applied sciences

MDPI

Article

Adaptive Feature Fusion for Small Object Detection

Qi Zhang [1,2], Hongying Zhang [1,2,*] and Xiuwen Lu [1,2]

[1] School of Information Engineering, Southwest University of Science and Technology, Mianyang 621000, China
[2] Sichuan Provincial Key Laboratory of Robotics for Special Environments, Southwest University of Science and Technology, Mianyang 621000, China
* Correspondence: zhywyd@163.com

Abstract: In order to alleviate the situation that small objects are prone to missed detection and false detection in natural scenes, this paper proposed a small object detection algorithm for adaptive feature fusion, referred to as MMF-YOLO. First, aiming at the problem that small object pixels are easy to lose, a multi-branch cross-scale feature fusion module with fusion factor was proposed, where each fusion path has an adaptive fusion factor, which can allow the network to independently adjust the importance of features according to the learned weights. Then, aiming at the problem that small objects are similar to background information and small objects overlap in complex scenes, the M-CBAM attention mechanism was proposed, which was added to the feature reinforcement extraction module to reduce feature redundancy. Finally, in light of the problem of small object size and large size span, the size of the object detection head was modified to adapt to the small object size. Experiments on the VisDrone2019 dataset showed that the mAP of the proposed algorithm could reach 42.23%, and the parameter quantity was only 29.33 MB, which is 9.13% ± 0.07% higher than the benchmark network mAP, and the network model was reduced by 5.22 MB.

Keywords: multi-scale feature fusion; adaptive fusion factor; attention mechanism; small object detection

Citation: Zhang, Q.; Zhang, H.; Lu, X. Adaptive Feature Fusion for Small Object Detection. *Appl. Sci.* **2022**, *12*, 11854. https://doi.org/10.3390/app122211854

Academic Editor: Amerigo Capria

Received: 27 October 2022
Accepted: 17 November 2022
Published: 21 November 2022

1. Introduction

As an important research topic in the field of computer vision, visual object detection technology has a wide range of applications in the fields of swarm intelligence, security surveillance, and the modern military [1]. Among many vision tasks, aerial detection is of great significance for national defense and civil use. Compared with large- and medium-sized objects whose detection accuracy has been improved to a whole new level, small objects have the characteristics of weak features and less information, especially in satellite remote sensing images and UAV free-angle shooting images, which are taken from similar backgrounds or images. It is very difficult to distinguish between adjacent objects, and it is more challenging when faced with complex environments such as low illumination and shadow occlusion.

The rapid development of deep learning and graphics processor technology has brought the performance of object detection algorithms to a new height. At present, object detection algorithms based on deep learning are mainly divided into two-stage and single-stage. The two-stage detection algorithm first generates a large number of candidate regions on the image, and then adjusts the position and range of the object region and classifies the object on this basis. Typical algorithms include Fast R-CNN [2], Faster R-CNN [3], Mask R-CNN [4], and other R-CNN series algorithms. The single-stage detection algorithm omits the generation of a priori frame, and directly generates object category probability and prediction frame coordinate information through regression analysis, mainly YOLO [5], SSD [6], YOLO9000 [7], YOLOv3 [8], YOLOv4 [9], and detection algorithms such as RetinaNet [10].

In object detection, it is generally considered that the size of the object is a small object when it is small relative to the size of the original image, and the existing small object definition methods are mainly divided into absolute size definition and relative size definition [11]. Taking the definition of COCO [12] objects, a common dataset in the field of object detection, COCO object definition, the object below 32 × 32 pixels in the figure is a small object. The relative size definition defines a small object from the relative ratio of the object to the image, and objects that are less than 10% of the image size or more are small objects [13]. In this paper, according to the definition of small objects in the general dataset COCO, objects smaller than 32 × 32 pixels are defined as small objects.

Because a small object has few pixels, its feature expression ability is weak, and there is less representative feature information. In order to enhance the expression ability of small object features, [14] adopted a top-up feature pyramid FPN feature fusion strategy to fuse deep features with shallow features, which increased the possibility of object detection, but also increased the background information. Hu J et al. [15] proposed the Squeeze and Excite (SE) block, which dynamically refines the features and significantly enhances the feature representation ability, but only focuses on enhancing the channel information of the features. R2-CNN [16] introduced the attention model to reduce the influence of complex background on small object detection and the false alarm rate, but due to the lightweight of its backbone network, the feature extraction ability was reduced. Zhang Yin et al. [17] proposed a feature enhancement module (FEM) for the problems of the low amount of effective information for feature extraction of the backbone network and the weak ability of feature map information representation. In this way, the multiple receptive field features in the lower-level feature map are fused, which enhances the feature extraction ability, but at the same time, brings a lot of feature redundancy.

It can be seen from the above that the detection of small objects in natural images faces the following challenges:

- The small object has few pixels, and the object scale spans large;
- The expression ability of small object features is weak, and the detection accuracy is low;
- The background image of the small object is complex, the background information is highly similar to the small object, and it is easily disturbed by the background.

In response to the above challenges, we make the following contributions and propose a small object detection algorithm combining multi-branch multi-scale feature fusion with the fusion factor and an improved attention mechanism, referred to as MMF-YOLO.

1. For the problem of large scale changes of small objects like objects, we designed a multi-scale, multi-path, and multi-flow feature fusion module, referred to as MMF-Net. Among them, the features of multi-branch fusion have unique fusion factors.
2. In order to focus the "gaze" of the network on the representative local object region after convolution and accurately separate the background information from the object information, a new attention mechanism (M-CBAM) was designed, which was added to the feature strengthening extraction to increase attention to the features to be extracted, so that the convolution can pay attention to the feature map with less sample information.
3. In order to improve the classification probability and regression coefficient of small objects, a shallow object detector was added for the small size of small objects. Simultaneously, in order to reduce the computational load of the network, the detection head in the deepest layer of the network was discarded. At the same time, in order to ensure the correct detection of larger-sized objects in small objects, shallower and deeper detection heads were reserved.

2. Materials and Methods

2.1. Materials

Computer vision has been a very active research direction since entering the 21st century. With the continuous innovation of image acquisition equipment, the explosive growth of visual information production, the continuous improvement of machine computing power, and the introduction of deep neural network model, the image processing technology in the field of vision is changing with each passing day, and the applicable field scenes are constantly expanding [18]. There are four mainstream tasks in computer vision: image classification, object detection, object identification, and image segmentation, and its technology is widely used in video analysis, remote sensing imaging [19], security monitoring [20], and medical [21] fields. Some of the latest research on computer vision in the field of deep learning is introduced below.

Aiming at recognizing small proportion, blurred, and complex traffic signs in natural scenes, Liu S et al. [22] proposed a traffic sign detection method based on RetinaNet-NeXt and adopted ResNeXt to improve the detection accuracy and effectiveness of RetinaNet, transfer learning and group normalization to accelerate network training, and improve the accuracy and recall of traffic sign detection. Sun X et al. [23] proposed a unified part-based convolutional neural network (PBNet) that was specifically designed for composite object detection in remote sensing images. A context refinement module was designed to generate more discriminative features by aggregating local and global context information, which enhanced the learning of part information and the ability of feature representation. A multi-mode medical image fusion with deep learning [24] will be proposed, according to the characters of multi-modal medical imaging, medical diagnostic technology, and practical implementation.

2.1.1. Small Object Detection

The research on object detection in the field of deep learning has been extensively studied, and has good effects and influence. With the development of deep learning, the detection of small objects has gradually begun. The authors of [25] proposed a scale matching method to align object scales between two datasets to obtain favorable representations of tiny objects. Although it effectively improved the detection performance of small objects, it required a lot of prior knowledge. SNIP [26] and SNIPER [27] use a scale regularization strategy to guarantee that the object size is within a fixed range of images of different resolutions. SNIPER uses super-resolution to recover the information of low-resolution objects, and adopts the strategy of regional sampling to further improve the training efficiency.

The addition of a super-resolution network will increase the network overhead and bring a burden to the network. Inspired by reference image super-resolution, EFPN [28] proposes an extended pyramid network from the perspective of enhancing feature map resolution to build a feature layer with more geometric details. It was designed for small objects by S.R. Noh et al. [29], who proposed a feature-level super-resolution method that uses high-resolution object features as supervision signals and matches the relevant receptive fields of input and object features; however, due to the imperfect construction of its FTT module, the restoration of the feature resolution is not effective. Chen Y et al. [30] proposed a feedback-driven data provider to balance the loss of small object detection. TridentNet [31] builds parallel multi-branch networks with different receptive fields by deepening the width of the network, and generates different small object features to improve performance, but increases the amount of network parameters. The above methods improve the performance of small object detection to varying degrees, but they are either for a certain type of object or not universal.

2.1.2. Feature Fusion

In the network, shallow features generally lack abstract semantic information and rich geometric details, while deep layers are just the opposite of shallow layers. In order to make the features have both deep fine-grained features and shallow high-resolution

spatial position information, the study is mainly from the perspective of feature fusion. Common fusion technologies mainly include feature splicing, feature summation, and the multiplication of corresponding elements. Although these fusion technologies can bring rich feature information between different stages, their processing of feature information is indistinguishable. That is, the convolution in the forward propagation treats all features equally. In fact, this will bring a lot of negative information, which will cause challenges to small object detection. Therefore, it is particularly important to selectively fuse features.

Zhang G J et al. [32] improved the RCNN and FPN structures, designed and integrated a global context network and a pyramid local context network, extracted context information globally and locally, and introduced a spatially aware attention module to guide the network to focus on more informative regions and generate more appropriate image features. M2Det [33] proposed a multi-level feature pyramid network (MLFPN) to build a more efficient feature pyramid to detect objects of different scales, using the decoding layer of the U-shaped module as the detection object feature. This improved the detection rate of objects, but increased the complexity of the network due to the complex U-shaped structure. The learnable weights for feature fusion were proposed in BiFPN [34], but ignored the influence of the dataset on weights. ASFF [35] proposed a data-driven pyramid feature fusion strategy that learns a method of spatially filtering conflicting information to suppress inconsistency, but its universality is low. Recently, Gong [36] began to study the weights of feature fusion, and generated a set of fusion weights by statistical methods and introduced them into the FPN structure, which further improved the detection performance of small objects.

Although these detection algorithms have excellent detection capabilities in natural images, their performance and application in small object detection in specific scenes are poor such as in UAV aerial images and satellite remote sensing images.

The main algorithms of small object detection are summarized in chronological order as shown in Table 1.

Table 1. Introduction to the mainstream algorithms of deep learning small object detection.

Chronological	Algorithm	Backbone	Method
2015	Faster R-CNN [3]	ResNet-50	It improves the fully connected layer and implements the stitching of multi-task loss functions.
2016	YOLOv1 [5]	GoogLeNet	It takes the entire graph as input to the network, regressing the location and category of the BBox directly at the output layer.
2016	SSD [6]	VGG16	The use of multi-feature mapping can be comparable to FasterRCNN in some scenarios, and the network optimization is simple.
2017	Mask R-CNN [4]	VGG16/Resnet	It redesigned the backbone network structure and replaced RoI Pooling with RoI Align.
2017	YOLOv2 [7]	Darknet19	It proposes a joint training method of object detection and classification using a new multi-scale training method.
2018	YOLOv3 [8]	Darknet53	It proposes multi-scale predictions.
2019	Trident [31]	ResNet	It designed the Trident parallel network to increase the width of the network and proposed the concept of parallel network.
2019	M2Det [33]	VGG-16/ ResNet-101	It proposed a multi-layer feature pyramid network.
2020	EfficientFet [34]	MobileNet	It proposed a two-way pyramid network, BiFPN.
2020	YOLOv4 [37]	CSPDarkent	It adopted various training skills and improved the loss function and maximum suppression method.
2021	EFPN [28]		Feature texture transfer (FTT) and foreground-background balance loss functions were designed to mitigate the area imbalance of the foreground and background.
2021	TPH-YOLOv5 [9]	CSPDarknet	It integrated transformer prediction heads (TPH) and CBAM into YOLOv5.
2021	YOLOX [38]	CSPDarknet	It decoupled the detection head and used dynamic sample matching.

2.2. Methods

2.2.1. MMF-YOLO Algorithm

The YOLO series network has a simple structure and improves the speed while ensuring the accuracy. Therefore, it has received extensive attention and application. In order to take into account the accuracy and speed at the same time, inspired by the YOLOX [38] network, this paper proposes the MMF-YOLO algorithm for the problems of overlapping small objects and large scale changes in complex scenes. It uses the backbone network of YOLOX, adds the MMF-Net multi-scale fusion module to the neck layer, and modifies the size of the object detection head. The overall algorithm structure is shown in Figure 1.

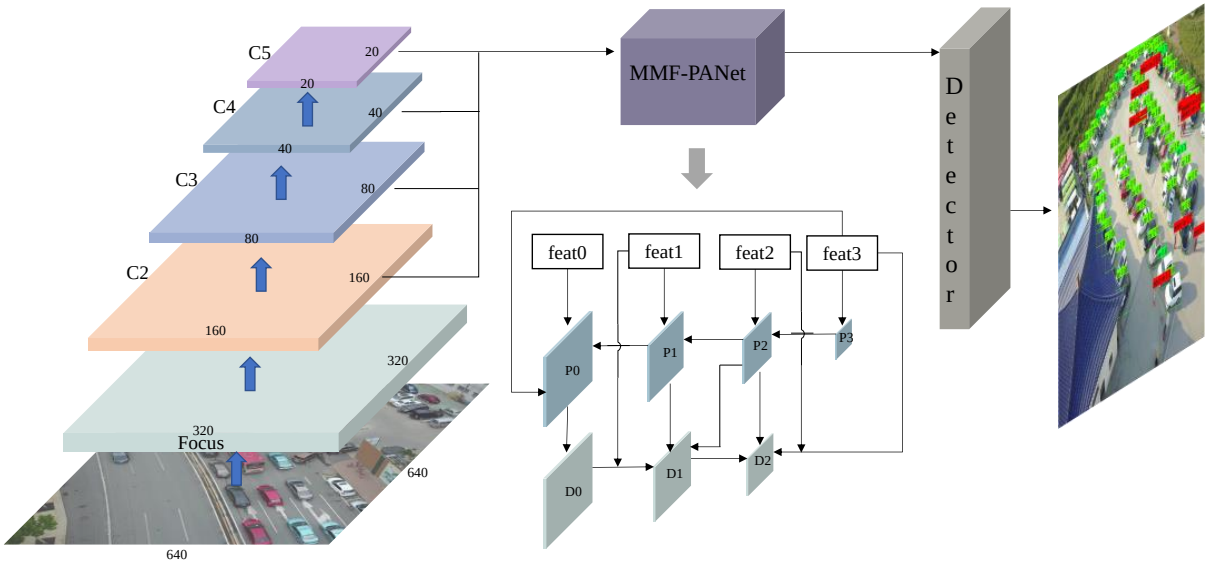


Figure 1. Overall framework of the MMF-YOLO algorithm.

From Figure 1, we can see that the network structure is roughly divided into three main bodies: the feature extraction network (backbone), the feature fusion network (neck), and the detection head (head). The whole process of the algorithm can be simply summarized as follows: the training images are extracted to obtain different levels of features, and then the features of different levels are output to the detection head through feature fusion, and the detection head completes the position adjustment of the prediction box to perform the detection task.

The specific process is as follows: the image goes through the backbone network CSPDarknet for feature extraction. First, the image goes through a Focus network structure to reduce the width and height to half the original size, so the width and height information is concentrated on the channel, and the channel is expanded to four times the original size. Then, the number of channels is further expanded through a convolution block, which consists of a conventional convolution, normalization, and Silu activation functions. Then, residual blocks of four different convolution kernels are used to filter the image information features, and to obtain four feature maps feat0, feat1, feat2, feat3 of the middle layer, respectively. These four features are input into the MMF-Net module for multiscale fusion.

Meanwhile, because the proportion of small objects in the picture is small, it is equivalent to local information relative to the entire picture, and the detection heads with a size of 20×20 in the three detection heads of the original YOLOX network are considered global relative to small objects, which is not conducive to the identification of small objects. Therefore, it is discarded, and in order to increase the classification score of small object detection, the feature fusion module through which feature feat0 passes is pulled out of a detection head.

The following will introduce the MMF-Net module structure and the improved attention mechanism and the size-changed object detection head structure, respectively.

2.2.2. MMF-PANet Feature Fusion Module

The network mainly relies on shallow features to detect small objects because the shallow features contain the complete shape and position information of the small objects, which are still in the initial stage of the network and have not been fully processed. The ability to express features is limited, and the receptive field is small. The global contextual information perception is weak and cannot be well adapted to small object detection. Therefore, in order to allow the shallow features to retain the spatial location information of their small objects, increase the connection of contextual information so that it can be fused with high-level semantics, and improve the utilization of foreground information, we propose a cross-scale, multi-path, and multi-flow feature fusion module with fusion factors, referred to as MMF-Net, which adds adaptive weights to each fusion branch, and its structure is shown in Figure 2.

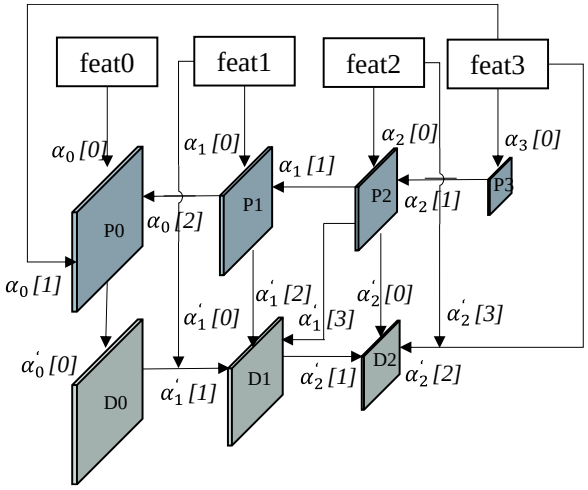


Figure 2. The MMF-PANet module structure with the branch fusion factor.

As can be seen from Figure 2, MMF-Net is composed of the fusion of features between different scales including two bidirectional fusion paths, namely the top–up and top–down paths. Compared with the original feature fusion network, five branches are added, two of which are the direct fusion of features across multiple levels and three skip connections, which can avoid the loss of object information during downsampling or pooling operations. The deep features are first fused with adjacent layers through the top–down path, while further feature extraction is completed, and P_3, P_2, P_1, P_0 are obtained, respectively. Then, when passing through a bottom–up aggregation path, it is fused with the features of the top–up path output and the output of the backbone network to obtain D_0, D_1, and D_2, respectively. In addition, considering that feature extraction at the same time as layer-by-layer splicing will lead to the loss of some feature information. In the feature fusion module, we added multiple radial paths of features feat1, feat2, and feat3 produced by the backbone network to P_0, D_0, D_1, and D_2, respectively.

Its calculation process can be expressed as:

$$P_3 = f_{conv}^{1\times1}(feat3) \tag{1}$$

$$P_2 = Cat(P_3, feat2) \tag{2}$$

$$P_1 = Cat(f_{conv}^{1\times1}(f_{conv}^{1\times1}(P_2)), feat1) \tag{3}$$

$$P_0 = Cat(Cat\begin{pmatrix} f_{conv}^{1\times1}(f_{conv}^{1\times1}(P_1)), \\ feat0 \end{pmatrix}, feat3) \tag{4}$$

$$D_0 = f_{conv}^{1\times1}(f_{conv}^{1\times1}(P_0)) \tag{5}$$

$$D_1 = Cat(f_{conv}^{3\times3}(D_0), P1) \tag{6}$$

$$D_0 = Cat(Cat\left(\begin{matrix} f_{conv}^{3\times3}(f_{conv}^{1\times1}(f_{conv}^{1\times1}(D_1))), \\ feat2 \\ f_{conv}^{1\times1}(P_2) \end{matrix}\right), feat3) \tag{7}$$

Equations (1)–(7) clearly show the direction of each feature in each path in the feature fusion module MMF-Net. In the formula, $f_{conv}^{3\times3}$ and $f_{conv}^{1\times1}$ represent the conventional convolution operations with convolution kernel sizes of 3×3 and 1×1, respectively, Cat represents feature map stitching operation, $feat3$, $feat2$, $feat1$, $feat0 \in R^{C\times H\times W}$.

The feature fusion module that adds multiple paths can bring a large number of other size features to each level, alleviating the limitation of a fixed size. The feature information carried by each scale feature map has its own characteristics. For example, the shallow features focus on the location, contour, and other information of the object, and the receptive field is relatively small, which is not sensitive to global information and depends on the context content. The deeper features mainly express the detailed information such as the texture semantics of the object, and the receptive field is relatively large, which can take into account the overall situation. In the fusion, all the information will be treated equally and fused indiscriminately, which will cause a problem. The background information of each feature is accumulated, so the network mistakenly believes that the received information is important information, and it is also used as the main feature for subsequent feature enhancement extraction, so it is easy to misjudge the detection of small objects and is counterproductive.

In order to solve this problem, we added a fusion factor to each fusion path, which can improve the scale invariance of features and reduce the interference of background information to screen effective features for the network, reduce feature redundancy, and enhance the representation ability of small object features. It adaptively assigns weights to each fusion channel, and assigns different weights to the feature information from different levels, so that the fused features have both shallow high-resolution spatial information and deep feature detail semantic information, contacts the context content, and use the context information to detect small objects.

We set a learnable fusion factor α for the features that flow to the fusion path, and used weight normalization to constrain the value range of each weight so that $\alpha \in (0, 1)$. In order to prevent the gradient from disappearing due to zero weight, given a parameter $e = 0.001$, the network adjusts the weight according to the value of each feature tensor and controls the proportion of each feature map in the fused feature information to represent the importance of features at different scales. The dynamic adjustment of the fusion factor determines the retention of all information in each feature. The fusion factor is added to the optimizer as a hyperparameter, and is optimized along with the parameters of the optimizer until an optimal weight is learned.

The feature calculation with the fusion factor added can be expressed as:

$$F_j = f_{conv}(\frac{\alpha_j[1]\cdot P_{j1}^{in} + \alpha_j[2]\cdot P_{j2}^{in} + \cdots + \alpha_j[i]\cdot P_{ji}^{in}}{\alpha_j[1] + \alpha_j[2] + \cdots + \alpha_j[i] + e}) \tag{8}$$

where f_{conv} represents a series of convolutional operations; $\alpha_j[i]$ represents the ith fusion factor of the jth layer feature; P_{ji} represents the ith feature to be fused in the j layer; F_j represents the feature output after fusing all the features and multiplies each input feature with the corresponding fusion weight factor to obtain the final feature.

2.2.3. Improvements to the Attention Mechanism

In order to effectively distinguish the foreground–background information in complex scenes during feature enhancement, only improving the possibility of small object information being enhanced, we decided to add an attention mechanism with little addi-

tional overhead to the network model, through which critical features are promoted and suppressed that are not vital to the current task. Considering that both spatial position information and detailed semantic information may be the key to the accuracy of small object detection, the coordinate attention mechanism (CA) [39] and the convolutional block attention module attention mechanism (CBAM) [40] can be used as attention mechanisms to be added. Both CA and CBAM simultaneously filter the information of the two dimensions of feature channel and feature space, but their internal mechanisms and impact factors are different. Nevertheless, it is not only the type of attention mechanism that affects the capture of an area of an intensive object, but also the location of the joining modules. Thus, to obtain the effect of the two attention mechanisms on the network model, the following experiments were performed on them separately, and the experimental results are shown in Table 2.

Table 2. Effects of different attention mechanisms on the network model performance.

Attention Mechanism	mAP/%
CA_v1	32.63
CBAM_v1	32.95
CA_v2	33.10
CBAM_v2	33.27

v1 in Table 2 means adding an attention mechanism between the backbone network and the neck, while v2 means adding an attention mechanism when the neck performs feature extraction. It can be seen from Table 1 that when CA and CBAM were added between the backbone network and the neck, it had a negative effect on the performance of the network. When the two were added to the neck separately, they showed different effects. CA did not affect the network performance improvement, while CBAM increased the mAP of the model by 0.17%. Therefore, we chose the CBAM to alleviate the problem of information interference caused by densely connected objects.

Furthermore, in order to allow the initial features entering the attention mechanism to better focus on the foreground information, and to filter the background information that is highly similar to the small object, we improved the attention mechanism.

We let the initial feature $f \in R^{C\times H\times W}$ first pass through the channel attention mechanism CAM, increasing the weight on the channel and reducing the features in other dimensions, so that the information on the channel can be focused during feature extraction. Then, the Hadamard product of the weights output by the CAM and the initial features is performed to obtain the intermediate features $f_c \in R^{C\times H\times W}$, where the information is focused on the channel. Then, the intermediate features are input into the spatial attention mechanism SAM, and the intermediate features go through the Sigmoid function in space, and the weight coefficients about the spatial position are obtained based on the channel information, and the coefficients are between (0, 1). It is multiplied with the initial feature tensor elements one-to-one, and the feature information $f_s \in R^{C\times H\times W}$ of the corresponding spatial position on the channel is obtained. Finally, f_c and f_s are added in the channel dimension to obtain the complementary information of the feature in the channel and space. The effective information is prepared in advance for the next convolution, and the final feature $f_{cs} \in R^{C\times H\times W}$ is output.

Unlike the original CBAM attention mechanism, we multiplied the initial features with the corresponding weights through the SAM attention mechanism. This was to make the network targeted when performing feature extraction, reducing the overhead of other dimensions, and speeding up the efficiency of the network operation. The structure of the improved attention mechanism is shown in Figure 3.

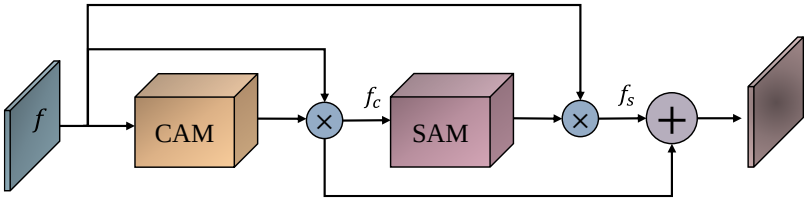


Figure 3. M-CBAM attention mechanism structure.

Figure 3 shows the overall architecture of the M-CBAM attention mechanism. The calculation process of the M-CBMA attention mechanism can be represented by Formula (9):

$$f_{cs} = Add(f * CAM(f), f * SAM(f * CAM(f))) \quad (9)$$

$*$ represents the Hadamard product, that is, the corresponding multiplication of two matrix elements, and Add represents the addition of the feature information on the channel of the feature map.

In order to verify the effectiveness of the improved attention mechanism, we added the basic network to the original attention mechanism CBAM and the improved attention mechanism M-CBAM to conduct the following comparative experiments. The experimental results are shown in Table 3.

Table 3. Comparison of the experimental results of the attention mechanism.

Attention Mechanism	mAP/%
-	33.10
CBAM	33.27
M-CBAM	33.53

As can be seen from the table, compared to the mAP of the original CBAM, our improved M-CBAM increased by nearly 0.2%, which was 0.43% higher than the basic network, which showed that our improved attention mechanism is more helpful to the network.

2.2.4. Feature Enhancement Extraction and Object Detection Head

The small object of the natural image is highly similar to the background information, especially at night or in the case of weak lighting conditions. In order to distinguish between the two lots of information, we used multi-scale multi-path multi-flow feature fusion module processing; although it can bring rich information at each scale, it can also inevitably bring feature overlap and background information that is highly similar to the object. In order to obtain more information that is beneficial to small objects, we performed feature enhancement extraction after feature fusion, and further filtered the fused information.

The feature enhancement extraction module (FEE) is a double residual structure, as shown in Figure 4, which consists of two paths. One of the paths uses the convolution kernel of 3×3 conventional convolution as the filter to extract features. Meanwhile, in order to extract the effective information of a large number of small objects, we chose to place the M-CBAM into the 1×1 filter and the 3×3 filter between filters. The purpose was to make the network focus on the channel and space dimensions before filtering the feature information, thus focusing the effective information of each fused feature on the channel and space while discarding unnecessary features. The structure of the feature enhancement extraction module with the addition of the M-CSAM attention mechanism is shown in Figure 4.

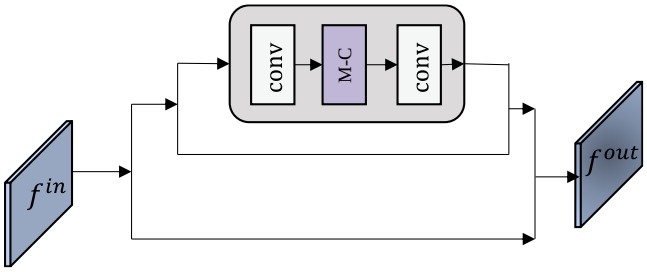


Figure 4. Simplified structure diagram of FEE.

Figure 4 shows the simple structure of the feature extraction module, M-C represents M-CBAM, and its specific calculation process can be expressed as:

$$P_a = Add\left(f_{conv}^{1\times1}(x), f_{conv}^{3\times3}\left(C\left(f_{conv}^{1\times1}(f_{conv}^{1\times1}(f^{in}))\right)\right)\right) \tag{10}$$

$$P_b = f_{conv}^{1\times1}(x) \tag{11}$$

$$f^{out} = Add(P_a, P_b) \tag{12}$$

Among them, f^{in} represents the input feature; P_a represents the feature output in the main path in the residual structure; P_b represents the secondary residual edge feature output; and f^{out} is the final output feature.

Unlike the large- and medium-sized objects in the image, the small objects are small in size, and some small objects even have only a few pixels to a dozen pixels, so their detection mainly relies on shallow features. Since the features of the shallow network have not been subjected to a large number of downsampling and the information about the small object in the features has not been lost, it is necessary to use the output of the shallow features as the detection head while simultaneously considering that the small-size detection head is not helpful for small object detection, and its existence will increase the network calculation amount. After balancing the two, we decided to abandon the small-size detection head extended by the deep network. Because the small object pixels were between 0 and 32 × 32, and the scale changed greatly, detection heads of different scales were required to complete the detection task. Therefore, the detection head extended from the shallower and deeper features in the original network was retained. Finally, the size of the detection head was determined as 160 × 160, 80 × 80, and 40 × 40, and its structure is shown in Figure 5.

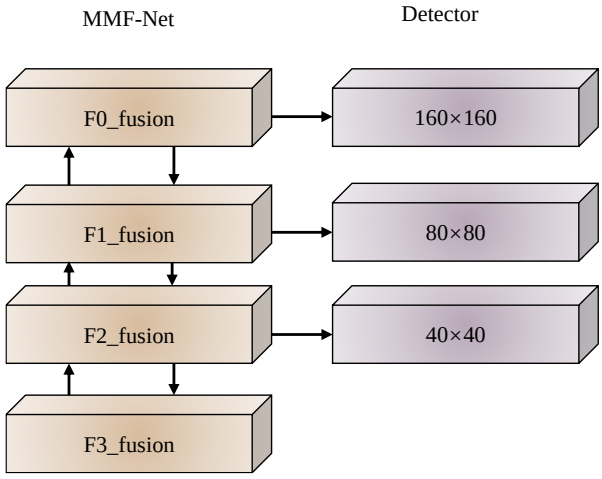


Figure 5. Improved schematic diagram of the object detection head structure.

It is worth noting that in order to avoid conflicts between classification and regression tasks in the same convolution, the structure of the MF-YOLO algorithm inherits the detection head of the YOLOX decoupling method, and its structure is shown in Figure 6.

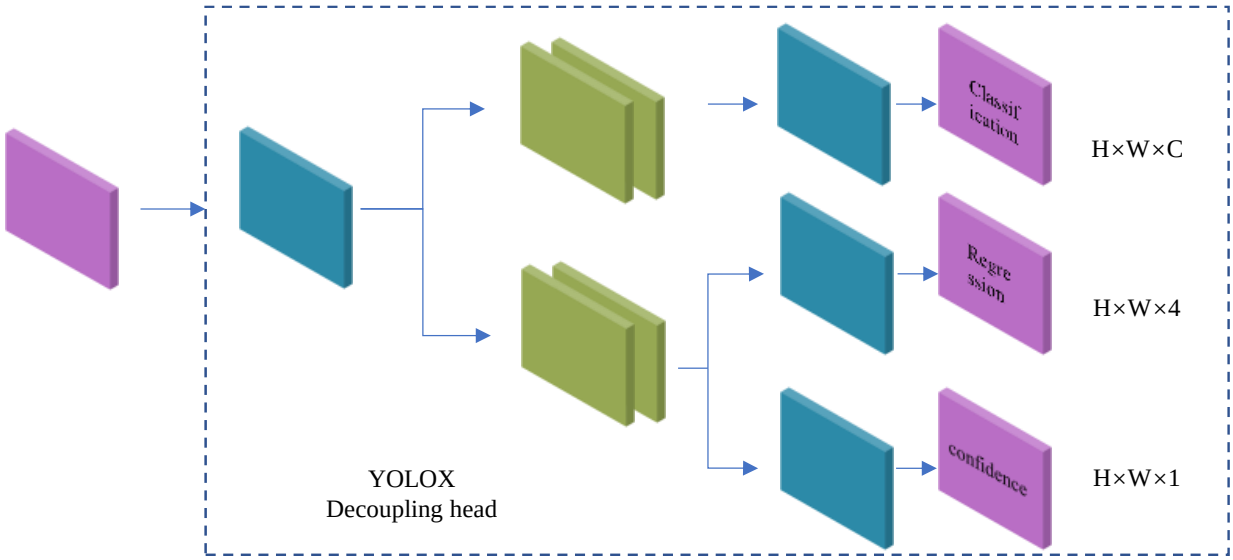


Figure 6. Comparison between the coupling detector and decoupling detector.

Figure 6 shows the two structures of the detector coupling head and the decoupling head. Our network used two 1 × 1 convolutions to complete the detection regression and classification tasks, respectively. This structure can speed up the model convergence speed and improve the detection accuracy.

3. Experiment

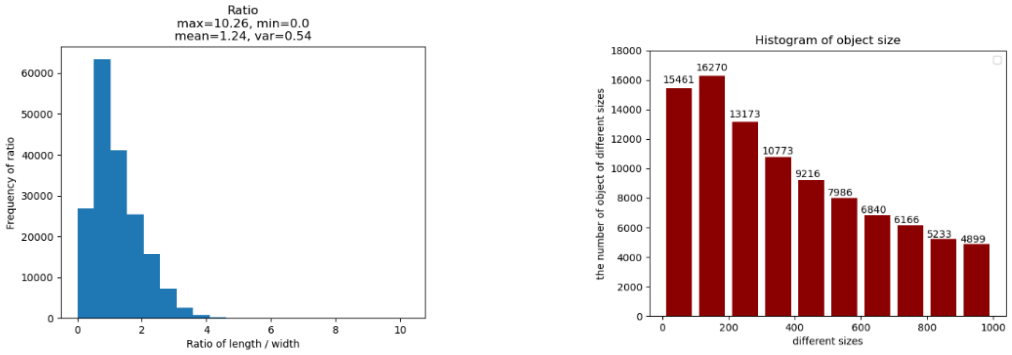
3.1. Experiment Platform

The experiment in this paper was based on the Ubuntu18.04 operating system, the deep learning environment was equipped with the CUDA11.0 and Pytorch1.7.0 frameworks, and the NIVDIA RTX3080Ti GPU was used to accelerate the model training. Taking YOLOX as the experimental base model, we used the YOLOXs model as the pre-training weight, and performed the network improvement and optimization on it.

3.2. Introduction to the Dataset VisDrone

In order to verify the effectiveness of the proposed method for problems such as small object continuity and occlusion, we selected VisDrone [41], a dataset captured by drones that contains a large number of scale-variant objects, occlusions, and class imbalances with complex backgrounds and variable angles. It is collected by the AISKYEYE team at the Lab of Machine Learning and Data Mining, Tianjin University, China. It was shot using different types of drones and included various scenes in various cities such as low light, rainy weather, exposure, etc. Unlike conventional detection datasets, each image contains hundreds of objects to be detected, and the dataset contains a total of 2.6 million annotated boxes, and some objects that are very close to each other may also have overlapping bounding boxes. Because the dataset is captured by drones, the annotation frames of pedestrians and distant objects are very small, which poses a certain challenge to the ability of the model to generate an a priori frame. The object size distribution of the VisDrone dataset is shown in the Figure 7.

The training pictures of different scenes and different climates in the VisDrone dataset are shown in Figure 8.



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

Figure 7. The distribution of the number of small objects in the VisDrone dataset and the distribution of the ground truth boxes. (**a**,**b**) Description of the size scale and number of objects in the dataset, respectively.(**a**) Ratio of the object length to width and the number of occurrences of each ratio in the dataset image. It can be seen from the figure that the object size ratio is mainly concentrated between 0 and 2. (**b**) Distribution of the number of objects of different sizes. The abscissa represents the object size, and the ordinate represents the quantity of objects of different sizes. It can be seen from the figure that the object size is mainly concentrated within 800 pixels, which is much smaller than the 32×32 size.

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

Figure 8. Training images of different scenes with different lighting in the VisDrone dataset. (**a**–**c**) represent small objects in low light, at night, and under exposure, respectively.

3.3. Model Training and Evaluation Metrics

In this experiment, the small object dataset was divided into the training set and verification set according to the ratio of 9:1, and Mosaic was used for image input into the network. The experiment used transfer learning, a total of 130 epochs were trained, the first 50 epochs were frozen training, the last 80 epochs were thawed, the initial learning rate was set to 0.01; in the frozen training stage, the learning rate increased according to the rate of 0.001 and in the thaw training according to the rate of 0.0001.

The experiments in this paper used evaluation metrics commonly used in deep learning, namely precision (Precision, P), recall (Recall, R), average precision (Average Precision, AP), and mean average precision (Mean Average Precision, mAP), and the model parameter size.

AP is the area enclosed by the horizontal and vertical coordinates and the curve. The calculation formulas of AP and mAP are:

$$\mathrm{AP} = \int_0^1 \mathrm{P(R)d(R)} \tag{13}$$

$$\mathrm{mAP} = \frac{1}{\mathrm{M}} \sum_{\mathrm{j}=1}^{\mathrm{M}} \mathrm{AP_j} \tag{14}$$

where P and R represent precision and recall, respectively.

4. Analysis of Experimental Results

4.1. Ablation Experiment Results

In order to verify the effectiveness of the proposed modules in the MMF-YOLO algorithm for small objects, the VisDrone dataset was used to perform ablation experiments on each module based on the YOLOXs model.

Head in Table 4 represents the detection head with the modified size, and Experiment ⑩ represents the experimental results of the basic network. Experiments ① and ② indicate that the base network was added to MMF-Net with the adaptive fusion factor and MMF-Net without the fusion factor, respectively.

Table 4. Ablation experiment results.

Order	MMF-Net		M-CBAM	Head	mAP	Paras/MB
	$+\alpha$	$-\alpha$				
①	√				35.73	38.12
②		√			35.32	38.12
③	√		√		41.21	29.3
④	√			√	38.01	29.3
⑤			√		33.53	34.27
⑥			√	√	39.94	27.75
⑦				√	36.51	27.67
⑧		√	√	√	41.69	29.37
⑨	√		√	√	42.23	29.33
⑩					33.10	34.11

Comparing the two with the basic network, it can be found that the proposed MMF-Net multi-branch cross-scale skip connection fusion module could effectively improve the detection accuracy of small objects, increase mAP to more than 35%, and only sacrificed a small number of parameters. Compared with the baseline, they increased by 2.63% and 2.42%, respectively. Comparing the two, it can be seen that the fusion factor α could further improve the accuracy of small object detection and increase the mAP of small objects to 35.73% without bringing any parameters.

Experiment ⑤ indicates that adding the improved M-CBAM attention mechanism to the base network increased mAP by 0.44%. Experiment ⑦ showed that when the improved Head module was added, the mAP of the model increased by 3.41% compared with the basic network, and the size of parameters was reduced by 6.44 MB. It was proven that the added large-size detection head was better than the small-size detection head for the task of small object detection, and would reduce a lot of computation. Experiment ⑨ means that the α-MMF-Net module and the M-CBAM attention mechanism and the improved detection head were added to the network at the same time, and its mAP reached 42.23%. Compared with the basic network, the mAP increased by nearly 10%, and the number of model parameters was simplified to 29.33 MB, which was reduced by nearly 5 MB compared to the original network, which verified the effectiveness of the proposed module for small object detection tasks.

4.2. Comparative Experiments

In order to verify the reliability of the proposed network, we roughly divided the training into three stages, where each stage had 40 epochs, and randomly selected one epoch in each stage to evaluate the baseline network YOLOX and MMF-YOLO, the results of which are shown in Table 5.

Table 5. Model evaluation at different stages.

40 Epoch/mAP (%)		80 Epoch/mAP (%)		120 Epoch/mAP (%)	
YOLOX	MMF_YOLO	YOLOX	MMF_YOLO	YOLOX	MMF_YOLO
11.57	13.68	31.84	40.35	33.23	41.45

We select the 40th epoch, the 80th epoch, and the 120th epoch for evaluation in the three stages. As can be seen in Table 5, the benchmark and MMF-YOLO networks had low mAPs in the 40th epoch experiment, but our network was slightly better, at 2.11% higher than YOLOX. At the 80th epoch, after sufficient training, the network began to converge. The mAP of YOLOX and MMF-YOLO reached a high level, while the mAP of MMF-YOLO was higher, reaching 40.35%, which was 8.51% higher than that of YOLOX. At the 120th epoch, the network was relatively stable and had completely converged. The accuracy of the two networks was close to the best level, reaching 33.2% and 41.45%, respectively. Our network showed better performance than the benchmark network at different stages. Although the performance in the first stage was poor, the overall level was higher, which proves that our network can effectively improve the accuracy of small object detection and proves the reliability of the network.

In order to further verify the effectiveness of the proposed algorithm for small object detection, we experimented with the MMF-YOLO algorithm and other mainstream algorithms for object detection on the VisDrone dataset, and set the input image size to 640 × 640. Due to the limitation of the experimental equipment, some experimental data were obtained by referring to the literature. The experimental results are shown in Table 6.

Table 6. Comparative experimental results.

Order	Model	Backbone	Input Resolution	mAP/%	Paras/MB
①	YOLOv5-L [9]	CSPDarknet	1920 × 1920	28.88	
②	TPH-YOLOv5 [9]	CSPDarknet	1536 × 1536	39.18	
③	Faster R-CNN [42]	ResNet-50	1000 × 600	21.7	
④	Cascade R-CNN [42]	ResNet-50	1000 × 600	23.2	
⑤	RetinaNet [42]	ResNet-50	1000 × 600	13.9	36.53
⑥	YOLOv3 [43]	Darknet53	800 × 1333	22.46	234.9
⑦	SSD [43]	VGG16	800 × 1333	21.10	95.17
⑧	YOLOv4 [44]	CSPDarknet	1000 × 600	30.7	244.11
⑨	YOLOX	CSPDarknet	640 × 640	33.10	34.11
⑩	MMF-YOLO	CSPDarknet	640 × 640	42.23	29.33

It can be seen from Table 6 that our network MMF-YOLO performed better on mAP, up to 42.23%, and its input resolution was only 640 × 640, and the parameter size was only 29.33 MB, which was 4.74 MB less than the original network. Compared with other networks in the YOLO series, YOLOv5, YOLOv3, YOLOv4, the mAP increased by about 14%, 20%, and 12%, respectively, which may be further improved if the input resolution is the same. Compared with the TPH-YOLOv5 network, the mAP increased by 3.05%, and it used YOLOv5-L as the backbone network, while our network used the depth and width of YOLOX-s, so its parameters were much larger than the improved network. Compared with other mainstream algorithms with different input resolutions, the mAP also had different improvements. Compared with the RetinaNet algorithm, the mAP had been improved by 28.33%.

Table 7 shows the model parameter size and model calculation amount of the mainstream deep learning algorithms. From Table 8, it can be seen that MMF-YOLO had the least model parameters, only 29.33 MB, which was 5.22 MB less than the benchmark network, sacrificing some computational complexity. The complexity increased to 63.5 G, which may be the amount of computation brought by the process of increasing multiple fusion paths.

Table 7. Comparison of the model parameters and computational amount.

Order	Model	Paras/MB	FLOPs
①	YOLOv5-L	178.09	114.413 G
②	Faster R-CNN	522.21	402.159 G
③	RetinaNet	36.53	166.711 G
④	YOLOv3	234.9	155.249 G
⑤	SSD	95.17	277.8 G
⑥	YOLOv4	244.11	
⑦	YOLOX	34.11	26.657 G
⑧	MMF-YOLO	29.33	63.562 G

Table 8. Comparison of various types of AP between the mainstream algorithms and the MMF-YOLO algorithm.

Model	Car /%	Bus /%	Truck /%	Van /%	Pedestrian /%	Motor /%	People /%	Tricycle /%	Awn-Tri /%	Bicycle /%
TPH-YOLOv5 [9]	68.9	61.8	45.2	49.8	29.0	30.9	16.8	27.3	24.7	15.7
Faster R-CNN [42]	51.7	31.4	19.0	29.5	21.4	20.7	15.6	13.1	7.7	6.7
Cas R-CNN [42], *	54.6	34.9	21.6	31.5	22.2	21.4	14.8	14.8	8.6	7.6
RetinaNet [42]	45.5	17.8	11.5	19.9	13.0	11.8	7.9	6.3	4.2	1.4
YOLOv4 [44]	64.3	44.3	22.7	22.4	24.8	21.7	12.6	11.4	7.6	8.6
YOLOv3-LITE [45]	70.8	40.9	21.9	31.3	34.5	32.7	23.4	15.3	6.2	7.9
YOLOX	69.4	46.6	42.3	40.2	33.0	34.7	22.1	22.5	12.4	11.3
MMF-YOLO	77.7	58.6	54.19	49.12	47.97	40.4	30.4	29.4	15.39	19.2

* Cas R-CNN is Cascade R-CNN.

Table 8 shows a comparison of the various class accuracies of mainstream networks and our algorithm. From Table 5, it can be seen that the AP of each category of Bi-YOLOX was higher than other one-stage classical network algorithms, and the car category with the highest AP reached 77.7%, which was 8.3% higher than the basic network and 32.2% higher than that of RetinaNet.

Compared with the prototype network, the bus category was improved by up to 12%, 40.8% higher than RetinaNet, and the pedestrian class with high missed detection rate and false detection rate was increased by as much as 14.97%. For the bicycle category with a smaller object size, AP increased by 8%. Compared with TPH-YOLOv5, the accuracy of MMF-YOLO in the three categories of bus, van, and awning-tricycle was slightly inferior, but the difference between the two was not large. This is because the resolution of its input image was much higher than that of the input image of the MMF-YOLO network. If the size of the input image is the same, the MMF-YOLO network is likely to overtake TPH-YOLOv5. Compared with the base network, the APs of each class of our network had different degrees of improvement and were higher than their average accuracy. In summary, the improved algorithm in this paper has significant advantages over other algorithms in terms of the average detection accuracy or category accuracy or model size.

Figure 9 visually shows the accuracy comparison of each category between the classical network and the MMF-YOLO network. It is clear from Figure 9 that our network occupied a high position in most categories and was higher than every category in the original network.

