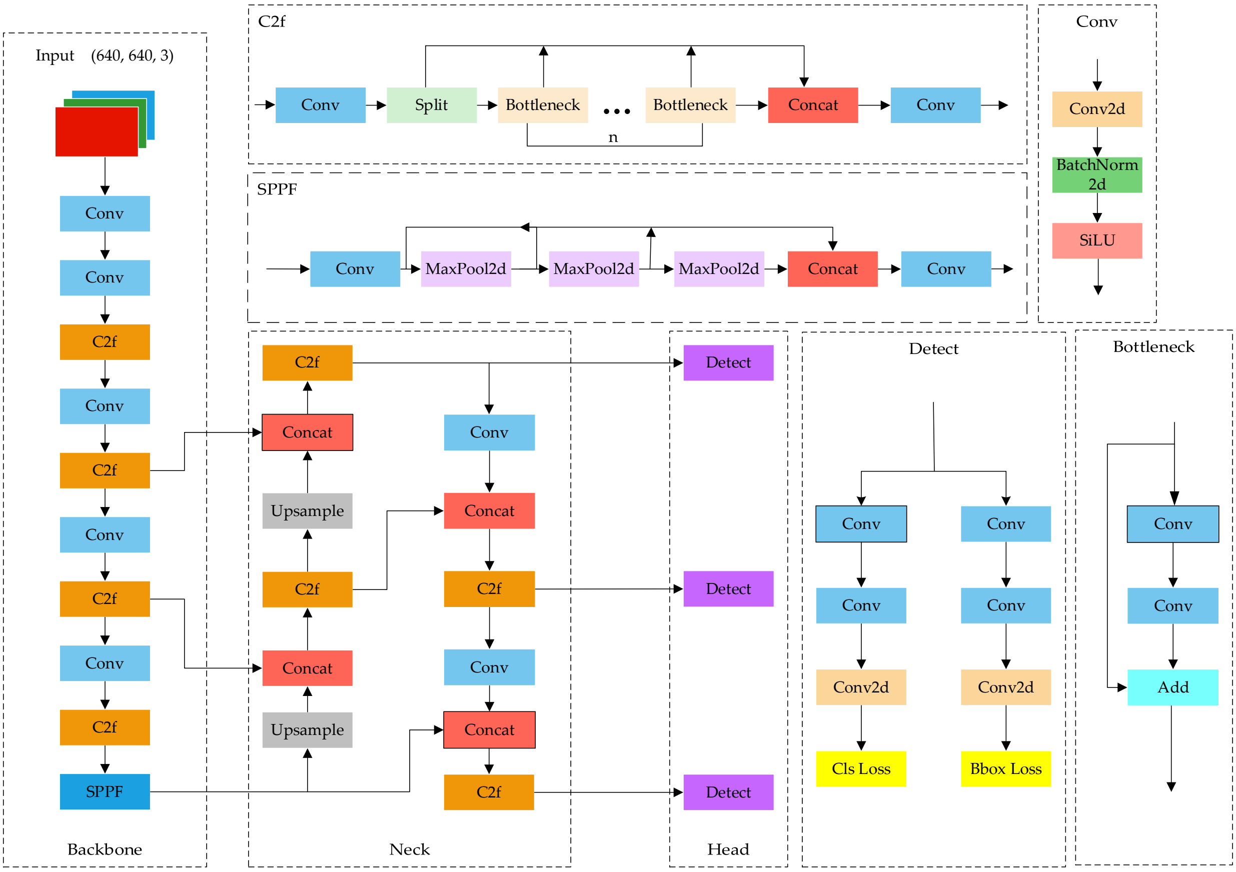


Figure 1. The structure of YOLOv8 model.

3. Method

3.1. YOLO-RAD Network Model

This paper proposes a pedestrian detection model, YOLO-RAD, tailored specifically for dense scenes to address the challenge of low detection accuracy caused by pedestrian occlusion and scale changes. Figure 2 shows the overall architecture of YOLO-RAD. Firstly, this paper introduces the concept of receptive field attention (RFA) and modifies the Conv and C2f modules in the YOLOv8 model. It adopts RFAConv and C2f_RFA modules to replace the original Conv and C2f modules, addressing the parameter sharing issue caused by the convolutional kernel and enhancing the feature extraction capability of the model. Secondly, this paper proposes a 4-layer adaptive spatial feature fusion (ASFF) module, which is added to the neck network. This module gradually fuses information between different feature layers, reducing conflicts in feature information during the fusion process and improving the model's ability to integrate feature information. Finally, to enhance the detection head, the small-target dynamic head structure (DyHead-S) is utilized to improve the overall detection performance of the model for small-scale pedestrians.

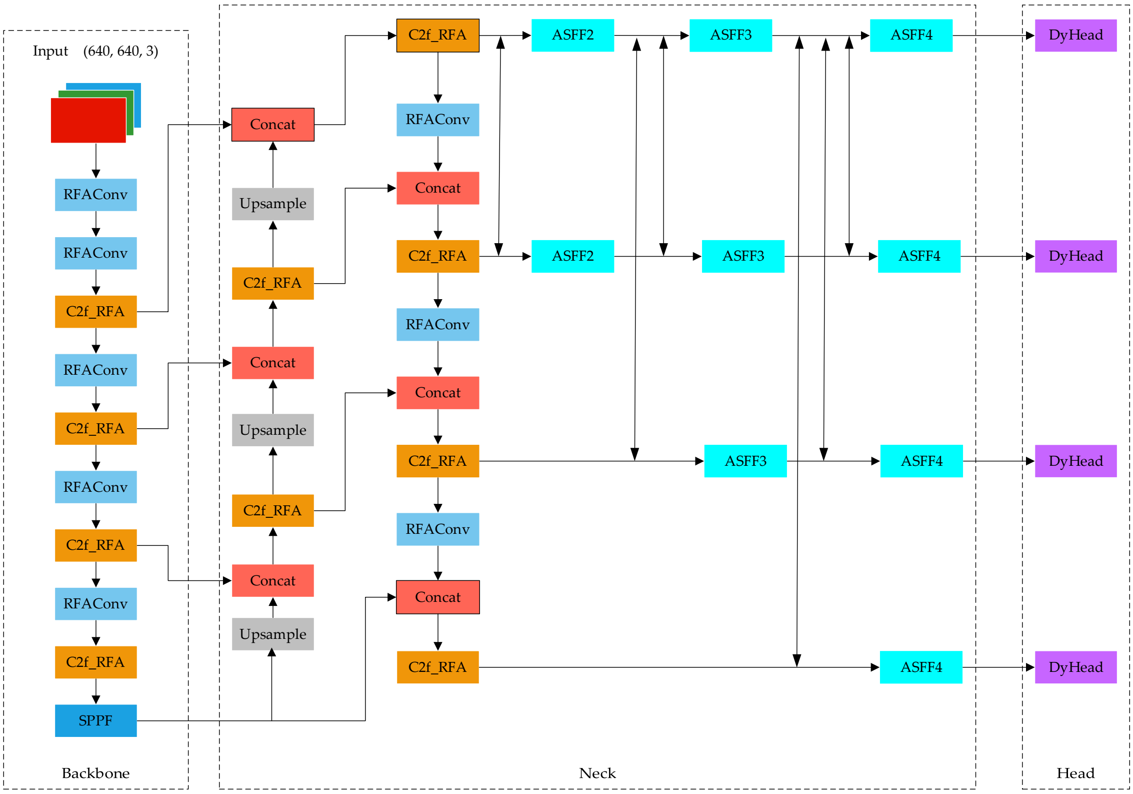


Figure 2. The structure of the YOLO-RAD model.

