mathematics

MDPI

Article

Eagle-YOLO: An Eagle-Inspired YOLO for Object Detection in Unmanned Aerial Vehicles Scenarios

Lyuchao Liao [1], Linsen Luo [1,*], Jinya Su [2], Zhu Xiao [3], Fumin Zou [4] and Yuyuan Lin [1]

[1] Fujian Provincial Universities Engineering Research Center for Intelligent Driving Technology, Fujian University of Technology, Fuzhou 350118, China
[2] Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK
[3] The College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[4] Fujian Provincial Key Laboratory of Automotive Electronics and Electric Drive, Fujian University of Technology, Fuzhou 350118, China
* Correspondence: lsen_luo@smail.fjut.edu.cn

Abstract: Object detection in images taken by unmanned aerial vehicles (UAVs) is drawing ever-increasing research interests. Due to the flexibility of UAVs, their shooting altitude often changes rapidly, which results in drastic changes in the scale size of the identified objects. Meanwhile, there are often many small objects obscured from each other in high-altitude photography, and the background of their captured images is also complex and variable. These problems lead to a colossal challenge with object detection in UAV aerial photography images. Inspired by the characteristics of eagles, we propose an Eagle-YOLO detection model to address the above issues. First, according to the structural characteristics of eagle eyes, we integrate the Large Kernel Attention Module (LKAM) to enable the model to find object areas that need to be focused on. Then, in response to the eagle's characteristic of experiencing dramatic changes in its field of view when swooping down to hunt at high altitudes, we introduce a large-sized feature map with rich information on small objects into the feature fusion network. The feature fusion network adopts a more reasonable weighted Bi-directional Feature Pyramid Network (Bi-FPN). Finally, inspired by the sharp features of eagle eyes, we propose an IoU loss named Eagle-IoU loss. Extensive experiments are performed on the VisDrone2021-DET dataset to compare it with the baseline model YOLOv5x. The experiments showed that Eagle-YOLO outperformed YOLOv5x by 2.86% and 4.23% in terms of the mAP and AP50, respectively, which demonstrates the effectiveness of Eagle-YOLO for object detection in UAV aerial image scenes.

Keywords: object detection; unmanned aerial vehicle; attentional mechanisms; Eagle-YOLO

MSC: 68T07

check for updates

Citation: Liao, L.; Luo, L.; Su, J.; Xiao, Z.; Zou, F.; Lin, Y. Eagle-YOLO: An Eagle-Inspired YOLO for Object Detection in Unmanned Aerial Vehicles Scenarios. *Mathematics* **2023**, *11*, 2093. https://doi.org/10.3390/math11092093

Academic Editors: Liang Zou, Liang Zhao and Yonghui Xu

Received: 17 March 2023
Revised: 21 April 2023
Accepted: 24 April 2023
Published: 28 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Object detection is a significant branch of image processing and computer vision and is the basis for many advanced computer vision applications. Therefore, object detection has attracted increasing attention in both academia and industry [1–3] in recent years. In particular, object detection in aerial photography scenarios with UAVs has led to various important applications, such as biological detection, disaster rescue, security maintenance, and power inspection. However, the high flexibility of UAVs makes the images vary drastically in scale (e.g., Figure 1a,b), and the high altitude shots lead to the presence of a large number of small objects that obscure each other (e.g., Figure 1c). At the same time, the backgrounds of the captured images are also complex and variable (Figure 1d). These issues make the object detection of UAV aerial images a vital challenge in practice.

In recent years, deep-neural-network-based object detection algorithms have improved by leaps and bounds compared to traditional object detection algorithms. The current mainstream object detection algorithms are based on deep learning models, broadly divided into anchor-based and anchor-free detectors. There are two main types of anchor-based detectors: the highly accurate two-stage detector and the highly efficient one-stage detector. The two-stage detector is used to generate regional proposals, and then the next step is to locate and classify objects in more detail based on the candidate regions. The one-stage detector algorithm is a "one-step" approach, which does not require the model to generate region proposals but rather to obtain the position and classification of the object in the input image. The idea of the anchor-free detectors algorithm is to view the object detection problem as a critical point detection and classification problem. Without the introduction of the anchor frame mechanism, the time-consuming design of the anchor frame size, scale, and other hyperparameters can be significantly reduced.

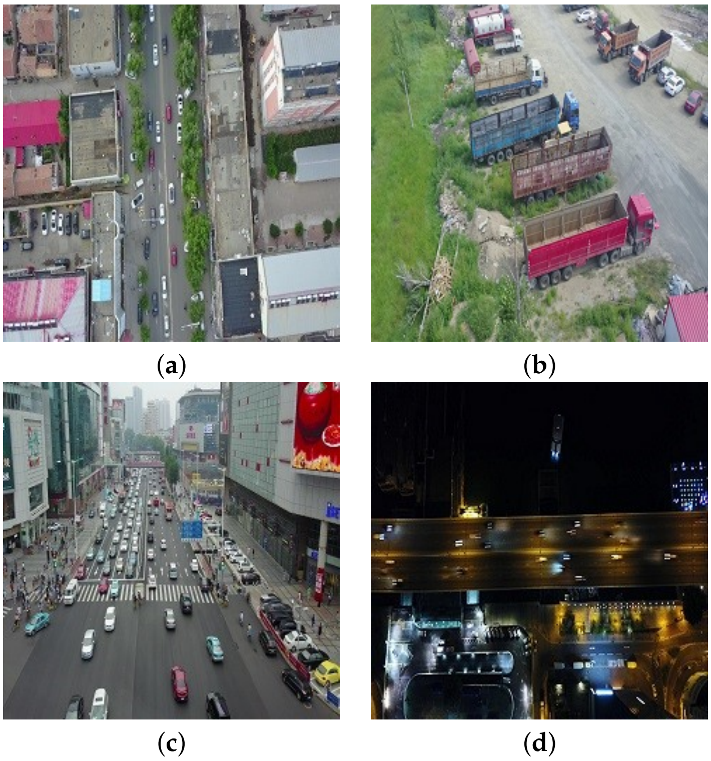
(**a**)
(**b**)
(**c**)
(**d**)

Figure 1. Schematic diagram of the characteristics of UAV aerial imagery. (**a**) Overhead image. (**b**) Low altitude image. (**c**) Object occlusion scene image. (**d**) