A Multi-Scale Object Detector Based on Coordinate and Global Information Aggregation for UAV Aerial Images

Liming Zhou , Zhehao Liu , Hang Zhao , Yan-e Hou * , Yang Liu , Xianyu Zuo , Lanxue Dang

Posted Date: 29 May 2023

doi: 10.20944/preprints202305.1967.v1

Keywords: UAV images; multi-feature fusion; information aggregation; multi-scale object detection

Article

A Multi-Scale Object Detector Based on Coordinate and Global Information Aggregation for UAV Aerial Images

Liming Zhou [1,2], Zhehao Liu [1,2], Hang Zhao [1,2], Yan-e Hou [1,2,*], Yang Liu [1,2,3], Xianyu Zuo [1,2] and Lanxue Dang [1,2]

[1] Henan Key Laboratory of Big Data Analysis and Processing, Henan University, Kaifeng, Henan, China; lmzhou@henu.edu.cn (L.Z.);liuzhehao@henu.edu.cn (Z.L.);bless0929@henu.edu.cn (H.Z.);houyane@henu.edu.cn (Y.H.); sea@vip.henu.edu.cn (Y.L.); xianyu_zuo@henu.edu.cn (X.Z.); danglx@vip.henu.edu.cn (L.D.);

[2] School of Computer and Information Engineering, Henan University, Kaifeng, Henan, China

[3] Henan Province Engineering Research Center of Spatial Information Processing and Shenzhen Research Institute, Henan University, Kaifeng 475004, China

* Correspondence: houyane@henu.edu.cn

Abstract: Unmanned Aerial Vehicles (UAVs) image object detection has great application value in military and civilian fields. However, the objects in the captured images from UAVs have problems of large scale variation, complex backgrounds, and a large proportion of small objects. To resolve these problems, a multi-scale object detector based on coordinate and global information aggregation is proposed, named CGMDet. Firstly, a Coordinate and Global Information Aggregation Module (CGAM) is designed by aggregating local, coordinate, and global information, which can obtain features with richer context information. Secondly, a Multi-Feature Fusion Pyramid Network (MF-FPN) is proposed, which can better fuse features of different scales and obtain features containing more context information through repeated use of feature maps, to better detect multi-scale targets. Moreover, more location information of low-level feature maps is integrated to improve the detection results of small targets. Furthermore, we modified the bounding box regression loss of the model to make the model more accurately regress the bounding box and faster convergence. Finally, the proposed CGMDet was tested on VisDrone and UAVDT datasets and mAP0.5 of 50.9% and 48% was obtained, respectively. At the same time, our detector achieved the best results compared to other detectors.

Keywords: UAV images; multi-feature fusion; information aggregation; multi-scale object detection

1. Introduction

UAV application technology has also made great progress in recent years. Due to the advantages of good mobility, convenient use, and low cost, UAVs have extremely high application value in disaster monitoring[1], geological investigation[2], air traffic control[3], emergency relief[4], and other aspects. Therefore, UAV image object detection has been paid more attention by researchers. However, as the shooting angle and height of UAV are changeable, UAV object detection faces the following two challenges: (1) Due to the problem of UAV shooting perspective, there are large differences between the scales of targets of the same category or different categories, and there are many small objects, which greatly tests the performance of the model to detect multi-scale targets and small targets. (2) In UAV images, there are usually a lot of objects blocked, and weak light leads to the boundary and features of the object are not obvious, so it is hard to extract discriminant features from the model.

With the development of deep learning, traditional object detection methods, such as HOG[5] and SIFT[6], are gradually eliminated due to the need for a large amount of prior knowledge. However, the object detection method based on deep learning does not need manual involvement and can dig deeper and more abstract features. Although the object detector based on deep learning has achieved

great results in the detection of natural images, there are still great challenges for the object detection of UAV images.

To resolve these problems, this paper proposed a multi-scale object detector based on coordinate and global information aggregation, named CGMDet. Firstly, we designed a Coordinate and Global Information Aggregation Module (CGAM), which can make the model focus on the coordinate information and global information, to alleviate the interference brought by the background. Secondly, a Multi-Feature Fusion Pyramid Network (MF-FPN) is proposed, which can better fuse features of different scales, and add feature maps of larger sizes to feature fusion, to detect multi-scale objects more effectively, especially small objects. In addition, the number of convolutional channels in the neck is reduced to decrease the number of parameters required by the network. Finally, we modified the bounding box regression loss to enable the model to more accurately regress bounding boxes. This modification allows high-quality anchors to contribute more gradients to the training process. Therefore, the model can achieve better detection accuracy and faster convergence speed.

In summary, the contributions of this study are as follows:

1. We proposed a multi-scale object detector based on coordinate and global information aggregation for UAV aerial images, which achieves better performance without increasing the model parameters;
2. To better locate the object and alleviate the interference caused by background factors, the CGAM is proposed. The module can capture local information, coordinate information, and global contextual information, and fuse them to reduce the interference of background factors on the feature extraction process, thereby obtaining more robust feature information;
3. To enable the model to better detect multi-scale and small objects, the MF-FPN is proposed. This structure better fuses features of different scales by reusing multi-scale features repeatedly to obtain feature maps with richer contextual information, thereby improving the ability of the model to detect multi-scale objects. Additionally, the feature map of larger size is added to the feature fusion structure enabling the model to better detect small objects;
4. To regress bounding boxes more accurately, we modified the bounding box loss to obtain high-quality prediction boxes and make the model converge faster;
5. We validated our CGMDet on two public UAV image datasets. The results illustrate that our model achieved great results on both datasets. In addition, our model was compared with others and achieved the best results.

2. Related Work

2.1. Object Detection

At present, the commonly used detectors are one-stage and two-stage detectors. Among them, the first step of the two-stage detector is to generate candidate regions, and the second step is to classify and regress each candidate region. The one-stage detector performs classification and regression directly. Classic two-stage detectors include R-CNN[7], Fast R-CNN[8], Faster R-CNN[9], SPP-Net[10], etc. The accuracy of the two-stage detector is higher than that of the one-stage detector, but the speed is slower than the one-stage detector. The commonly used one-stage detectors currently include SSD[11], RetinaNet[12], YOLO series[13–18], etc. Recently, some anchor-free detectors have been invented. The anchor-free method uses the features of object centers or key points to replace the complex anchor design. For example, FCOS[19] treats each pixel on the feature map as a training sample and uses a four-dimensional vector to regress the predicted box. CenterNet[20] represents objects using their center points and predicts bounding boxes by predicting the offset of the center point and the width and height of the object. The above detector has achieved good results in natural images. But, for UAV images, existing detection methods still face significant challenges.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

2.2. Attention Mechanism

Currently, attention mechanisms are widely used because they allow the model to focus more on important information and ignore unimportant ones. The attention mechanism has played a significant role in object detection. The Squeeze-and-excitation block (SE)[21] is a classic channel attention mechanism that can apply a weight to each channel of the feature, allowing the model to focus more on the important channel information. However, to save computation, SE performs a squeeze operation during the processing, which can result in the loss of some channel information. To avoid losing channel information, Efficient Channel Attention Module (ECA)[22] and Effective Squeeze-and-excitation Block (ESE Block)[23] have been proposed. To avoid channel information loss, the ECA uses a 1D convolution operation instead of the two fully connected layers in SE. The ESE Block removes the squeeze operation and uses one fully connected layer instead of the two fully connected layers in SE. In addition to channel attention mechanisms, Coordinate Attention (CA)[24] obtains feature maps integrated with spatial coordinate information by performing adaptive average pooling algorithms along the x and y directions of the feature map, respectively. Furthermore, the CBAM[25] introduces both channel and spatial attention mechanisms to allocate different weights for different channels and spatial regions, respectively, to obtain highly responsive feature information and improve network performance.

To better extract contextual feature information, we designed a Coordinate and Global Information Aggregation Module (CGAM), which can extract the local feature and continuously focus on the coordinate information. In addition, it combines global information to obtain features with richer contextual information.

2.3. Multi-Scale Feature Fusion

Targets in unmanned aerial vehicle (UAV) images have characteristics such as large-scale variations and a high proportion of small objects, which pose significant challenges to object detection tasks. In deep networks, low-level feature maps typically contain rich positional information, while high-level features typically contain rich semantic information. Therefore, for better performance, low-level and high-level features are usually fused. FPN[26] fuses features of adjacent scales through a top-down path and horizontal connections. PANet[27] proposes a bidirectional fusion structure that combines features in both top-down and bottom-up directions. Zhao et al.[28] proposed MLFPN extracts more representative multi-level and multi-scale features through the TUM and FFM, and then integrates features through the SFAM to obtain features with rich contextual information. Tan et al.[29] proposed BiFPN, which adds lateral skip connections to the top-down and bottom-up pathways and assigns a learnable weight to each feature map during fusion to emphasize the importance of different feature maps for better feature fusion.

To obtain features with richer context information, we designed a Multi-Feature Fusion Pyramid Network (MF-FPN) in this paper. MF-FPN utilizes feature maps repeatedly and assigns a learnable weight for each feature map to obtain features with richer contextual information.