3.2. Receptive Field Attention

3.2.1. RFAConv Module

The Conv module is a fundamental component of the YOLO algorithm, responsible for extracting image features and performing object detection. However, with standard convolution operations, emphasis is placed on local connections and weight sharing. In other words, the convolution kernel's weight corresponds to the entire input graph. Since objects in different positions of the image vary in shape and size, the information at different positions differs. Standard convolution, with shared parameters, does not capture positional differences effectively. Consequently, the performance of convolutional neural networks is limited to some extent. The RFAConv module integrates standard convolution with receptive field attention, comprehensively addressing the issue of parameter sharing in convolution kernels. Additionally, it considers the importance of each feature in the receptive field.

The core idea of RFAConv is to integrate spatial attention with standard convolution, prioritizing the importance of different features within the receptive field. This allows the network to process local areas of the image more efficiently and enhance feature extraction accuracy. Moreover, RFAConv enables the network to identify and emphasize crucial regions of the input feature map, adjusting the convolutional kernel's weight accordingly. By doing so, the network can allocate computational resources more effectively, focusing on informative features while capturing a wide range of information. This enhances overall processing efficiency and network performance. Through this approach, the issue of convolution kernel parameter sharing is successfully addressed.

Using a 3×3 convolution kernel as an example, as illustrated in Figure 3, 'spatial features' represent the original feature map, while 'receptive field spatial features' are obtained by transforming spatial features using non-overlapping sliding windows. Each 3×3 window in the spatial feature space represents a receptive field slider when extracting features.

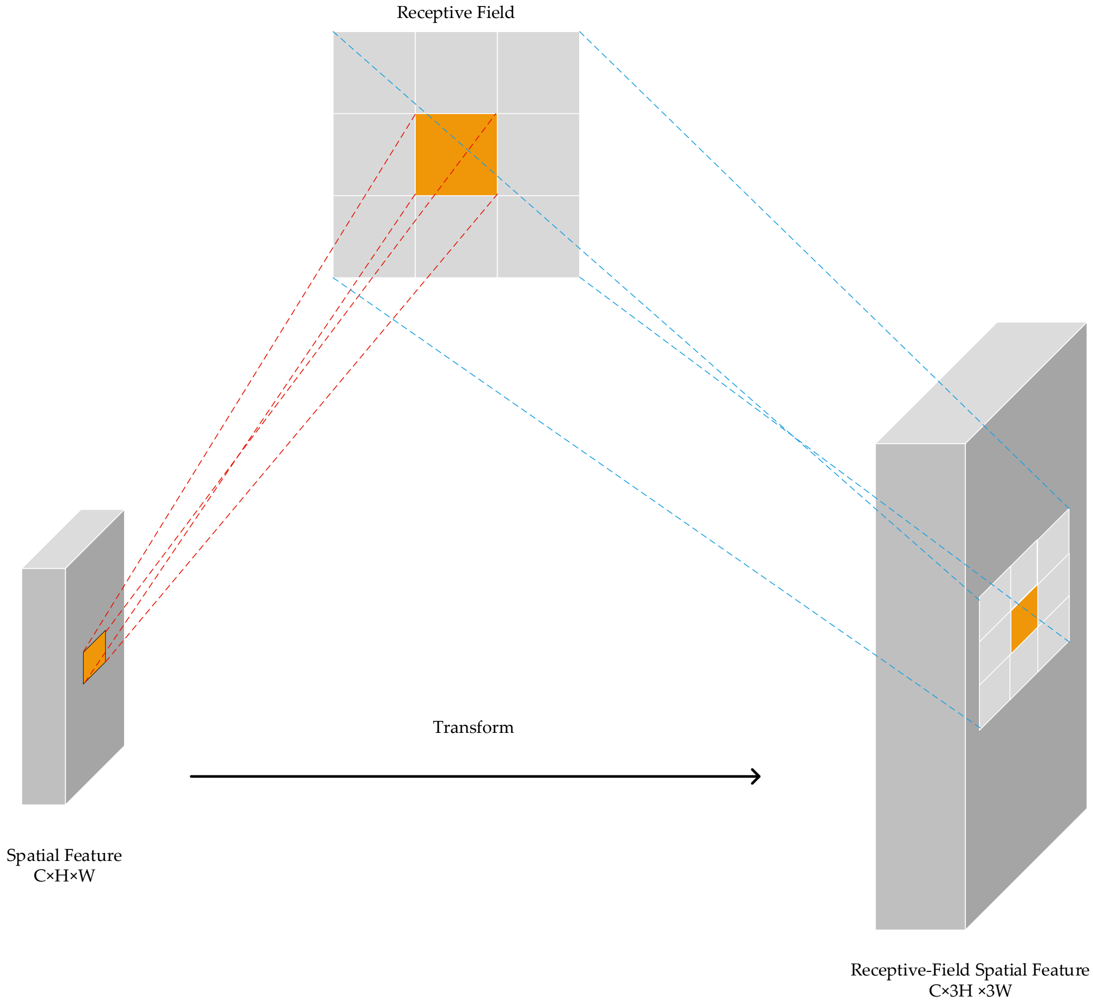


Figure 3. The receptive field spatial features are obtained by transforming the spatial features.

To generate dynamic unfolding features based on the size of the receptive field, group convolution (Group Conv) is employed. When a 3 × 3 convolutional kernel is used for feature extraction, each 3 × 3 window in the receptive field spatial features represents a receptive slider. After the receptive field features are extracted using group convolution, the original features are mapped to new features. However, interacting with each receptive field feature incurs additional computational overhead. To mitigate computational costs and reduce the number of parameters, global information for each receptive field feature is aggregated using AvgPool. Subsequently, information interaction occurs through 1 × 1 group convolution operations. Finally, Softmax is applied to emphasize the importance of each feature in the receptive field. The structure of RFAConv is depicted in Figure 4.

The calculation process of the receptive field attention convolution is depicted in Equation (2):

$$\mathrm{F} = \mathrm{Softmax}\left(\mathrm{g}^{1\times 1}(\mathrm{AvgPool}(\mathrm{X}))\right) \times \mathrm{ReLu}\left(\mathrm{Norm}\left(\mathrm{g}^{\mathrm{k}\times \mathrm{k}}(\mathrm{X})\right)\right) = \mathrm{A}_{\mathrm{rf}} \times \mathrm{F}_{\mathrm{rf}}, \quad (2)$$

Firstly, feature maps are processed through average pooling to aggregate global information for each receptive field feature. Subsequently, 1 × 1 convolutional layers are employed to exchange information, highlighting the significance of individual features within the receptive field through a normalization process. Within the receptive field sliding window, importance levels are assigned to different features, and spatial features within the receptive field are prioritized to ensure that the generated convolutional kernels can extract key features first, generating attention maps for subsequent convolutional kernel weight allocation. Then, the original feature map undergoes convolution to generate receptive field spatial features with the same size and dimensions as the attention map. Both attention and receptive field spatial features are computed through grouped convolution to reduce parameter and computational load in network operations. Finally, features are extracted from the receptive field spatial features based on the weights of the attention map and adjusted to an appropriate size to obtain the output of receptive field attention convolution. Here, $g^{1\times1}$ represents a group convolution with a size of i × i, k denotes the size of the convolution kernel, "Norm" signifies normalization, X represents the input feature map, and F is the result obtained by multiplying the attention map A_{rf} with the transformed receptive field spatial features F_{rf}.