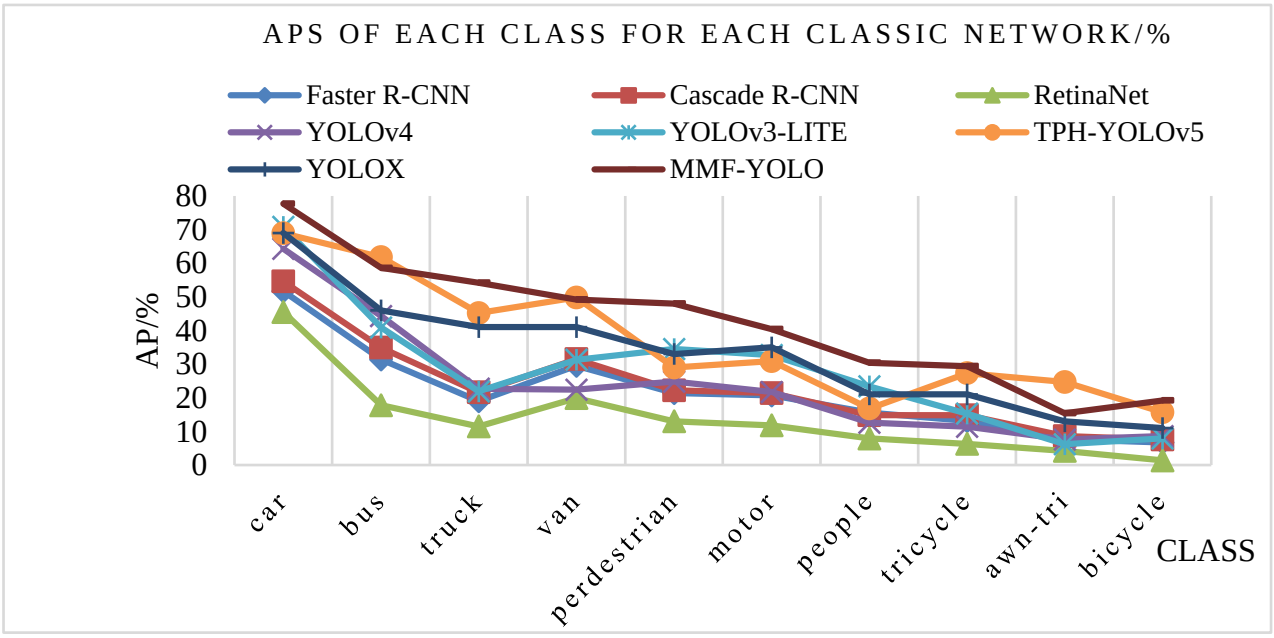
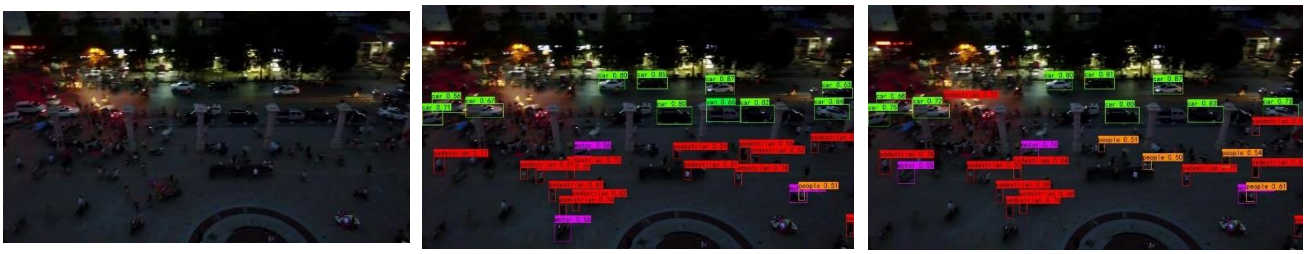


Figure 9. Comparison of the mainstream algorithm and MMF-YOLO network algorithm for each category of AP.

4.3. Visualize the Results

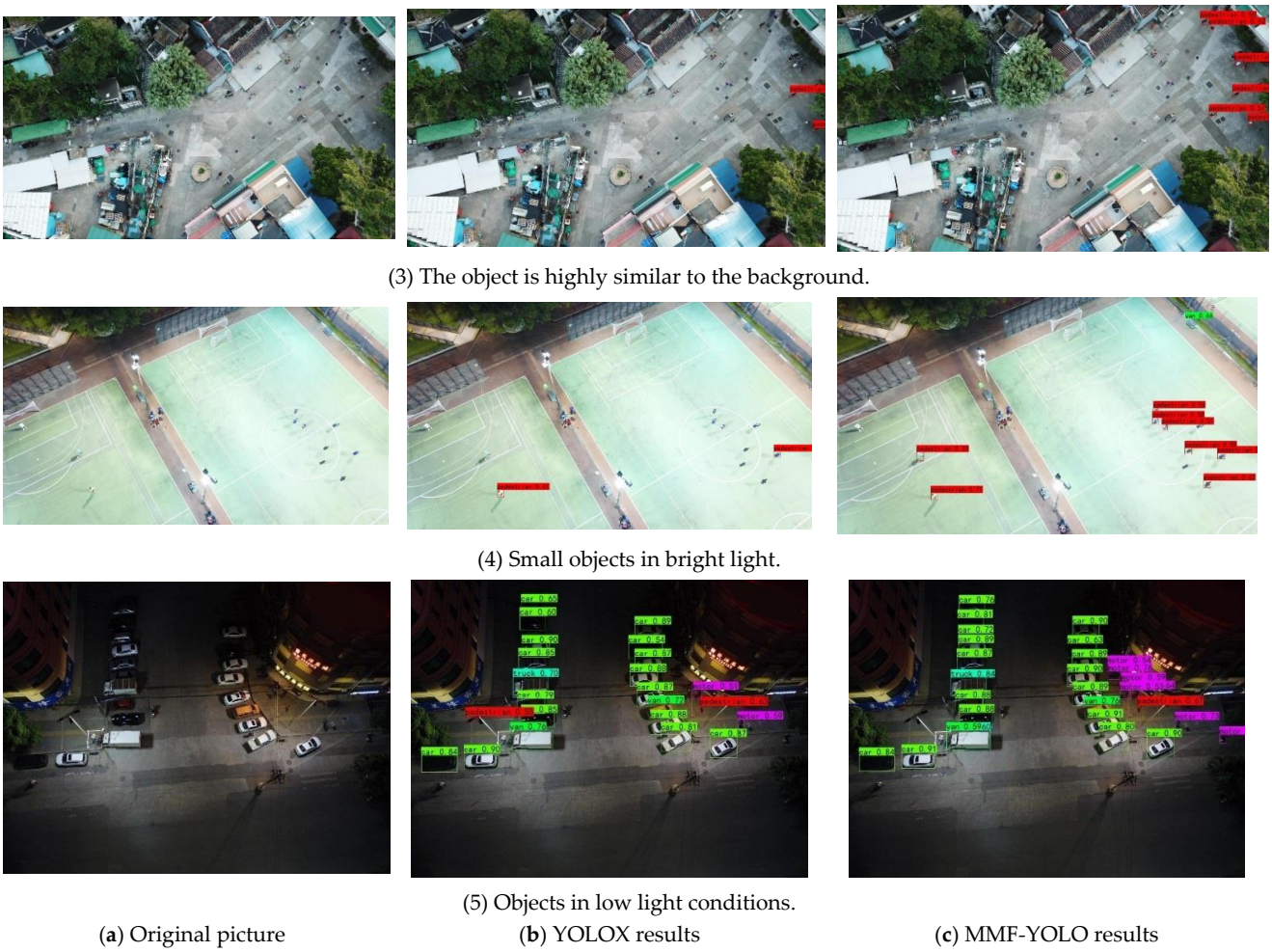
In order to analyze the detection results of the algorithm more directly, we randomly selected sample images of small objects in the VisDrone test set for testing, and visualized the comparison and analysis. Due to the rotation of the picture taken by the drone, the object in the picture is blurred and the size of the distant object is small. Figure 10 shows the detection results of a complex background, object occlusion, and multi-scale dense small objects in the VisDrone dataset, respectively.



(1) Small objects in complex backgrounds.

(2) Overlapping and dense objects.

Figure 10. *Cont.*



(3) The object is highly similar to the background.

(4) Small objects in bright light.

(5) Objects in low light conditions.

(**a**) Original picture (**b**) YOLOX results (**c**) MMF-YOLO results

Figure 10. Result detection of small objects in each scene, various weather, and different lighting conditions in the VisDrone dataset. (**a**) represents the image to be detected, (**b**) represents the detection result of the YOLOX model, and (**c**) represents the detection result of the MMF-YOLO model. (1) represents the situation of a complex background under low light at night. It can be seen from (1) that the YOLOX network had more missed detection instances for the people class, and some van classes were incorrectly identified as the car class, and pedestrian and motor were detected as the pedestrian class. However, the MMF-YOLO network is able to avoid these problems by correctly detecting classes such as cars and people. The analysis showed that the pixels in the dataset such as people are highly similar to the background, and the YOLOX network mistakenly identifies them as background information, resulting in missed detection. (2) represents a dense small object with many overlapping objects. The YOLOX network has different degrees of missed detection of the people, pedestrian, and tricycle categories, and misidentified background information such as truck categories, and our network could not only correctly detect these categories, but also had a high level of confidence. (3) and (4) show the detection results of different scenes during the day and under strong light, respectively. There were a lot of missed detections in the YOLOX detection results, and the MMF-YOLO network could not completely detect small objects. However, compared with the YOLOX network detection results, it was greatly improved, and it could detect small objects that people may not be able to distinguish at a glance. (5) displays the detection results of small objects with relatively large object pixels under the condition of weak night light. Both YOLOX and MMF_YOLO networks correctly detected small objects of larger size, but the MMF_YOLO networks generally had a high level of confidence and were able to detect small objects with very few pixels that are not easily discoverable such as the motor and pedestrian categories.