3. Methods

This paper proposed a multi-scale object detector based on coordinate and global information aggregation for UAV aerial images, named CGMDet. Firstly, we designed a Coordinate and Global Information Aggregation Module (CGAM), which allows the backbone to more focus on the coordinate information and global context information during the feature extraction process, to enhance the ability of the network to extract features. Then, a Multi-Feature Fusion Pyramid Network (MF-FPN) was proposed to better fuse multi-scale features. Finally, we modified the bounding box loss to obtain better detection results. Figure 1 shows the overall architecture of CGMDet. Our CGMDet can accurately detect targets with mixed backgrounds and take into account the detection performance of targets with different scales.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

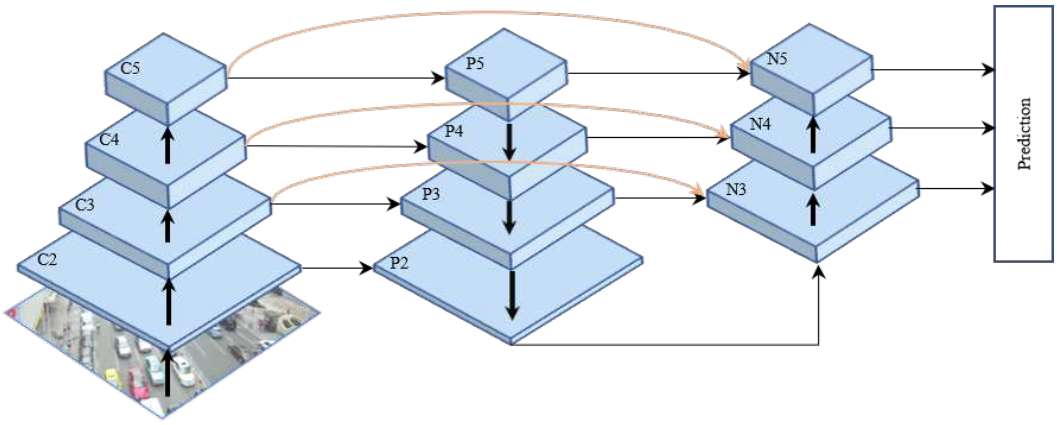


Figure 1. The overall architecture of CGMDet.

First, a 640 × 640 image is input into the backbone to extract features. The image passes through four CBS modules, which consist of convolution, Batch Normalization[30], and SiLU[31] activation function. In the first and third CBS modules, the convolution has a stride of 1, while in the second and fourth modules, it has a stride of 2. After obtaining feature maps of four times downsampling, the CGAM module extracts features of different sizes. SPPCSPC[18] aims to aggregate features with different receptive fields to obtain richer semantic contextual information. Then, different scales of feature maps are fed into the neck, where our proposed MF-FPN method is used to fuse features of different scales. Because most targets in UAV images are small, and low-level features are better for small target detection, we also input feature maps of size 160 × 160 into the feature fusion. Therefore, the feature map sizes that need to be fused in the neck are 160 × 160, 80 × 80, 40 × 40, and 20 × 20. Finally, feature maps of size 80 × 80, 40 × 40, and 20 × 20 are used for object detection. Figure 2 shows the detailed network structure of our CGMDet.

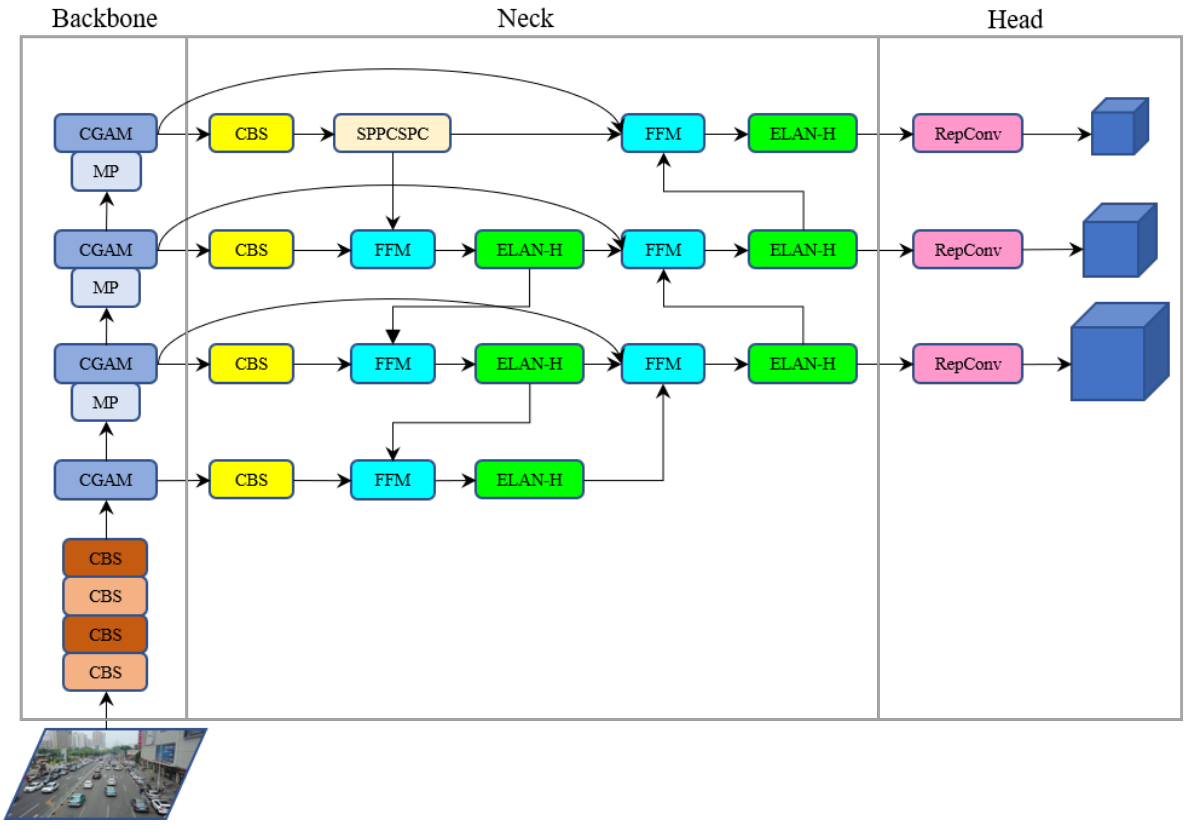


Figure 2. The detailed architecture of CGMDet.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

3.1. Coordinate and Global Information Aggregation Module

Because of the camera angle of the UAV, there are often many cases where targets are occluded. At the same time, the light is dimmer in nighttime scenes. The feature extraction process in these two scenarios is easily disturbed by background factors, which is easy to cause missed detections and false detections. To alleviate the interference of background factors, we design a Coordinate and Global Information Aggregation Module (CGAM), which can integrate global information, coordinate information, and local information extracted by convolution, to obtain more robust features. The structure of CGAM is shown in Figure 3.

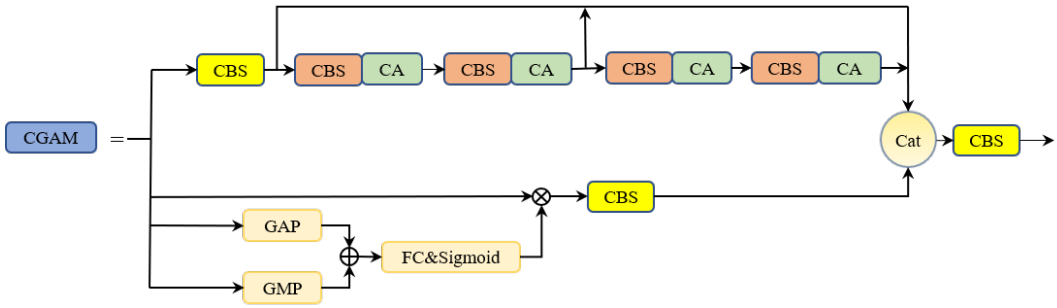


Figure 3. Coordinate and Global Information Aggregation Module.

The CGAM consists of two branches. The first branch introduces the coordinate attention mechanism, which constantly focuses on the coordinate information when using convolution for feature extraction. The second branch obtains global information on the feature map through two pooling operations. By fusing the features extracted from the two branches, richer contextual features are obtained.

The first branch of CGAM first uses a 1×1 convolution to reduce the number of channels in the input feature $X \in \mathbb{R}^{C \times H \times W}$ by half, obtaining the first intermediate feature map $M_1 \in \mathbb{R}^{\frac{C}{2} \times H \times W}$. As shown in formula (1):

$$M_1 = Conv_{1\times1}(F) \tag{1}$$

The features map is then extracted using 3×3 convolution and the coordinate attention mechanism, obtaining the second and third intermediate output feature maps $M_2, M_3 \in \mathbb{R}^{\frac{C}{2} \times H \times W}$. As shown in formulas (2) and (3):

$$M_2 = CA\left(Conv_{3\times3}\left(CA\left(Conv_{3\times3}\left(M_1\right)\right)\right)\right) \tag{2}$$

$$M_3 = CA\left(Conv_{3\times3}\left(CA\left(Conv_{3\times3}\left(M_2\right)\right)\right)\right) \tag{3}$$

where CA represents the coordinate attention mechanism. The coordinate attention is shown in Figure 4.

The coordinate attention mechanism first performs pooling operations on the input feature $F \in \mathbb{R}^{C \times H \times W}$ along the horizontal and vertical directions, obtaining features $f^h \in \mathbb{R}^{C \times H \times 1}$ and $f^w \in \mathbb{R}^{C \times 1 \times W}$. As shown in formulas (4) and (5):

$$f_c^h = \frac{1}{W} \sum_{0 \le i \le W} F_c(h, i) \tag{4}$$

$$f_c^w = \frac{1}{H} \sum_{0 \le j \le H} F_c(j, w) \tag{5}$$

where F_c and f_c represent the c-th channel of the input and output features, respectively. W and H represent the width and height of the input feature, respectively. Then, f^h and f^w are concatenated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

along the spatial dimension, and the number of channels is reduced using a 1×1 convolution. Further, feature $Q \in \mathbb{R}^{\frac{C}{r} \times 1 \times (W+H)}$ is obtained by passing it through batch normalization and an activation function, where r is a scaling factor. Batch normalization is used to prevent gradient explosion or vanishing, making the model more stable during training, and the activation function introduces non-linear factors to enhance the expression ability of the model. The formula is shown as (6):

$$Q = \delta\left(BN\left(Conv_{1\times1}\left([f^h, f^w]\right)\right)\right) \tag{6}$$

where $[\cdot]$ denotes the channel concatenation operation. BN denotes batch normalization. δ represents a non-linear activation function. Then, the feature tensor Q is split along the spatial dimension to obtain two feature tensors $y^h \in \mathbb{R}^{\frac{C}{r} \times H \times 1}$ and $y^w \in \mathbb{R}^{\frac{C}{r} \times 1 \times W}$. Increase the number of channels for y^h and y^w to the same as the input feature map F by 1×1 convolution, and then the attention weights g^h and g^w are obtained by the sigmoid function. The formulas are shown as (7) and (8):

$$g^h = \sigma(Conv_{1\times1}(y^h)) \tag{7}$$

$$g^w = \sigma(Conv_{1\times1}(y^w)) \tag{8}$$

where σ denotes the sigmoid function. Finally, g^h and g^w are multiplied by the feature map F. As shown in formula (9):

$$CA = F \otimes g^h \otimes g^w \tag{9}$$

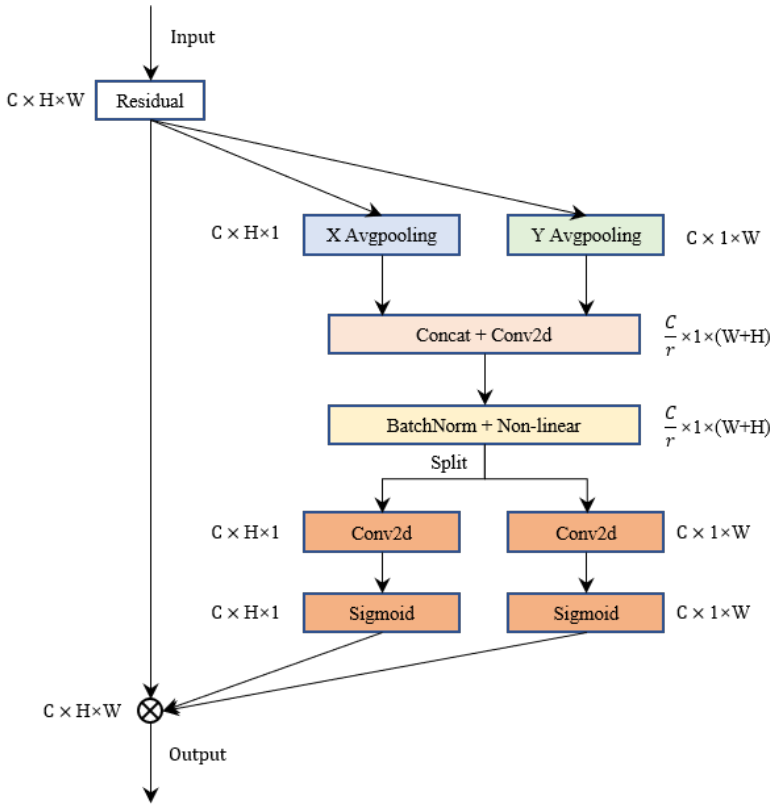


Figure 4. Coordinate Attention.

The second branch of the CGAM module first uses global pooling operations to add the global contextual information of the backbone network. For the input feature $X \in \mathbb{R}^{C \times H \times W}$, perform global average pooling and global maximum pooling operations first, then add the results, and finally allocate weights for each channel through a fully connected layer and a sigmoid function, making the model focus on the highly responsive channel information. As shown in formula (10):

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

$$O = \sigma\left(FC\left(GAP\left(X\right) \oplus GMP\left(X\right)\right)\right) \tag{10}$$

where GAP and GMP represent global average pooling and global maximum pooling, respectively. σ represents the sigmoid function. FC represents a fully connected layer. Then, multiply the result with the input feature X and use a 1×1 convolution to get the output $M_4 \in \mathbb{R}^{\frac{C}{2} \times H \times W}$ of the second branch. As shown in formula (11):

$$M_4 = Conv_{1\times1}\left(X \otimes O\right) \tag{11}$$

The CGAM module first performs channel concatenation on all intermediate output features M_1, M_2, M_3, and M_4 from the two branches, and then uses a 1×1 convolution to obtain the output feature $Z \in \mathbb{R}^{2C \times H \times W}$ of CGAM. As shown in formula (12):

$$Z = Conv_{1\times1}\left(\left[M_1, M_2, M_3, M_4\right]\right) \tag{12}$$

Our CGAM module can extract coordinate information, global information, and local information simultaneously and fuse them to obtain more robust features, thereby accurately locating the target, reducing the focus of the model on the background, and improving the detection ability of the model.

3.2. Multi-Feature Fusion Pyramid Network

The object scale changes greatly in UAV images, and there are many small objects. To enhance the ability of the network to detect multi-scale targets, the Multi-Feature Fusion Pyramid Network (MF-FPN) is proposed. As shown in Figure 1, we added a fusion path with a skip connection in the neck. Our MF-FPN structure fuses feature maps by reusing them repeatedly, to obtain more contextual information. To alleviate the difficulty in detecting small objects, a feature map of size 160×160 is added to the MF-FPN structure for fusion. The feature fusion method of MF-FPN uses the Feature Fusion Module (FFM) shown in Figure 5.

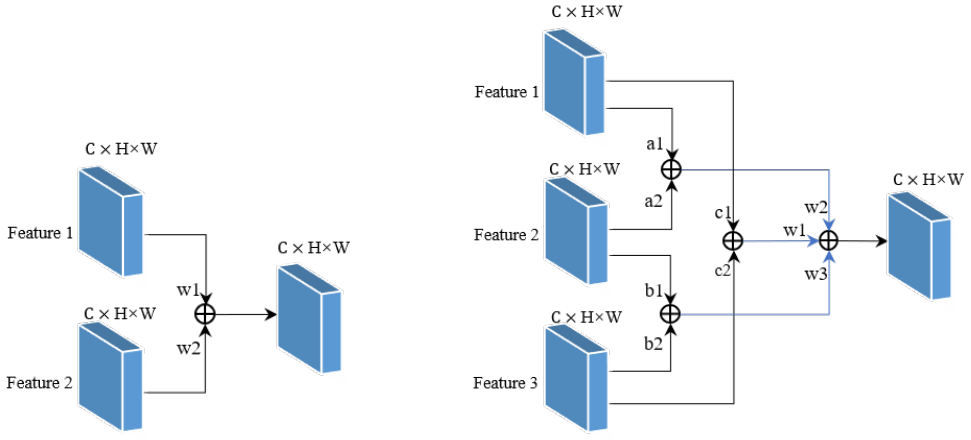


Figure 5. The structure of the Feature Fusion Module (FFM).

There are two main ways to fuse features. If only two feature maps need to be fused, such as the top-down path in the neck, the method shown in Figure 5(a) is used. This method assigns two learnable weights to the two feature maps to determine the importance of each feature map, as shown in formula (13):

$$P = \frac{w_1 F_1 + w_2 F_2}{w_1 + w_2 + \Delta} \tag{13}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

where w_1 and w_2 are learnable parameters, and Δ is a small number to avoid numerical instability. For the case of fusing three feature maps, we use the method shown in Figure 5(b). The calculation method is shown in formula (14). We first use formula (13) to fuse the three feature maps pairwise, where each feature map participates in two fusions, achieving the effect of reusing features. Then, we obtain three different intermediate feature maps and finally assign three learnable weights to fuse these three feature maps, obtaining the output feature map with rich contextual information for the final prediction.

$$N = \frac{w_1 P\left(F_1, F_2\right) + w_2 P\left(F_1, F_3\right) + w_3 P\left(F_2, F_3\right)}{w_1 + w_2 + w_3 + \Delta} \tag{14}$$

Since the fusion features contain feature maps with different scales and channel numbers, it is necessary to adjust the size and number of channels of the feature maps to be consistent before fusion.To preserve more feature information, the convolutional channels in the model are usually large, and larger channels bring more parameters to the model. The number of parameters of convolution can be calculated by the formula (15).

$$Params = K_h \times K_w \times C_{in} \times C_{out} \tag{15}$$

where K_w and K_h represent the size of the convolutional kernel. C_{in} and C_{out} represent the number of input and output channels of the convolution, respectively. Therefore, to decrease the parameters of the model, we modified the convolutional channel numbers in the neck of the model. First, the channel numbers of the 3×3 convolutions in the three ELAN-H modules in the top-down path were adjusted to 32. Then, the output channel numbers of the first two 1×1 convolutions in the ELAN-H modules in the bottom-up path were adjusted to $1/4$ of the input channel numbers. The changes in the number of parameters for all ELAN-H modules in the neck are shown in Table 6 in Section 4.3.

The pseudo-code of the MF-FPN is shown in Algorithm 1. We first fuse adjacent two features in $X = \{x_1, x_2, x_3, x_4\}$ from top to bottom to obtain four intermediate feature maps $M = \{m_1, m_2, m_3, m_4\}$. Then, the features from X and M are fused using the bottom-up path and skip connections to obtain three final features of different scales, denoted as $Y = \{y_1, y_2, y_3\}$, which will be used for prediction.

Algorithm 1 The feature fusion method of MF-FPN.

Input: $\mathbf{X} = \{x_1, x_2, x_3, x_4\}$, X refers to four different scale feature maps of backbone network output. The scale of x_1 is the smallest and x_4 is the largest.
Step 1: $\mathbf{M} = \{\}$, **M** refers to the intermediate feature map generated by the top-down branch of MF-FPN. $Conv\left(\right)$ represents a series of convolution operations required, $Reshape\left(\right)$ represents upsampling and downsampling operations required, and $FFM\left(\right)$ represents our feature fusion operation.
  for $i = 1$ to 4 **do**
    if $i = 1$ **then**
      $m_i = Conv\left(x_i\right)$
    else
      $m_i = Conv\left(FFM\left(x_i, Reshape\left(x_{i-1}\right)\right)\right)$
    end if
    $\mathbf{M}.append\left(m_i\right)$
  end for
Step 2: $\mathbf{Y} = \{\}$, **Y** refers to the feature map generated by the bottom-up branch of MF-FPN for use in prediction.
  for $i = 1$ to 4 **do**
    $y_i = Conv\left(FFM\left(x_i, M_i, Reshape\left(M_{i+1}\right)\right)\right)$
    $\mathbf{Y}.append\left(y_i\right)$
  end for
Output: Return **Y**.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

3.3. Loss Function

The CIOU[32] loss is commonly used as the bounding box regression loss in existing models. The definition of CIOU is as follows:

$$L_{CIOU} = 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \alpha v \tag{16}$$

$$IOU = \frac{|B \cap B^{gt}|}{|B \cup B^{gt}|} \tag{17}$$

where ρ represents the Euclidean distance. b and b^{gt} denote the center points of the predicted and ground truth box, respectively. c represents the diagonal length of the minimum bounding rectangle of the ground truth box and predicted box. v and α are defined as follows:

$$v = \frac{4}{\pi^2}\left(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}\right)^2 \tag{18}$$

$$\alpha = \frac{v}{(1 - IOU) + v} \tag{19}$$

where h^{gt} and w^{gt} are the height and width of the ground truth box. h and w are the height and width of the predicted box.

But regressing the height and width of the bounding box accurately cannot be achieved only through the aspect ratio. Because when $w = kw^{gt}$ and $h = kh^{gt}$ $(k \in R^+)$, $v = 0$. The EIOU loss[33] not only retains the advantages of the CIOU but also minimizes the differences between the height and width of the predicted and ground truth boxes, resulting in better localization performance. The definition of EIOU is shown in formula (20).

$$L_{EIOU} = 1 - IOU + \frac{\rho^2\left(b, b^{gt}\right)}{c^2} + \frac{\rho^2\left(h, h^{gt}\right)}{h^c} + \frac{\rho^2\left(w, w^{gt}\right)}{w^c} \tag{20}$$

where h^c and w^c are the height and width of the minimum bounding box surrounding the ground truth and predicted box. EIOU can directly regress the height and width of the prediction box. Furthermore, to make the model converge faster, we use the Focal EIOU loss[33], which combines the Focal loss with the EIOU loss, as the bounding box regression loss for CGMDet. It allows high-quality anchors to contribute more gradients to the training process, thereby improving the convergence speed of the model. Its definition is as follows:

$$L_{Focal-EIOU} = IOU^{\lambda} L_{EIOU} \tag{21}$$

where λ is an adjustable parameter, which we set to 0.5. Additionally, the confidence and classification losses of the model are calculated using Binary Cross Entropy with Logits Loss (BCEWithLogitsLoss)[34]. The definition is as follows:

$$L_{BCE} = -\sum_{n=1}^{N} \hat{y}_i \log\left(\sigma\left(y\right)\right) + \left(1 - \hat{y}_i\right) \log\left(\sigma\left(1 - y\right)\right) \tag{22}$$

where N is the number of input vectors. $\hat{y}_i$ and y are the predicted and truth vectors, respectively. σ is the sigmoid function. The overall loss of CGMDet can be obtained by combining the classification loss, confidence loss, and bounding box regression loss. The definition is as follows:

$$\begin{aligned} Loss &= \lambda_1 L_{box} + \lambda_2 L_{obj} + \lambda_3 L_{cls} \\ &= \lambda_1 L_{Focal-EIOU}(P_{box}, T_{box}) + \lambda_2 L_{BCE}(P_{obj}, T_{obj}) + \lambda_3 L_{BCE}(P_{cls}, T_{cls}) \end{aligned} \tag{23}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

where L_{box}, L_{obj}, and L_{cls} represent the bounding box regression loss, confidence loss, and classification loss, respectively. P_{box} and T_{box} denote the predicted and ground truth box, respectively. P_{obj} and T_{obj} denote the predicted and truth confidence, respectively. And P_{cls} and T_{cls} represent the predicted and truth class probability, respectively. The hyperparameters λ_1, λ_2, and λ_3 are set to 0.05, 0.7, and 0.3 by default.

4. Experiments

To verify the effectiveness of our detector, we conducted experiments on two publicly available UAV image datasets and compared it with other detectors.

4.1. Datasets

The VisDrone benchmark[35] contains 10,209 static images, of which 6,471 images are used for training, 3,190 for testing, and 548 for validation. The resolution of the images is approximately 2000×1500 pixels, collected by various drone platforms in different scenarios, as well as under different weather and lighting conditions. Figure 6 shows some images from this dataset.

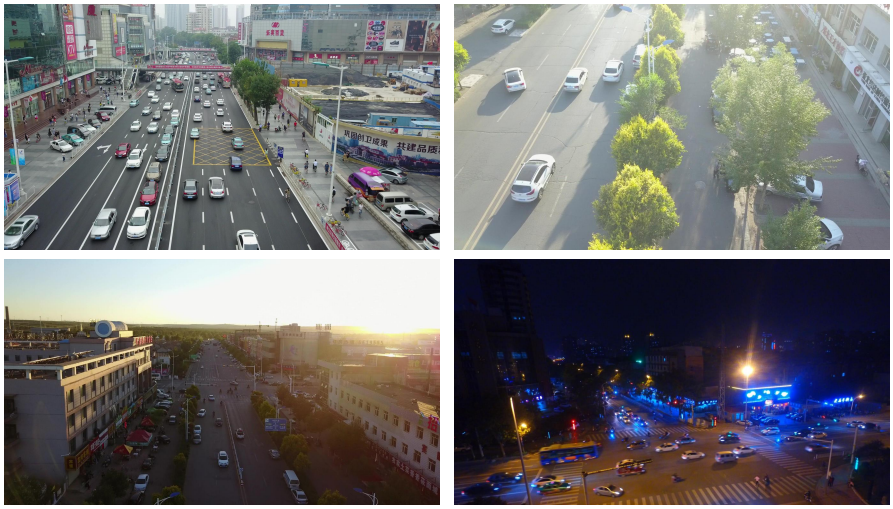


Figure 6. Some examples in VisDrone.

The UAVDT benchmark[36] consists of 40,735 images, of which 24,206 images are used for training and 16,529 for validation. This dataset contains images with different weathers, flight heights, shooting angles, and occlusion scenes. The images in the dataset have a resolution of approximately 1080×540 pixels. The dataset includes three predefined classes: car, truck, and bus. Figure 7 shows some images from this dataset.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

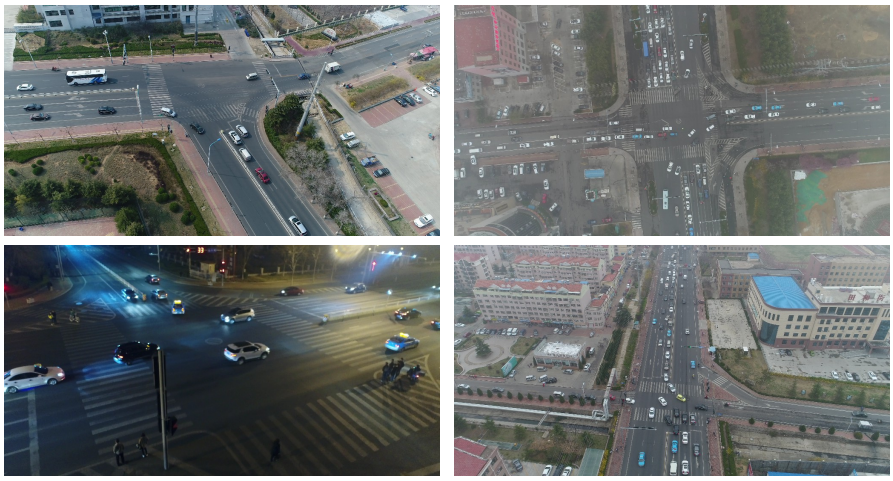


Figure 7. Some examples in UAVDT.

4.2. Implementation and Evaluation Criteria

4.2.1. Implementation

This paper validated the proposed object detector on Ubuntu 18.04.6 LTS system, trained and tested on NVIDIA GeForce RTX 3090 (24G) as the graphics processing unit, with an Intel(R) Xeon(R) Silver 4114 CPU @2.20GHz and Python version 3.6. The CUDA version used was 11.7 and the PyTorch version used was 1.10.2.

During model training, the input image size was set to 640×640, and Stochastic Gradient Descent (SGD) optimizer with momentum was used. The initial learning rate was set to 0.01, the momentum parameter was set to 0.937, the weight decay coefficient was set to 0.0005, and the batch size was set to 8. The total number of training iterations for the VisDrone dataset was 300, and for the UAVDT dataset was 200.

4.2.2. Evaluation Criteria

Precision P, Recall R, Average Precision AP, and mean Average Precision mAP are used as metrics to evaluate the performance of our detector. P represents how many predicted positive samples are correct. R represents how many positive samples are predicted. The definitions of P and R are as follows:

$$P = \frac{TP}{TP + FP} \tag{24}$$

$$R = \frac{TP}{TP + FN} \tag{25}$$

where TP represents how many samples were correctly predicted to be positive. FP represents how many samples were incorrectly predicted to be positive. FN represents how many samples were incorrectly predicted to be negative. P and R usually are trade-offs between each other. Therefore, AP can better measure the detection capability of the network. The definition of AP is shown in formula (26):

$$AP = \int_0^1 P(R)dR \tag{26}$$

where $P(R)$ represents the precision value when the recall value on the P-R curve is R. mAP is the average of all class AP values, which can represent the average detection performance of the detector on the dataset. The definition is shown in formula (27):

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

$$mAP = \frac{1}{K}\sum_{i=1}^{K} AP_i \tag{27}$$

where AP_i denotes the AP value of the i-th category. K denotes the number of target categories.

To better describe the ability of our detector to detect multi-scale objects, we also used COCO evaluation metrics[37], such as AP_S, AP_M, and AP_L. AP_S represents the AP value of small objects with an area less than 32×32. AP_M represents the AP value of medium objects with an area between 32×32 and 96×96. AP_L represents the AP value of large objects with an area greater than 96×96.

4.3. Experimental Results

4.3.1. Experimental Results on the VisDrone Dataset

We evaluated our model using the VisDrone dataset and compared our model with other models to validate its effectiveness. As shown in Table 1, for the VisDrone dataset, our detector outperforms other detectors and achieves the best results. Compared with the YOLOv7[18], our model achieved an improvement of 1.9% in mAP0.5, and an improvement of 1.6% and 1.2% in mAP0.75 and mAP, respectively. The detection performance of our detector for small and medium targets has been greatly improved. The AP_S has increased by 1.3% compared to the YOLOv7, and the AP_M has increased by 1.2%. Although the AP_L has decreased by 0.4%, we believe that the benefits of our proposed model outweigh its drawbacks in multi-scale object detection. Compared with QueryDet, although our model only improved by 0.6% in mAP0.75, mAP and mAP0.5 have improved by 1% and 2.8%, respectively. Compared to RetinaNet, Cascade-RCNN, Faster-RCNN, YOLOv3, YOLOX, YOLOv5l, and HawkNet, our CGMDet obtained the best results in mAP0.5, mAP0.75, and mAP, as well as AP_S, AP_M, and AP_L.

Table 1. Comparison with state-of-the-art detectors on the VisDrone dataset.

Method	mAP0.5	mAP0.75	mAP	AP_S	AP_M	AP_L
RetinaNet[12]	35.9	18.5	19.4	14.1	29.5	33.7
Cascade R-CNN[38]	39.9	23.4	23.2	16.5	36.8	39.4
Faster R-CNN[9]	40.0	20.6	21.5	15.4	34.6	37.1
YOLOv3[15]	31.4	15.3	16.4	8.3	26.7	36.9
YOLOX[39]	45.0	26.6	26.7	17.4	37.9	45.3
YOLOv5l[17]	36.2	20.1	20.5	12.4	29.9	36.4
HawkNet[40]	44.3	25.8	25.6	19.9	36.0	39.1
QueryDet[41]	48.1	28.8	28.3	\	\	\
YOLOv7[18]	49.0	27.8	28.1	18.9	39.4	**47.8**
CGMDet(Ours)	**50.9**	**29.4**	**29.3**	**20.2**	**40.6**	47.4

We also listed the mAP0.5 for each category to describe in more detail which categories our model has improved on. As shown in Table 2, our model has a higher mAP0.5 than other models for each category. In addition, except for the tricycle category, which has the same result as YOLOv7, all other categories have greatly improved, especially the bus and bicycle categories, which have increased by 3.3% and 3.8%, respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

Table 2. Detection results for each category on the VisDrone dataset.

Method	pedestrian	people	bicycle	car	van	truck	tricycle	awing-tricycle	bus	motor	mAP0.5
YOLOv3 [15]	12.8	7.8	4.0	43.0	23.5	16.5	9.5	5.1	29.0	12.5	31.4
YOLOv5l [17]	44.4	36.8	15.6	73.9	39.2	36.2	22.6	11.9	50.5	42.8	37.4
YOLOv7 [18]	57.6	48.7	21.6	85.4	51.9	45.8	37.9	18.3	63.0	60.0	49.0
CGMDet (Ours)	**59.7**	**50.7**	**25.4**	**86.2**	**53.4**	**47.4**	**37.9**	**20.2**	**66.3**	**61.6**	**50.9**

To make it more deployable on mobile devices, we also designed a tiny version of the model and conducted experiments. The inference time was obtained by calculating the average prediction time of all images in the test set. The results in Table 3 show that our CGMDet-tiny achieved the best results. Compared to YOLOv7-tiny, our CGMDet-tiny has achieved a 4% improvement in mAP0.5, 3.6% improvement in mAP0.75, and 3.1% improvement in mAP. Meanwhile, AP_S, AP_M, and AP_L have increased by 2.8%, 4.1%, and 1%, respectively. Moreover, our model has only increased by 1.4M parameters. Although our CGMDet-tiny is not as fast as other models in inference time, it can still detect in real time. Moreover, judging from the results of detection performance, it is worth trading inference time for detection accuracy.

Table 3. Comparison of detection results of tiny version on VisDrone dataset.

Method	mAP0.5	mAP0.75	mAP	AP_S	AP_M	AP_L	Params(M)	Inference Time(ms)
YOLOX-tiny[39]	35.7	19.2	19.7	12.2	28.3	31.7	5.0	\
YOLOv5s[17]	28.7	14.0	15.1	9.2	22.2	31.8	7.0	10.8
YOLO-UAVlite[42]	36.6	19.7	20.6	12.9	29.3	33.4	1.4	\
YOLOv7-tiny[18]	35.8	17.3	18.6	11.4	27.4	36.6	6.0	9.4
CGMDet-tiny(Ours)	**39.8**	**20.9**	**21.7**	**14.2**	**31.5**	**37.6**	7.4	21.2

To better illustrate the advantages of our model, we provide detection results of several images in different scenarios. As shown in Figure 8, (a1)-(a4) are the results of YOLOv7, and (b1)-(b4) are the detection results of our CGMDet. From the red dashed box in (a1) and (b1) of Figure 8, YOLOv7 recognized the text on the ground as a car, while our model can recognize it as the background. From Figure 8 (a2) and (b2), our model can also distinguish between two objects with very similar features that are close together. Due to the indistinct features of small targets, it is difficult for the model to learn, and it is easy to recognize similar backgrounds as targets. However, our improved model is better able to detect small targets and can effectively distinguish the background, as shown in Figure 8 (a3) and (b3). In addition, we also tested the detection performance in nighttime scenes, as shown in the red dashed box in Figure 8(a4) and (b4). YOLOv7 failed to detect it, while our model accurately marked it out.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