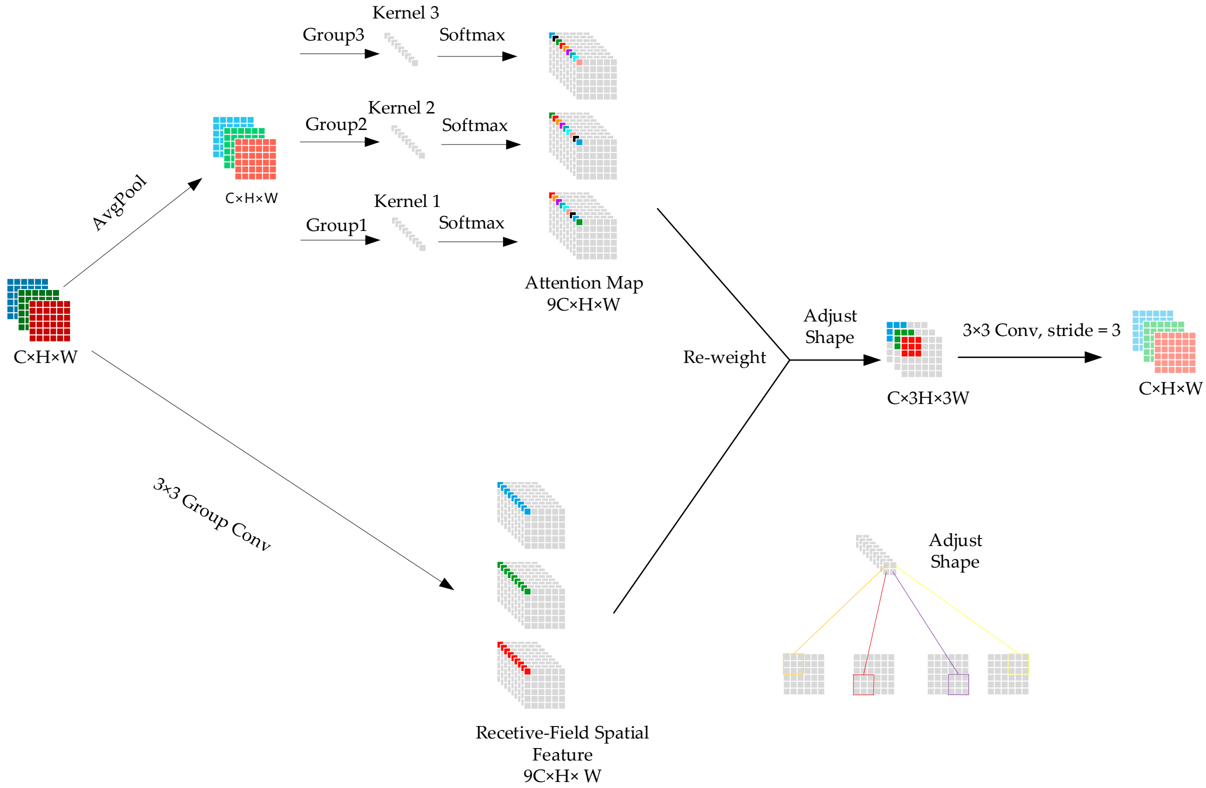


Figure 4. The structure of RFAConv.

3.2.2. C2f_RFA Module

This paper introduces a novel C2f_RFA module by incorporating the concept of RFA into the C2f module of YOLOv8. This module utilizes the backbone network to extract features from input images and integrates these features through the neck network, which plays a crucial role in comprehending the overall context of the image. To enhance the model's performance, we replaced the second convolution in the Bottleneck part of the C2f module with RFAConv. By integrating RFAConv into C2f, the model gains the ability to comprehensively handle features at various positions in the image, thereby improving its adaptability and effectiveness in addressing complex scenes. The modified structures of the Bottleneck and C2f are illustrated in Figures 5 and 6, respectively.

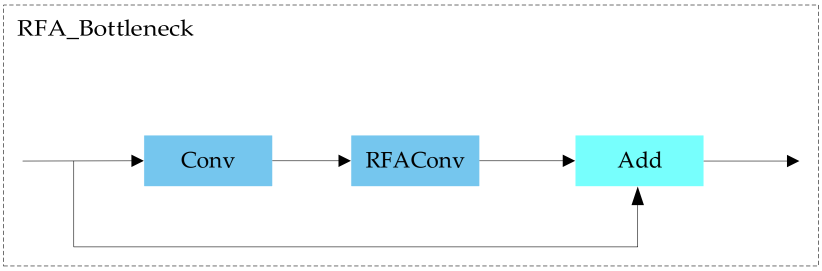


Figure 5. RFAConv replaces Standard Conv.

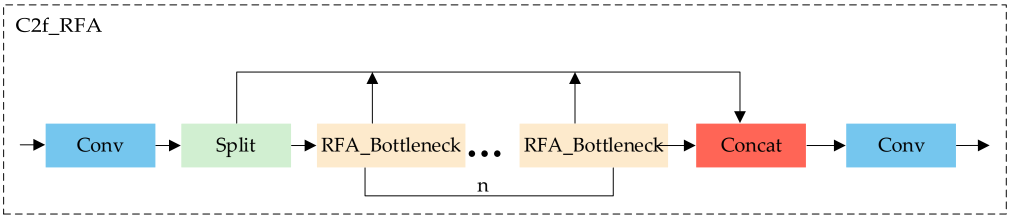


Figure 6. C2f_RFA structure based on RFA_Bottleneck.

3.3. Adaptive Spatial Feature Fusion Module

In the process of detecting pedestrians in crowded scenes, pedestrians exhibit a variety of gestures, such as standing, walking, and running, along with varying distances between pedestrians. This leads to changes in pedestrian posture in the image and the formation of targets of different scales. Adaptive spatial feature fusion (ASFF) networks utilize spatial filtering to suppress inconsistencies in spatial features at different scales during the fusion process, retaining only the information relevant for combination. By adaptively fusing different feature layers, ASFF effectively utilizes feature information of varying scales, significantly reducing the loss of target feature information. Additionally, we designed a 4-layer ASFF module and introduced a shallow feature layer in the backbone network to enrich the shallow feature information of pedestrians. In YOLOv8, we utilize ASFF to fuse feature information from four different feature layers, each with different resolutions and channel numbers. Features from other layers are first aligned to the same resolution and number of channels before being fused together, resolving potential conflicts in functional information between different levels. This adaptive fusion of features from different levels ensures that conflicting information is filtered out while retaining and emphasizing dominant features. The ASFF process is illustrated in Figure 7.

In the multi-level feature fusion process, ASFF fuses four different layers of feature information, requiring their dimensions to be adjusted to the same size initially. For each scale, both upsampling and downsampling need to be performed. For the upsampled part, the number of channels in the other layers is first adjusted to match the number of channels in the current layer using 1 × 1 convolutions. Subsequently, interpolation is applied to improve the resolution. Regarding downsampling, different convolutional operations are used depending on the downsampling factor. For 2× downsampling, a 2 × 2 convolution with a stride of 2 is utilized; for 4× downsampling, a 4 × 4 convolution with a stride of 4 is employed; and for 8× downsampling, an 8 × 8 convolution with a stride of 8 is used. The adaptive spatial feature fusion operation combines the features from different levels with their respective weights to create a fused feature vector at the desired level l. The process of ASFF fusing the four channel features is depicted in Equation (3):

$$y_{ij}^{l} = \alpha_{ij}^{l} \cdot x_{ij}^{1\to l} + \beta_{ij}^{l} \cdot x_{ij}^{2\to l} + \gamma_{ij}^{l} \cdot x_{ij}^{3\to l} + \delta_{ij}^{l} \cdot x_{ij}^{4\to l}, \quad (3)$$

Here, y_{ij}^{l} is the resulting feature vector at position (i, j) after the adaptive spatial fusion at level l. It is the combined feature vector from four different levels. $x_{ij}^{n\to l}$ are the feature vectors at position (i, j) from levels n to level l. They represent the low-level, mid-level, and potentially high-level features being fused. α_{ij}^{l}, β_{ij}^{l}, γ_{ij}^{l}, δ_{ij}^{l} are the spatial weights assigned to the features from the four levels at level l, and they satisfy $\alpha_{ij}^{l} + \beta_{ij}^{l} + \gamma_{ij}^{l} + \delta_{ij}^{l} = 1$, meaning they represent a linear combination of the input features.

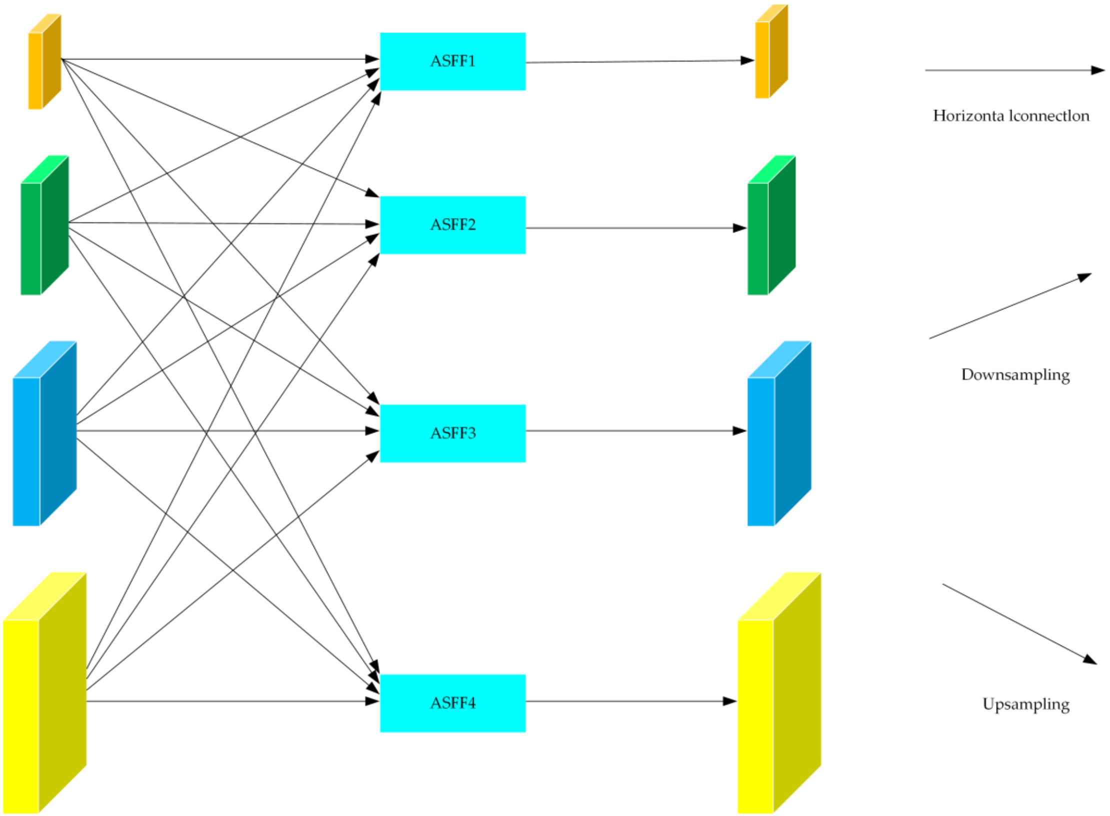


Figure 7. The process of ASFF.

3.4. Small-Target Dynamic Head Structure

In scenes with a large number of people, each pedestrian in the same image may exhibit different postures, scales, and positions. To address these challenges, the detection model's head must possess a certain degree of spatial perception. Therefore, we introduced a dynamic head (DyHead).

DyHead, as illustrated in Figure 8, operates by enhancing the feature map through a series of attention modules to improve object detection performance. First, the feature map is adjusted to the same scale, forming a tensor. This tensor then passes through three different attention modules sequentially: the scale perception module, the spatial perception module, and the task perception module. In the scale perception module (π_L), the L dimensions of the tensor are globally average pooled to capture average information of features at different levels. A 1 × 1 convolutional layer is then applied to extract this information and enhance nonlinearity through the ReLU activation function. The result is multiplied with the original feature map via a hard sigmoid activation function, producing a weight plot reflecting the importance of features at different scales. In the spatial awareness module (π_S), a 3 × 3 convolutional layer learns offset values and weights of the feature map. Deformable convolution is then employed to enable flexible focusing on key regions in the spatial dimension, enhancing the recognition of target shape and position. In the task awareness module (π_C), tensors are globally average pooled in L × S dimensions to reduce the dimension and computational burden. This information is further processed through two fully connected layers and a normalized layer. The ReLU operation dynamically adjusts the channel of the feature map according to different detection tasks (such as classification, frame regression), making the features of different tasks more prominent.

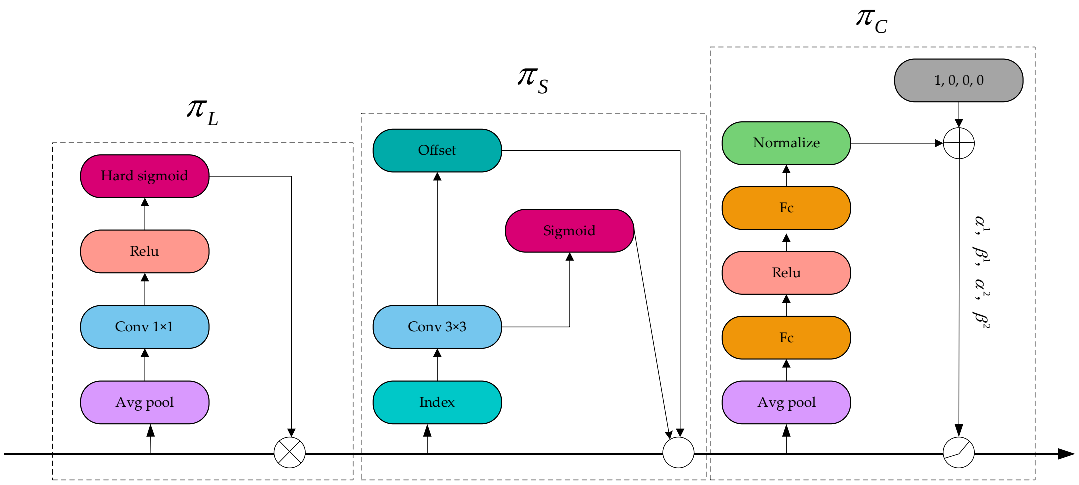


Figure 8. The detailed structure of DyHead.

To enhance DyHead's capability to detect small objects, we implemented a strategy by adding a new detection head on top of the dynamic head, specifically designed for detecting small-scale pedestrians, as depicted in Figure 9. This design incorporates a second layer of feature mapping into the overall feature fusion framework, thereby preserving shallower semantic information crucial for identifying small-scale pedestrians. To achieve this, we introduced an additional feature map during the feature extraction phase, with a size of 160 × 160 pixels, aimed at capturing more detailed information about small-scale pedestrians. To ensure the effective fusion of the new feature map with other feature maps, we first upsampled and then downsampled it. These modifications increased the number of dynamic heads from three to four, significantly enhancing the perception and sensitivity of dynamic heads to small targets. Hence, we named this head structure the small-target dynamic head structure (DyHead-S).

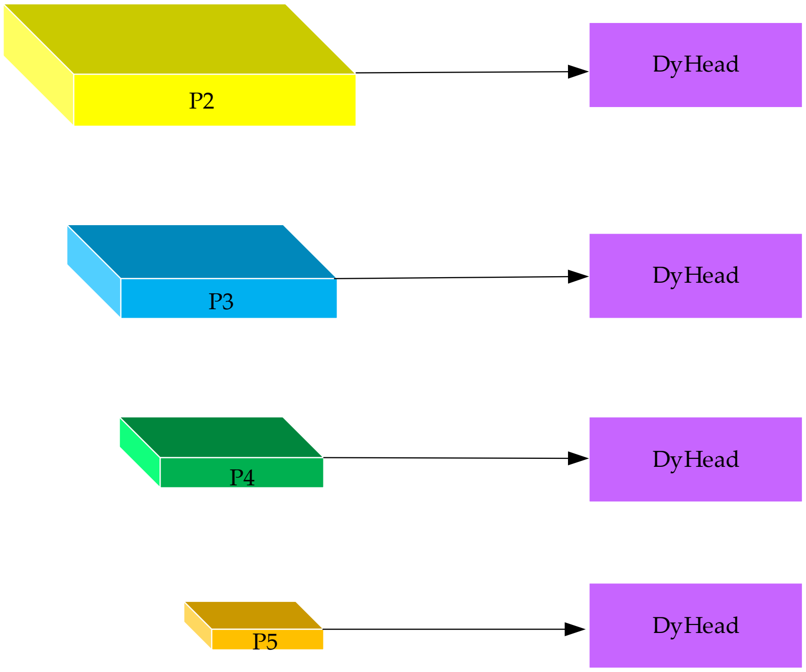


Figure 9. The framework of DyHead-S.

4. Experiment

4.1. Dataset Introduction

The dataset utilized in this study comprises the publicly available WiderPerson dataset [38] and the CrowdHuman dataset [39], which were employed to validate the algorithm's robustness. As separate annotation files for a test set were not provided in the WiderPerson and CrowdHuman datasets, the original training set was divided into training and validation sets at an 8:2 ratio, with the original validation set serving as the test set.

The WiderPerson dataset functions as a benchmark dataset tailored specifically for pedestrian detection in crowded scenes. It consists of 13,382 images collected from various scenes and is annotated with approximately 400,000 occlusion labels. The dataset originally comprised five categories: "pedestrians," "riders," "partially-visible persons," "ignore regions," and "crowd." Given the focus of this study on dense pedestrian detection, the categories "ignore regions" and "crowd" were deemed unnecessary for practical applications. Furthermore, the "crowd" category is annotated with large bounding boxes in the dataset. As a result, these two labels were omitted, and the remaining categories ("pedestrians," "riders," "partially-visible persons") were amalgamated into a single category, termed "pedestrians."

The CrowdHuman dataset, developed by SenseTime, constitutes a substantial collection of crowd images captured in real-world environments like streets and parks. It encompasses a total of 24,370 images, with around 470,000 pedestrian instances spread across both the training and validation sets. On average, each image contains 23 pedestrian instances, showcasing a notable presence of multi-scale and occluded targets. This dataset proves especially beneficial for examining and tackling pedestrian detection challenges, particularly under occlusion conditions.

4.2. Evaluation Indicators

This paper evaluates the model's detection performance using commonly used metrics such as precision, recall, and mean average precision.

Precision, denoted as P, represents the ratio of correctly detected positive samples to all samples predicted as positive by the model, as indicated in Equation (4). Recall, denoted as R, represents the ratio of correctly detected positive samples to all true positive samples, as illustrated in Equation (5).

$$\mathrm{P} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{4}$$

$$\mathrm{R} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{5}$$

where TP represents the correctly detected positive samples by the model, FP represents the negative samples mistakenly labeled as positive by the model, and FN represents the positive samples missed by the model.

Mean average precision, abbreviated as mAP, refers to the average of AP across all classes by the model. A higher mAP indicates better overall performance of the model across the entire dataset, as illustrated in Equation (6).

$$\mathrm{mAP} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{AP_i}}{\mathrm{k}}, \tag{6}$$

where AP is a metric obtained by averaging the precision–recall curve of the model for a specific class, with k representing the number of classes.

4.3. Experimental Environment and Parameter Configuration

The experiments in this paper were conducted on the Ubuntu operating system, and the parameters used are listed in Table 1.

Table 1. Experimental environment parameter configuration.

Type	Name	Configuration
Software	OS	Ubuntu20.04
	Python	3.8
	CUDA	11.8
	Pytorch	2.0.0
Hardware	CPU	Intel(R) Xeon(R) Gold 6430
	GPU	GeForce RTX 4090
Parameter	Image size	640 × 640
	Epochs	300
	Batch size	8
	Optimizer	SGD
	Learning rate	0.01
	Momentum	0.937
	Weight decay	0.0005

4.4. Comparison Experiment

In this study, we compared our model to other detection models on the same dataset, including YOLOv5n, YOLOv6n, YOLOv7-tiny, YOLOv8n, RT-DETR-l, YOLO-World-n, and RTMDet-tiny. All models were trained for 300 epochs, and none used pre-training weights. The training results of each model are shown in Table 2.

Table 2. Results of different models.

Model	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%
YOLOv5n	79.4	61.9	72.8	43.9
YOLOv6n	80.5	65.8	76	48.2
YOLOv7-tiny	81.6	65.9	76.3	45.5
YOLOv8n	81.5	65.3	76.4	48.1
YOLO-World-n [40]	81.6	65.3	76.6	48.5
RT-DETR-l [41]	80.4	62.7	73.9	44.3
RTMDet-tiny	-	-	74	47.7
YOLO-RAD	81.5	67.9	78.9	50.8

In Table 2, compared with other models, the YOLO-World model has the highest accuracy, with P, mAP@0.5, and mAP@0.5:0.95 reaching 81.6%, 76.6%, and 48.5%, respectively. In addition, it can be seen from the table that the accuracy rate of our proposed model reaches 81.5%, although the accuracy is not the highest, only 0.1% lower than that of YOLOv7-tiny and YOLO-World models, our model recall rate is the best, reaching 67.9%. The significant increase in the recall rate of the YOLO-RAD model indicates that the algorithm's ability to detect target objects, capture more true positives, and reduce missed tests has been enhanced. This high recall rate is critical in practical applications such as safety monitoring and disease diagnosis, where sensitivity is essential to minimize false negatives. It is worth noting that our model also achieved the best performance in mAP@0.5 and mAP@0.5:0.95, reaching 78.9% and 50.8%, respectively. Compared to YOLO-World, our algorithm improves by 2.3% in both mAP@0.5 and mAP@0.5:0.95.

4.5. Ablation Experiment

To evaluate the impact of each enhancement module proposed in YOLO-RAD, we conducted an ablation study with consistent parameters and training procedures. The study was performed using the WiderPerson dataset, and the results are summarized in Table 3, where "$\surd$" denotes the utilization of the corresponding module.

Table 3. Results of ablation experiment.

Model	RFA	ASFF	DyHead-S	P/%	R/%	mAP@0.5/%	mAP@0.5:0.95/%
YOLOv8n				81.5	65.3	76.4	48.1
	√			81.5	66.3	77.1	48.9
		√		81.1	67.6	77.8	49.5
			√	80.4	67.1	78	49.8
	√	√		81.7	67.4	78	49.7
	√		√	80.9	67.8	78.5	50.3
		√	√	81.4	67.7	78.6	50.5
	√	√	√	81.5	67.9	78.9	50.8

In this study, we conducted a series of ablation experiments on the benchmark model YOLOv8n to evaluate the effect of adding different modules on pedestrian detection performance. The accuracy index values of the initial benchmark model are (P) 81.5%, (R) 65.3%, (mAP@0.5) 76.4%, and (mAP@0.5:0.95) 48.1%. We added RFA, ASFF, and DyHead-S, respectively. The results show that after RFA was added alone, the accuracy of the model was slightly improved; P is 81.5%, R is 66.3%, mAP@0.5 is 77.1%, and mAP@0.5:0.95 is 48.9%. When ASFF was added alone, the recall rate of the model increased significantly, with R reaching 67.6%, while mAP@0.5 and mAP@0.5:0.95 also increased significantly, reaching 77.8% and 49.5%, respectively. After the addition of DyHead-S alone, the model achieved a certain degree of improvement in R and mAP@0.5 indices and especially improved in mAP@0.5:0.95, which reached 49.8%. After further combination experiments, we observed that the model improved in P, R, mAP@0.5, and mAP@0.5:0.95. In particular, when RFA, ASFF, and DyHead-S were introduced at the same time, the comprehensive performance of the model reached the best level; P was 81.5%, R was 67.9%, mAP@0.5 was 78.9%, and mAP@0.5:0.95 was 50.8%. These results show that the proposed YOLO-RAD model achieves excellent performance in pedestrian detection tasks.

Figure 10 shows the curves of accuracy rate, recall rate, mAP@0.5 and mAP@0.95 after the addition of RFA, ASFF, and DyHead-S and compares them with the basic model YOLOv8n. It can be clearly seen that the three independent improvements are all effective.

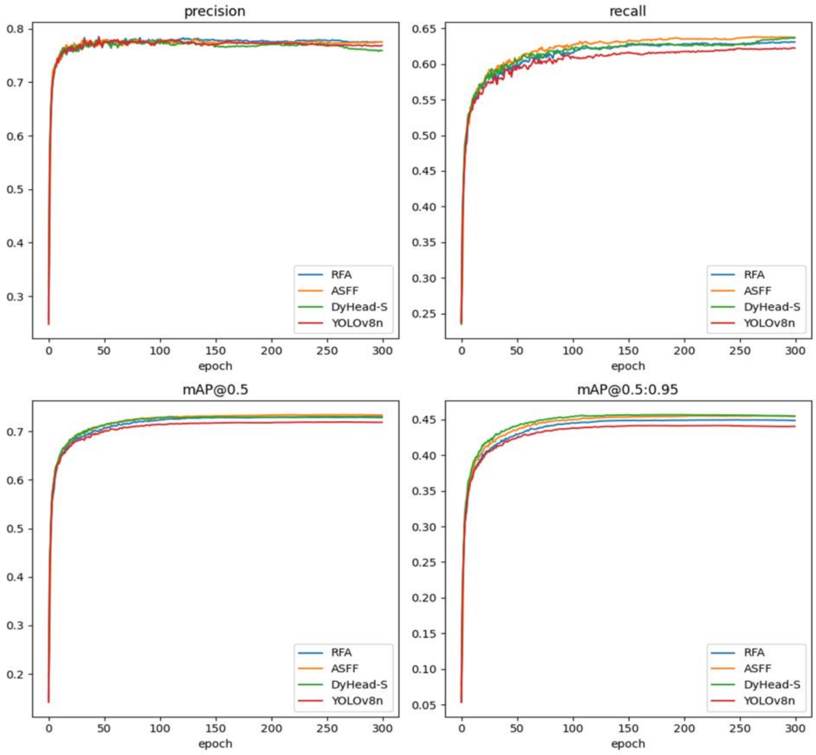


Figure 10. A visual comparison of three improvement points: precision, recall, mAP@0.5, and mAP@0.5:0.95.

The PR curves and mAP of YOLOv8n and YOLO-RAD are visualized in Figures 11 and 12, respectively. From this, we can see that the accuracy of YOLO-RAD is higher than that of the basic model. With the increase in the number of training epochs, the mAP@0.5 and mAP@0.95 of the improved algorithm are gradually improved compared with the original YOLOv8n algorithm.

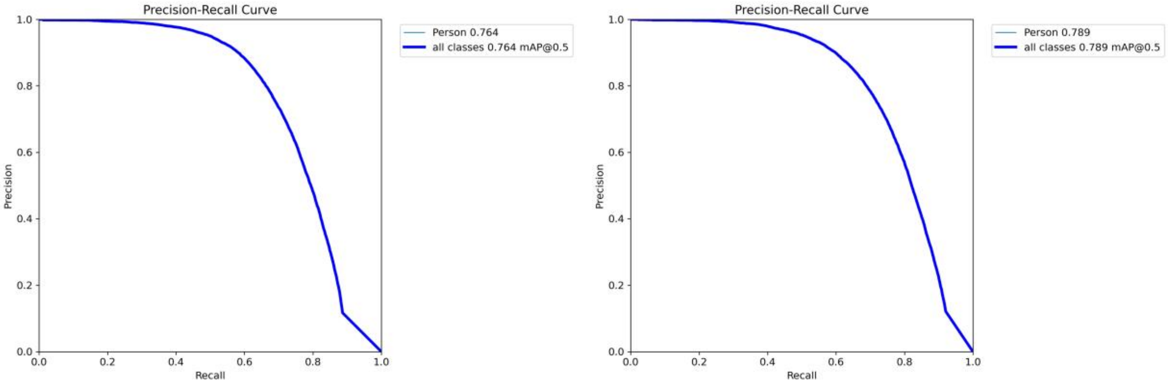


Figure 11. Comparison of PR curves before and after improvement.

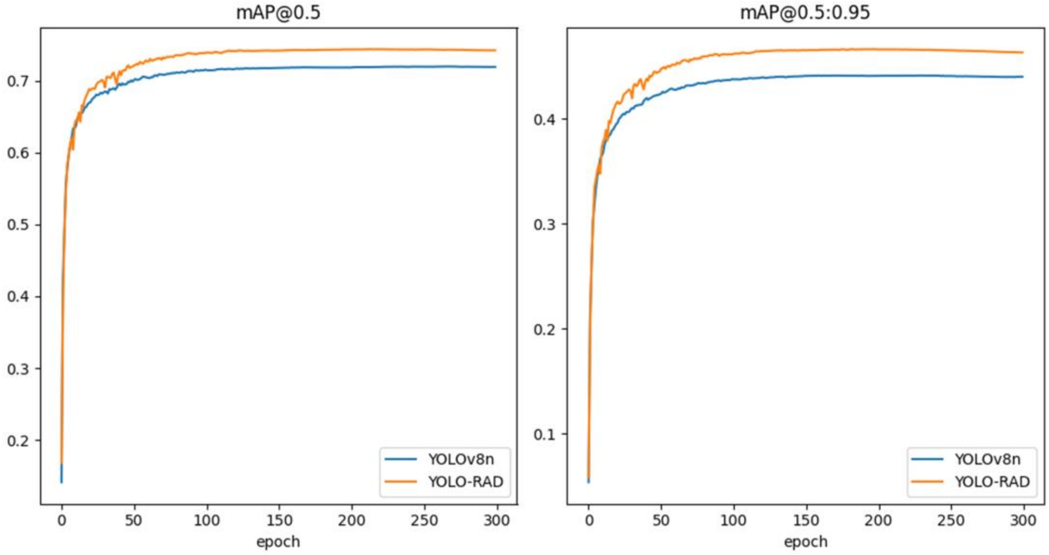


Figure 12. Comparison of mAP@0.5 and mAP@0.95 curves before and after improvement.

4.6. Validation on Other Datasets

To assess the generality and robustness of the proposed method, we chose the CrowdHuman public dataset for validation. The validation results are presented in Table 4.