5. Conclusions

The factors affecting the detection accuracy of small objects mainly include the size of the object pixel, background information being highly similar to the object pixel, object overlap, and pixel blur. According to the problems of small object scale, large object scale span in the image, and similar object and background information, we proposed an improved MMF-YOLO algorithm. In order to alleviate the problem that the scale of small objects is large and the object information is highly similar to the background information, a feature fusion module capable of adaptive learning was proposed. By adding cross-scale paths and skip connections, it brings deep high-level semantic information and shallow spatial location information to fixed-size features, and filters redundant features for fusion features through adaptive fusion factors such as the improved attention mechanism M-CBAM, adding weight to the object information, filtering background information, and overcoming the problem of object overlap. Finally, by adjusting the size of the object detection head, the algorithm is adapted to the detection of objects of different scales, which increases the probability of small objects being detected. Experiments show that the proposed algorithm can significantly improve the mAP of small objects by 10%, and the size of the parameters was reduced by about 5 MB.

Although the proposed algorithm improved the detection accuracy of small objects, the detection effect of small objects with only a few pixels in the case of being occluded was generally average, and the optimization of the prediction frame was not considered. Therefore, this will be strengthened in future work.

Author Contributions: Conceptualization, Q.Z. and H.Z.; Methodology, Q.Z. and X.L.; Software, Q.Z.; Validation, Q.Z.; Formal analysis, Q.Z.; Investigation, Q.Z., H.Z. and X.L.; Resources, Q.Z.; Data curation, Q.Z.; Writing—original draft preparation, Q.Z.; Writing—review and editing, X.L.; Visualization, Q.Z.; Supervision, H.Z.; Project administration, H.Z.; Funding acquisition, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 61872304).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chen, F.; Ding, Q.; Hui, B.; Chang, Z.; Liu, Y. Multi-scale kernel correlation filter algorithm for visual tracking based on the fusion of adaptive features. *Acta Optics* **2020**, *40*, 109–120.
2. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
3. Ren, S.; He, K.; Girshick, R.; Sun, J. *Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; p. 28.
4. 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; pp. 2961–2969.
5. 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; pp. 779–788.
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
7. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 21–26 July 2017, Honolulu, HI, USA; IEEE: New York, NY, USA, 2017; pp. 6517–6525.
8. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
9. Zhu, X.; Lyu, S.; Wang, X.; Zhao, Q. TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2778–2788.

10. 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, IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
11. Xinbo, G.; Mengjingcheng, M.; Haitao, W.; Jiaxu, L. Research progress of small target detection. *Data Acquis. Process.* **2021**, *36*, 391–417. [CrossRef]
12. Lin, T.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *Computer Vision–ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; Volume 8693, pp. 740–755.
13. Hongguang, L.; Ruonan, Y.; Wenrui, D. Research progress of small target detection based on deep learning. *J. Aviat.* **2021**, *42*, 107–125.
14. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA., 21–26 July 2017; IEEE: New York, NY, USA, 2017; pp. 936–944.
15. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
16. Pang, J.; Li, C.; Shi, X.Z.; Feng, H. R2 -CNN: Fast Tiny Object Detection in Large-Scale Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5512–5524. [CrossRef]
17. Yin, Z.; Guiyi, Z.; Tianjun, S.; Kun, Z.; Junhua, Y. Small object detection in remote sensing images based on feature fusion and attention. *J. Opt.* **2022**, 1–17. Available online: http://kns.cnki.net/kcms/detail/31.1252.O4.20220714.1843.456.html (accessed on 26 October 2022).
18. Jinkai, W.; Xijin, S. Review of Applied Research on Computer Vision Technology. *Comput. Age* **2022**, 1–4+8. [CrossRef]
19. Yuan, X.; Shi, J.; Gu, L. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Syst. Appl.* **2021**, *169*, 114417. [CrossRef]
20. Raghunandan, A.; Mohana; Raghav, P.; Ravish Aradhya, H.V. Object Detection Algorithms for Video Surveillance Applications. In Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 3–5 April 2018.
21. Kermany, D.S.; Goldbaum, M.; Cai, W.; Valentim, C.C.S.; Liang, H.; Baxter, S.L.; McKeown, A.; Yang, G.; Wu, X.; Yan, F.; et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell* **2018**, *172*, 1122–1131.e9. [CrossRef] [PubMed]
22. Liu, S.; Cai, T.; Tang, X.; Zhang, Y.; Wang, C. Visual recognition of traffic signs in natural scenes based on improved RetinaNet. *Entropy* **2022**, *24*, 112. [CrossRef] [PubMed]
23. Sun, X.; Wang, P.; Wang, C.; Liu, Y.; Fu, K. PBNet: Part-based convolutional neural network for complex composite object detection in remote sensing imagery. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 50–65. [CrossRef]
24. Li, Y.; Zhao, J.; Lv, Z.; Li, J. Medical image fusion method by deep learning. *Int. J. Cogn. Comput. Eng.* **2021**, *2*, 21–29. [CrossRef]
25. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale Match for Tiny Person Detection. In Proceedings of the 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), Snowmass, CO, USA, 1–5 March 2020; pp. 1246–1254. [CrossRef]
26. Singh, B.; Davis, L.S. An analysis of scale invariance in object detection snip. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3578–3587.
27. Singh, B.; Najibi, M.; Davis, L.S. *Sniper: Efficient multi-scale training. Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2018; p. 31.
28. Deng, C.; Wang, M.; Liu, L.; Liu, Y.; Jiang, Y. Extended feature pyramid network for small object detection. *IEEE Trans. Multimed.* **2021**, *24*, 1968–1979. [CrossRef]
29. Noh, J.; Bae, W.; Lee, W.; Seo, J.; Kim, G. Better to follow, follow to be better: Towards precise supervision of feature super-resolution for small object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 9725–9734.
30. Chen, Y.; Zhang, P.; Li, Z.; Li, Y. Stitcher: Feedback-driven data provider for object detection. *arXiv* **2020**, arXiv:2004.12432.
31. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z. Scale-aware trident networks for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6054–6063.
32. Zhang, G.J.; Lu, S.J.; Zhang, W. CAD-Net: A context-aware detection network for objects in remote sensing imagery. *IEEE Trans Geosci Remote Sens* **2019**, *57*, 10015–10024. [CrossRef]
33. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2Det: A single-shot object detector based on multi-level feature pyramid network. In Proceedings of the AAAI Conference on Artificial Intelligence, Atlanta, GA, USA, 8–12 October 2019; Volume 33, pp. 9259–9266.
34. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of The IEEE/CVF Conference on Computer Vision And Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.
35. Liu, S.; Huang, D.; Wang, Y. Learning spatial fusion for single-shot object detection. *arXiv* **2019**, arXiv:1911.09516.
36. Gong, Y.; Yu, X.; Ding, Y.; Peng, X.; Zhao, J.; Han, Z. Effective fusion factor in FPN for tiny object detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, Virtual Conference, 5–9 January 2021; pp. 1160–1168.

37. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv* **2020**, arXiv:2004.10934. [CrossRef]
38. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
39. Hou, Q.; Zhou, D.F. Coordinate attention for efficient mobile network design. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
40. 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; pp. 3–19.
41. Zhu, P.F.; Wen, L.; Du, D.; Bian, X.; Ling, H.; Hu, Q.; Nie, Q.; Cheng, H.; Liu, C.; Liu, X.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea, 27–28 October 2019.
42. Yu, W.P.; Yang TJ, N.; Chen, C. Towards Resolving the Challenge of Long-tail Distribution in UAV Images for Object Detection. In Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Virtual, 5–9 January 2021; IEEE Press: Waikoloa, HI, USA, 2021; pp. 3257–3266.
43. Xiaojun, L.; Wei, X.; Yunpeng, L. Small target detection algorithm for UAV aerial imagery based on enhanced underlying features. *Comput. Appl. Res.* **2021**, *38*, 1567–1571. [CrossRef]
44. Ali, S.; Siddique, A.; Ateş, H.F.; Güntürk, B.K. Improved YOLOv4 for aerial object detection. In Proceedings of the 2021 29th Signal Processing and Communications Applications Conference (SIU), Istanbul, Turkye, 9–11 June 2021.
45. Zhao, H.; Zhou, Y.; Zhang, L.; Peng, Y.; Hu, X.; Peng, H.; Cai, X. Mixed YOLOv3-LITE: A Lightweight Real-Time Object Detection Method. *Sensors* **2020**, *20*, 1861. [CrossRef] [PubMed]