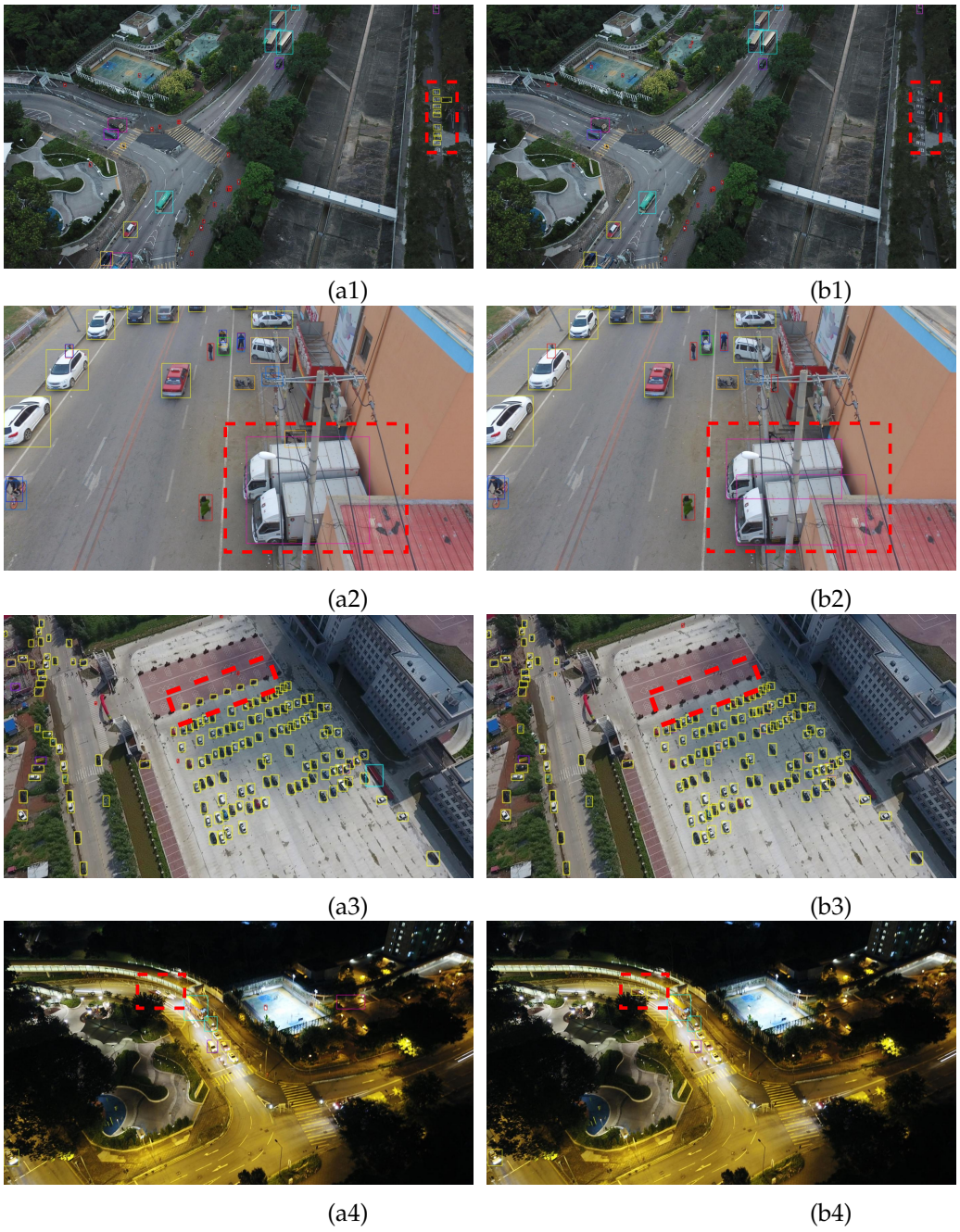


Figure 8. The detection results of the VisDrone dataset under different scenes. **(a1-a4)** are the detection results of YOLOv7; **(b1-b4)** are the results of the proposed model.

4.4. Experimental Results on the UAVDT Dataset

We also evaluated our model using the UAVDT dataset and compared our model with others. As shown in Table 4, compared with YOLOv7, our model has increased mAP0.5 by 3.0%, mAP0.75 by 3.2%, and mAP by 2.3%. The detection performance of YOLOv7 on the UAVDT dataset is worse than that of YOLOv5l. Compared with YOLOv5l, the mAP0.5 is 1.2% lower, the mAP0.75 is 1.3% lower, and the mAP is 1.1% lower. However, our model outperforms YOLOv5l in terms of mAP0.5, mAP0.75, and mAP. We also listed the mAP0.5 results for each category in the table, which showed that our model improved by 2.9% over YOLOv7 in the car category, and the results for the truck and bus categories both improved by 3.4%. In addition, our model outperforms YOLOv5l by 3.3% in the truck category

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

and is also 0.6% and 2.2% higher in the car and bus categories, respectively. At the same time, our model's performance is also superior to YOLOv3 and YOLOX.

In Table 4, we also included the results of the tiny version for comparison. Our CGMDet-tiny improved the results for the car, truck, and bus categories by 2.3%, 1.0%, and 0.9%, respectively, compared to YOLOv7-tiny. It also increased the mAP0.5 by 1.4% and improved the mAP0.75 and mAP by 2.2% and 1.8%, respectively. In addition, our CGMDet-tiny outperforms YOLOX-tiny in all metrics. However, compared to YOLOv5s, our CGMDet-tiny only outperforms by 0.3% in terms of mAP.

To illustrate the superiority of our CGMDet, we present detection results for several images in different scenarios. As shown in Figure 9, (a1)-(a4) are the results of YOLOv7, and (b1)-(b4) are the results of our proposed model. From Figure 9 (a1) and (b1), our model performs significantly better than YOLOv7 in detecting small targets. From Figure 9 (a2) and (b2), even in low light conditions at night, our model has a significant improvement over the baseline. In addition, as shown in Figure 9(a3) and (b3), YOLOv7 detected the left tree as a car, and the bus as a truck, and did not detect the objects that were truncated at the bottom of the image or slightly occluded on the right side of the image. In contrast, our model correctly recognized the tree as the background, accurately identified the object categories, and accurately detected the objects at the bottom and right of the image.

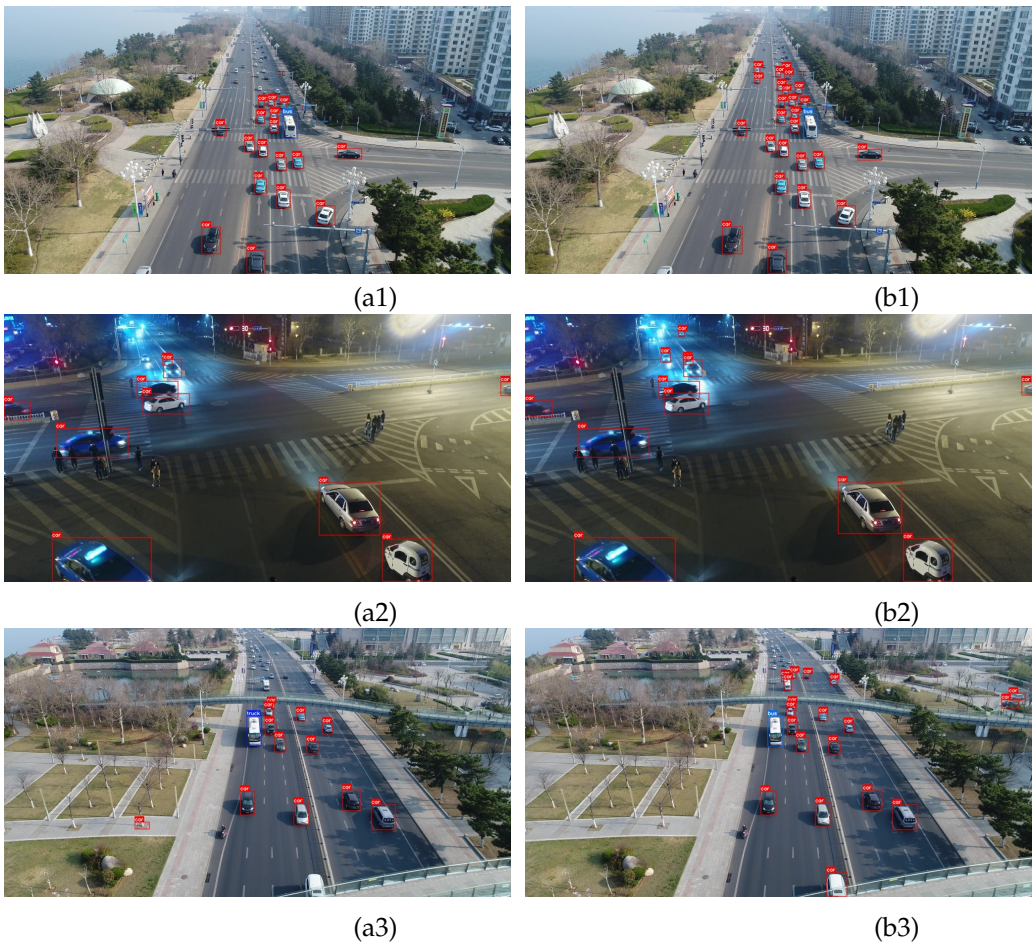


Figure 9. Detection results of different scenes in the UAVDT dataset. **(a1-a3)** are the detection results of YOLOv7; **(b1-b3)** are the detection results of our CGMDet.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

Table 4. Comparison with state-of-the-art detectors on the UAVDT dataset.

Method	car	truck	bus	mAP0.5	mAP0.75	mAP
YOLOv3[15]	30.8	3.9	26.4	36.3	20.6	20.4
YOLOX[39]	39.4	5.7	25.3	37.9	26.1	23.5
YOLOX-tiny[39]	40.5	0.5	22.0	36.1	22.7	21.0
YOLOv5l[17]	80.7	12.7	45.2	46.2	30.7	27.7
YOLOv5s[17]	78.1	13.3	45.6	45.0	28.6	26.8
YOLOv7[18]	78.4	12.6	44.0	45.0	29.4	26.8
YOLOv7-tiny[18]	75.5	6.9	46.7	43.0	24.3	25.0
CGMDet(Ours)	**81.3**	**16.0**	47.4	**48.0**	**32.6**	**29.1**
CGMDet-tiny(Ours)	77.8	7.9	**47.6**	44.4	26.5	26.8

4.5. Ablation Experiments

We used the VisDrone dataset to conduct ablation experiments for our model to verify the effectiveness of our improved methods. For fairness, all experimental settings have the same parameters and are conducted in the same environment. As shown in Table 5, we use YOLOv7 as the baseline and achieve a 49% mAP. And the results show that each improvement can enhance the detection ability of the model to some extent.