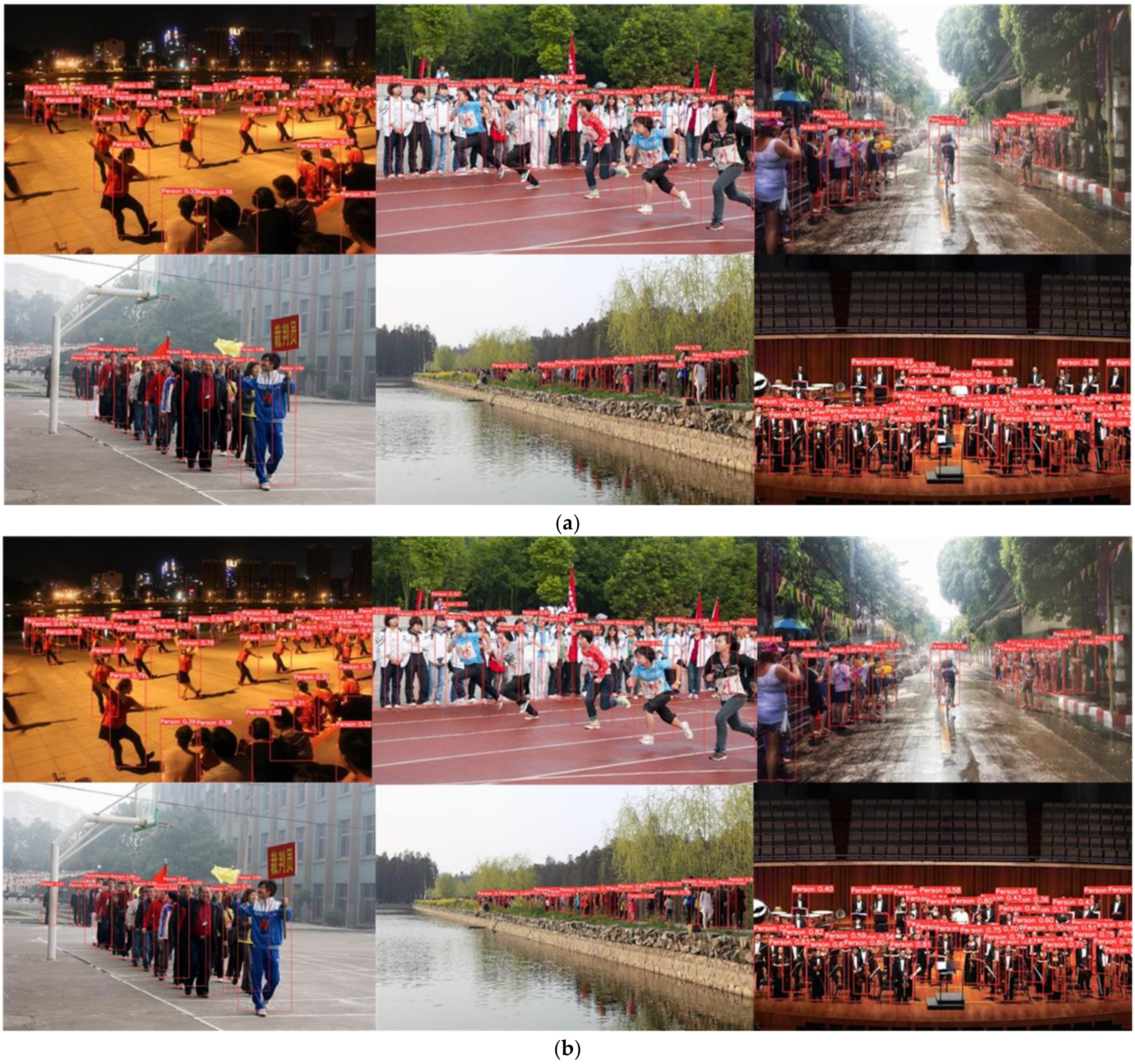
Table 4. Results of CrowdHuman experiment.

Class	P/% YOLOv8n/Our	R/% YOLOv8n/Our	mAP@0.5/% YOLOv8n/Our	mAP@0.5:0.95 YOLOv8n/Our
All	84.4/85	69/75.3	77.9/83.9	48.3/55.1
Head	86.3/86.3	70.8/77.6	78.5/85	48/54.8
Person	82.5/83.7	67.2/73	77.3/82.8	48.5/55.4

We conducted detailed comparative experiments on the CrowdHuman dataset with the original YOLOv8n, comparing their performance on various categories and overall recognition results. From these experimental results, we can clearly see the significant advantages of the YOLO-RAD model in various indicators. Specifically, compared to the YOLOv8n, the YOLO-RAD improved by 0.6%, 6.3%, 6%, and 6.8% in the four evaluation indicators, respectively. We further found that in the Head label category, recall rates increased by 6.8%, while mAP@0.5 and mAP@0.5:0.95 saw increases of 6.5% and 6.8%, respectively. In the Person label category, we observed a 1.2% increase in accuracy, a 5.8% increase in recall, a 5.5% increase in mAP@0.5, and a 6.9% increase in mAP@0.5:0.95. These results clearly show that our proposed YOLO-RAD method shows significant detection performance advantages on different datasets.

4.7. Detection Performance

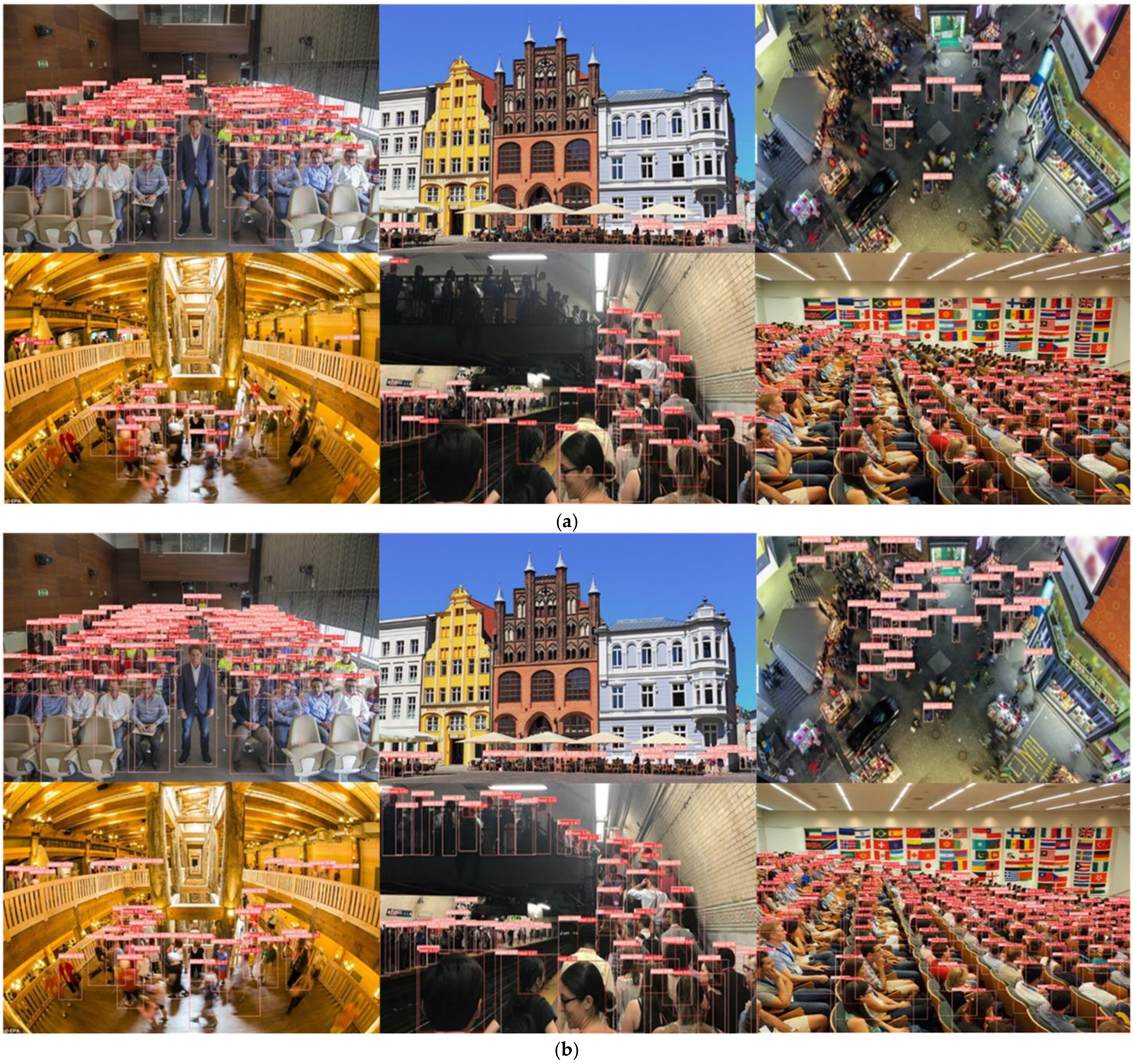
To visually demonstrate the superiority of the proposed algorithm in recognizing pedestrians in dense environments, we conducted detections using YOLOv8n and YOLO-RAD on images from the WiderPerson and CrowdHuman datasets. The detection performance is illustrated in Figures 13 and 14.