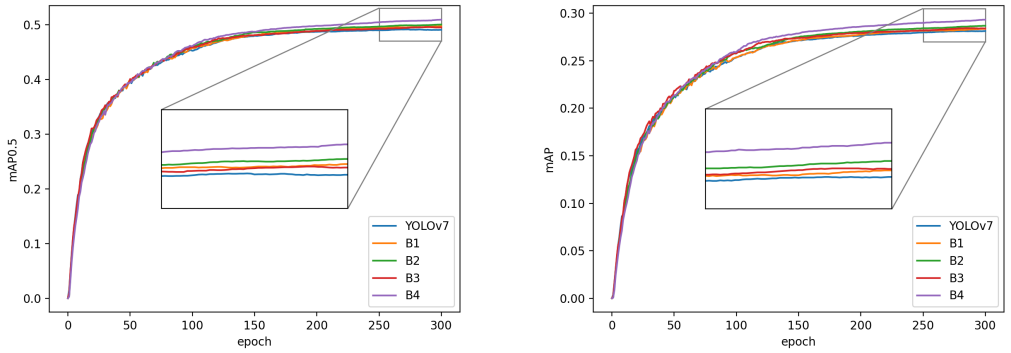
- **CGAM**: To reflect the effectiveness of the CGAM, we replaced the ELAN[43] module in the YOLOv7 backbone with our CGAM module. Compared with YOLOv7, using CGAM increased the mAP0.5 by 0.7%. This is because our CGAM can extract local information, coordinate information, and global information simultaneously, making the extracted feature map richer in contextual information, and thereby improving the ability of the backbone network to extract features.;
- **MF-FPN**: To demonstrate the effectiveness of MF-FPN, we replaced the neck part of YOLOv7 with the proposed MF-FPN. Compared with YOLOv7, the improved model with MF-FPN increased mAP0.5 by 1%, and the parameters of the model also decreased by 2.2M. This indicates that our proposed MF-FPN can better fuse multi-scale features with richer contextual information under fewer parameters, thereby obtaining feature maps with richer contextual information.;
- **Focal-EIOU Loss**: To reflect the effectiveness of Focal-EIOU Loss, we replaced the CIOU loss in YOLOv7 with Focal-EIOU Loss. Compared with CIOU Loss, Focal-EIOU Loss can more accurately regress the bounding box and allow high-quality anchor boxes to make more contributions during training, thereby improving detection performance. Compared with YOLOv7, the model's mAP0.5 increased by 0.5%.;
- **Proposed Method**: When CGAM, MF-FPN, and Focal-EIOU loss were all incorporated into YOLOv7, our model was obtained. Compared with YOLOv7, the precision increased by 0.5%, the recall increased by 2.1%, the mAP0.5 increased by 1.9%, and the parameter size of our model was reduced by 0.7M compared to the baseline. The results show that our improvement methods are very effective, and each improvement can enhance the performance of the model.;

Table 5. Ablation experiments.

Method	CGAM	MF-FPN	Focal-EIOU	Precision	Recall	mAP0.5	Params(M)	GFLOPs
YOLOv7				58.3	49.2	49.0	36.5	103.3
B1	✓			55.8	51.2	49.7	38.0	104.7
B2		✓		**61.1**	48.2	50.0	34.3	104.0
B3			✓	59.0	50.3	49.5	36.5	103.3
B4	✓	✓	✓	58.8	**51.3**	**50.9**	35.8	105.3

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

We also plotted the change of mAP0.5 and mAP with epochs as shown in Figure 10. It is evident from the figure that compared with YOLOv7, each of our improvement points can increase mAP0.5 and mAP with the increase of epochs. Especially, the model we proposed by integrating all the improvement points has a significant improvement compared to YOLOv7.



(a1) (b1)

Figure 10. Comparison of mAP0.5 and mAP for different improvement points.

In addition, we also listed the changes in the convolutional parameter sizes of each ELAN-H module in the neck part of the model. As shown in Table 6, ELAN-H_3 and ELAN-H_4 are two additional modules that we added to our model.

Table 6. Change of parameters in the neck part of the model.

Module	Baseline	Ours
ELAN-H_1	1.26M	0.1M
ELAN-H_2	0.32M	0.07M
ELAN-H_3	\	0.07M
ELAN-H_4	\	0.21M
ELAN-H_5	1.26M	0.85M
ELAN-H_6	5.05M	3.4M

To better demonstrate the effectiveness of CGMDet, we use Grad-CAM[44] to visualize the model's execution results in the form of heatmaps. As shown in Figure 11, the first row of the image shows that compared with YOLOv7, our model reduces the focus on similar objects around small targets and can more accurately detect small targets. The second row shows that our model alleviates the interference of background factors. The third row shows the heat map results generated by our model in low-light nighttime scenes, where we can observe that even in low-light conditions, our model can accurately focus on the target while reducing the attention to the background.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 29 May 2023 doi:10.20944/preprints202305.1967.v1

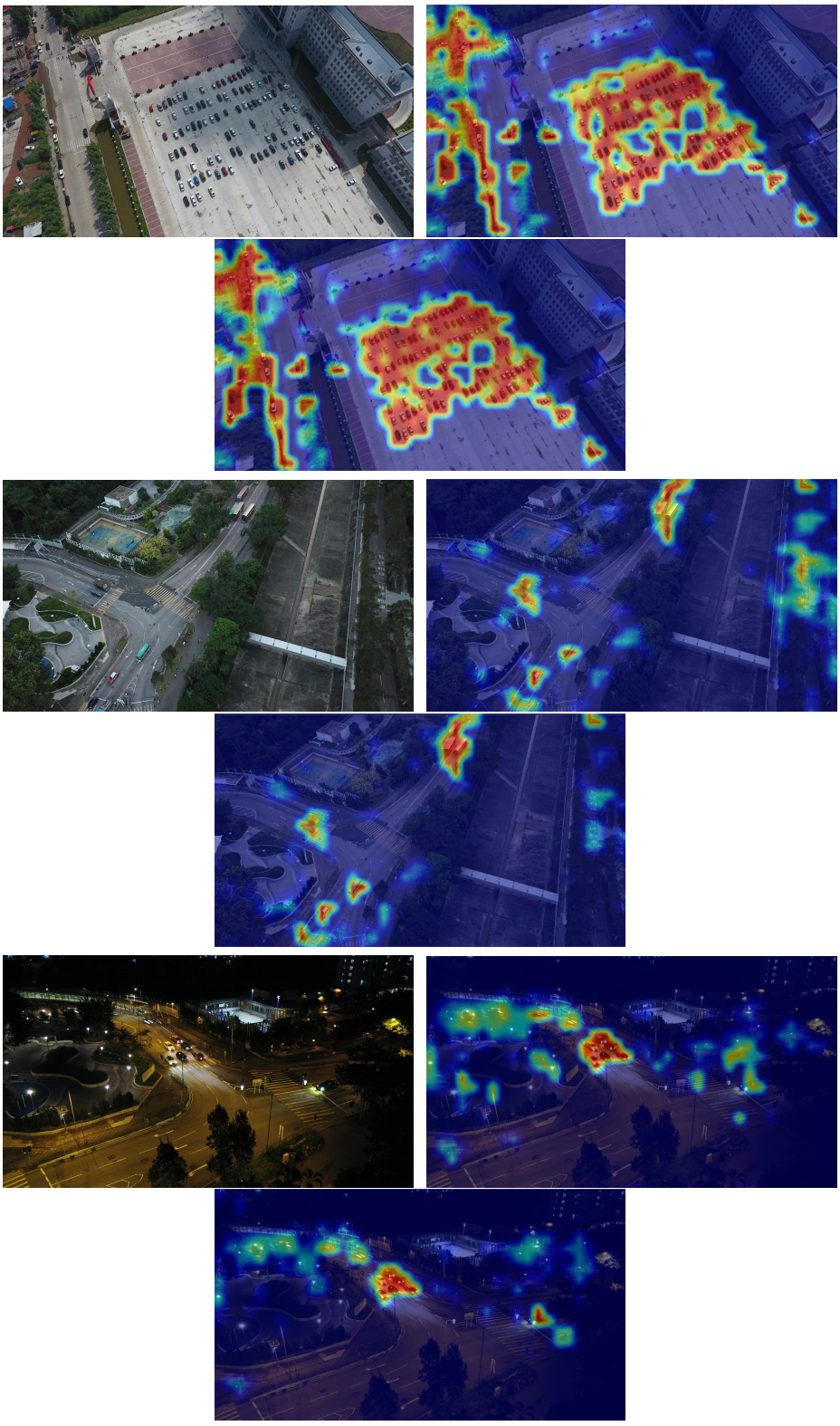


Figure 11. Example of heatmap visualization. The first column is the original image, the second column is the visualization result of YOLOv7, and the third column is the visualization result of CGMDet.

5. Conclusions

In this study, a multi-scale object detector based on coordinate and global information aggregation for UAV aerial images is proposed. This detector can focus more on the features of the objects and better detect multi-scale objects. We designed a Coordinate and Global Information Aggregation

 doi:10.20944/preprints202305.1967.v1

Module that integrates local information, coordinate information, and global information to obtain more robust features, which effectively alleviates the interference of background factors in the feature extraction process. The proposed Multi-Feature Fusion Pyramid Network greatly enhances the model's performance to detect different scale targets and reduce the numbers of the convolution channel to decrease the model's parameters. Finally, by modifying the bounding box loss, high-quality anchor boxes can contribute more gradients to the model, thereby helping to regress the bounding boxes more accurately and improving the convergence speed of the model. Our experiments show that compared to the baseline detectors, CGMDet improves mAP0.5 by 1.9% on the VisDrone dataset and 3.0% on the UAVDT dataset. In future work, we will focus on the detection of dense objects and the development of lightweight models.

Author Contributions: Conceptualization, Liming Zhou; methodology, Zhehao Liu; software, Zhehao Liu; validation, Liming Zhou, Zhehao Liu, Hang Zhao and Yan-e Hou; formal analysis, Yang Liu; investigation, Lanxue Dang; resources, Xianyu Zuo; data curation, Yan-e Hou; writing—original draft preparation, Liming Zhou and Zhehao Liu; writing—review and editing, Liming Zhou and Yang Liu; visualization, Hang Zhao; supervision, Hang Zhao; project administration, Zhehao Liu; funding acquisition, Xianyu Zuo. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Basic Research Program of China (Grant no. 2019YFE0126600); the Major Project of Science and Technology of Henan Province (Grant no. 201400210300); the Key Scientific and Technological Project of Henan Province (Grant no. 212102210496); the Key Research and Promotion Projects of Henan Province (Grant nos. 212102210393 and 202102110121); Kaifeng Science and Technology Development Plan (Grant no. 2002001); National Natural Science Foundation of China (Grant no. 62176087); and Shenzhen Science and Technology Innovation Commission (SZSTI), Shenzhen Virtual University Park (SZVUP) Special Fund Project (Grant no. 2021Szvup032).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Acknowledgments: We sincerely thank the anonymous reviewers for the critical comments and suggestions for improving the manuscript.

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

Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
HOG	Histogram of Oriented Gradients
SIFT	Scale Invariant Feature Transform
SSD	Single Shot MultiBox Detector
YOLO	You Only Look Once
FCOS	Fully Convolutional One-Stage Object Detection
CGAM	Coordinate and Global Information Module
MF-FPN	Multi-Feature Fusion Pyramid Network
SE	Squeeze-and-excitation
ECA	Efficient Channel Attention
ESE	Effective Squeeze-and-excitation
CA	Coordinate Attention
CBAM	Convolutional Block Attention Module
FPN	Feature Pyramid Network
PANet	Path Aggregation Network
MLFPN	Multi-Level Feature Pyramid Network
TUM	Thinned U-shape Module
FFM	Feature Fusion Module
BiFPN	Bidirectional Feature Pyramid Network
IOU	Intersection over Union

 doi:10.20944/preprints202305.1967.v1

P-R curve Precision-Recall curve
GFLOPs Giga Floating-point Operations Per Second

References

1. Wei, W.; Chao, F. and Ting, L. Multiperiod unmanned aerial vehicles path planning with dynamic emergency priorities for geohazards monitoring. *IEEE Transactions on Industrial Informatics*, **2022**, *18(12)*, 8851–8859.
2. Villarreal, C.A.; Garzón, C.G.; Mora, J.P.; Rojas, J.D. and Ríos, C.A. Workflow for capturing information and characterizing difficult-to-access geological outcrops using unmanned aerial vehicle-based digital photogrammetric data. *Journal of Industrial Information Integration*, **2022**, *26*, p.100292.
3. Hailong, H.; Savkin, A.V. and Chao, H. Decentralized autonomous navigation of a uav network for road traffic monitoring. *IEEE Transactions on Aerospace and Electronic Systems*, **2021**, *57(4)*, 2558–2564.
4. Do-Duy, T.; Nguyen, L. D.; Duong, T. Q.; Khosravirad, S. R. and Claussen, H. Joint optimisation of real-time deployment and resource allocation for uav-aided disaster emergency communications. *IEEE Journal on Selected Areas in Communications*,**2021**, *39(11)*, 3411–3424.
5. Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, **2005**, *1*, 886–893.
6. Lowe, D. G. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, **1999**, *2*, 1150–1157.
7. Girshick, R.; Donahue, J.; Darrell, T. and Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, **2014**, 580–587.
8. Girshick, R Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, **2015**, 1440–1448.
9. Ren, S.; He, K.; Girshick, R. and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, bf 2017, em 39(6), 1137–1149.
10. He, K.; Zhang, X.; Ren, S. and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2015**, *37(9)*, 1907–1916.
11. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C. Y. and Berg, A. C. Ssd: Single shot multibox detector. *In Proceedings of the European Conference on Computer Vision*, **2016**, 21–37.
12. Lin, T. Y.; Goyal, P.; Girshick, R.; He, K. and Dollár, P. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, **2017**, 2999–3007.
13. Redmon, J.; Divvala, S.; Girshick, R. and Farhadi, A. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2016**, 779–788.
14. Redmon, J. and Farhadi, A. Yolo9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2017**, 6517–6525.
15. Redmon, J. and Farhadi, A. Yolov3: An incremental improvement. *arXiv*, **2018**, arXiv:1804.02767.
16. Bochkovskiy, A.; Wang, C. Y. and Liao, H. Y. M. Yolov4: Optimal speed and accuracy of object detection. *arXiv*, **2020**, arXiv:2004.10934.
17. Ultralytics. Yolov5. *Accessed: Jun. 18, 2022. [Online]. Available: https://github.com/ultralytics/yolov5*.
18. Wang, C. Y.; Bochkovskiy, A. and Liao, H. Y. M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv*, **2022**, arXiv:2207.02696.
19. Tian, Z.; Shen, C.; Chen, H. and He, T. Fcos: Fully convolutional one-stage object detection. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, **2019**, 9626–9635.
20. Zhou, X.; Wang, D. and Krähenbühl, P. Objects as points. *arXiv*, **2019** arXiv:1904.07850.
21. Hu, J.; Shen, L. and Sun, G. Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, **2018**, 7132–7141.
22. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W. and Hu, Q. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **2020**, 11531–11539.
23. Lee, Y. and Park, J. CenterMask: Real-Time Anchor-Free Instance Segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **2020**, 13903–13912.
24. Hou, Q.; Zhou, D. and Feng, J. Coordinate Attention for Efficient Mobile Network Design. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **2021**, 13708–13717.

25. Woo, S.; Park, J.; Lee, J. Y. and Kweon, I. S. Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. *Proceedings of the European Conference on Computer Vision (ECCV)*, **2018**, 3–19.
26. Lin, T. Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B. and Belongie, S. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2017**, 936–944.
27. Liu, S.; Qi, L.; Qin, H.; Shi, J. and Jia, J. Path aggregation network for instance segmentation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, **2018**, 8759–8768.
28. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L. and Ling, H. M2Det: A Single-Shot Object Detector Based on Multi-Level Feature Pyramid Network. *The Thirty-Third AAAI Conference on Artificial Intelligence,AAAI*, **2019**, *33*, 9259–9266.
29. Tan, M.; Pang, R. and Le, Q. V. Efficientdet: Scalable and efficient object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **2020**, 10778–10787.
30. Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International conference on machine learning* , **2015**, *37*, 448–456.
31. Elfwing, S; Uchibe, E and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, **2018**, *107*, 3–11.
32. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R. and Ren, D. Distance-iou loss: Faster and better learning for bounding box regression. *AAAI Conference on Artificial Intelligence*, **2020**, *34(07)*, 12993–13000.
33. Zhang, Y. F.; Ren, W.; Zhang, Z.; Jia, Z.; Wang, L. and Tan, T. Focal and efficient iou loss for accurate bounding box regression. *Neurocomputing*, **2022**, *506*, 146–157.
34. Sun, Z.; Leng, X.; Lei, Y.; Xiong, B.; Ji, K. and Kuang, G. BiFA-YOLO: A Novel YOLO-Based Method for Arbitrary-Oriented Ship Detection in High-Resolution SAR Images. *Remote Sensing*, **2021**, *13(21)*, 4209.
35. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q. and Ling, H. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2022**, *44*, 7380-7399.
36. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W. Huang Q. and Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. *Proceedings of the European conference on computer vision (ECCV)*, **2018**, 370-386.
37. Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P. and Zitnick, C. L. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, **2014**, 740-755.
38. Cai, Z. and Vasconcelos, N. Cascade R-CNN: Delving Into High Quality Object Detection. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, **2018**, 6154–6162.
39. Ge, Z.; Liu, S.; Wang, F.; Li, Z. and Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv*, **2021**, arXiv:2107.08430.
40. Lin, H.; Zhou, J.; Gan, Y.; Vong, C. M. and Liu, Q. Novel up-scale feature aggregation for object detection in aerial images. *Neurocomputing*, **2020**, *411*, 364–374.
41. Yang, C.; Huang, Z. and Wang, N. Querydet: Cascaded sparse query for accelerating high-resolution small object detection. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **2022**, 13668–13677.
42. Liu, C.; Yang, D.; Tang, L.; Zhou, X. and Deng, Y. A Lightweight Object Detector Based on Spatial-Coordinate Self-Attention for UAV Aerial Images. *Remote Sensing*, **2023**, *15(1)*, 83.
43. Wang, C. Y.; Liao, H. Y. M. and Yeh, I. H. Designing network design strategies through gradient path analysis. *arXiv*, **2022**, arXiv:2211.04800.
44. Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D. and Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, **2017**, 618–626.