(**a**)

(**b**)

Figure 13. Comparison of detection results on WiderPerson: (**a**) the detection performance of YOLOv8n; (**b**) the detection performance of YOLO-RAD.



(**a**)

(**b**)

Figure 14. Comparison of detection results on CrowdHuman: (**a**) the detection performance of YOLOv8n; (**b**) the detection performance of YOLO-RAD.

5. Conclusions

This paper introduces an improved pedestrian detection model that can significantly improve pedestrian detection performance in dense scenes. The main purpose is to solve some problems of the existing object detection models for pedestrian detection in dense scenes, especially those of occlusion and small scale. The YOLO-RAD model in this paper first combines the Conv module and C2f module with receptive field attention (RFA) to solve the parameter sharing problem of the convolution kernel and significantly improve the ability of the network to extract pedestrian features. Secondly, in order to better fuse the feature information of different feature layers, a four-layer adaptive spatial feature fusion (ASFF) module is designed to reduce the feature information conflicts between different feature layers in the fusion process. Finally, in order to enhance the detection performance of small-scale pedestrians, we propose a small-target dynamic head structure (DyHead-S) to improve the detection head of the model and improve the ability of the model to detect small-scale pedestrians. By combining these methods, the model can better adapt to the pedestrian detection task in dense scenes, so as to improve the detection performance of the whole model. To evaluate the validity of the proposed model, two widely recognized public datasets were used to conduct experimental studies, and the experimental results clearly demonstrate the superiority of the proposed YOLO-RAD model over existing methods. Not only did it show excellent performance in identifying crowded pedestrians, but it also significantly improved recall rates. In addition, we also consider the practicability of the proposed method. With the increase in urbanized population and the advancement of security technology, the demand for pedestrian detection in urban monitoring is rising. Our YOLO-RAD model exhibits good portability and detection efficacy, making it capable of running efficiently on devices such as monitoring systems. Therefore, we believe that the model holds broad application prospects and can play a significant role in urban security, intelligent transportation, and other fields. In future studies, we will continue to optimize the performance and efficiency of this model to further enhance its practical value in real-world scenarios.

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

Funding: This research was funded by the Jilin Provincial Department of Science and Technology, grant numbers 20240302076GX.

Data Availability Statement: Data are contained within the article.

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

References

1. Harris, C.; Stephens, M. A combined corner and edge detector. *Alvey Vision Conference* **1988**, *15*, 23.1–23.6.
2. Dalal, N.; Bill, T. 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; Volume 1, pp. 886–893.
3. Cosma, C.; Brehar, R.; Nedevschi, S. Pedestrians detection using a cascade of LBP and HOG classifiers. In Proceedings of the 2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 5–7 September 2013; pp. 69–75.
4. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
5. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: New York, NY, USA, 2014; pp. 580–587.
6. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; IEEE: New York, NY, USA, 2015; pp. 1440–1448.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1137–1149. [CrossRef] [PubMed]
8. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: New York, NY, USA, 2017; pp. 2961–2969.
9. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
10. 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.
11. 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, 27–30 June 2016; IEEE: New York, NY, USA, 2016; pp. 779–788.
12. 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; IEEE: New York, NY, USA, 2017; pp. 7263–7271.
13. Redmon, J.; Ali, F. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Bochkovskiy, A.; Wang, C.Y.; Liao HY, M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.

15. YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 12 April 2021).
16. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Wei, X. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
17. 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, 17–24 June 2023; IEEE: New York, NY, USA, 2023; pp. 7464–7475.
18. 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; IEEE: New York, NY, USA, 2017; pp. 2980–2988.
19. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
20. Zhang, X.; Liu, C.; Yang, D.; Song, T.; Ye, Y.; Li, K.; Song, Y. Rfaconv: Innovating spatial attention and standard convolutional operation. *arXiv* **2023**, arXiv:2304.03198.
21. Yang, G.; Lei, J.; Zhu, Z.; Cheng, S.; Feng, Z.; Liang, R. AFPN: Asymptotic feature pyramid network for object detection. In Proceedings of the 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Honolulu, HI, USA, 1–4 October 2023; pp. 2184–2189.
22. Dai, X.; Chen, Y.; Xiao, B.; Chen, D.; Liu, M.; Yuan, L.; Zhang, L. Dynamic head: Unifying object detection heads with attentions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; IEEE: New York, NY, USA, 2021; pp. 7373–7382.
23. Yang, X.; Liu, Q. Scale-sensitive feature reassembly network for pedestrian detection. *Sensors* **2021**, *21*, 4189. [CrossRef]
24. Ma, J.; Wan, H.; Wang, J.; Xia, H.; Bai, C. An improved one-stage pedestrian detection method based on multi-scale attention feature extraction. *J. Real-Time Image Process.* **2021**, *18*, 1965–1978. [CrossRef]
25. Yan, C.; Zhang, H.; Li, X.; Yuan, D. R-SSD: Refined single shot multibox detector for pedestrian detection. *Appl. Intell.* **2022**, *52*, 10430–10447. [CrossRef]
26. Yang, R.; Wang, Y.; Xu, Y.; Qiu, L.; Li, Q. Pedestrian detection under parallel feature fusion based on choquet integral. *Symmetry* **2021**, *13*, 250. [CrossRef]
27. Murthy, C.B.; Hashmi, M.F.; Muhammad, G.; AlQahtani, S.A. AlQahtani. YOLOv2PD: An efficient pedestrian detection algorithm using improved YOLOv2 model. *Comput. Mater. Contin.* **2021**, *69*, 3015–3031.
28. Liu, Z.; Song, X.; Feng, Z.; Xu, T.; Wu, X.; Kittler, J. Global context-aware feature extraction and visible feature enhancement for occlusion-invariant pedestrian detection in crowded scenes. *Neural Process. Lett.* **2023**, *55*, 803–817. [CrossRef]
29. Qin, Y.; Qian, Y.; Wei, H.; Fan, Y.; Feng, P. FE-CSP: A fast and efficient pedestrian detector with center and scale prediction. *J. Supercomput.* **2023**, *79*, 4084–4104. [CrossRef]
30. He, Y.; Zhu, C.; Yin, X.C. Occluded pedestrian detection via distribution-based mutual-supervised feature learning. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 10514–10529. [CrossRef]
31. YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 10 January 2023).
32. 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; IEEE: New York, NY, USA, 2017; pp. 2117–2125.
33. Li, H.; Xiong, P.; An, J.; Wang, L. Pyramid attention network for semantic segmentation. *arXiv* **2018**, arXiv:1805.10180.
34. Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Yang, J. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21002–21012.
35. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
36. Feng, C.; Zhong, Y.; Gao, Y.; Scott, M.R.; Huang, W. Tood: Task-aligned one-stage object detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, Washington, DC, USA, 11–17 October 2021; pp. 3490–3499.
37. Lyu, C.; Zhang, W.; Huang, H.; Zhou, Y.; Wang, Y.; Liu, Y.; Chen, K. Rtmdet: An empirical study of designing real-time object detectors. *arXiv* **2022**, arXiv:2212.07784.
38. 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]
39. Shao, S.; Zhao, Z.; Li, B.; Xiao, T.; Yu, G.; Zhang, X.; Sun, J. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv* **2018**, arXiv:1805.00123.
40. Cheng, T.; Song, L.; Ge, Y.; Liu, W.; Wang, X.; Shan, Y. YOLO-World: Real-Time Open-Vocabulary Object Detection. *arXiv* **2024**, arXiv:2401.17270.
41. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Chen, J. Detrs beat yolos on real-time object detection. *arXiv* **2023**, arXiv:2304.08069.

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.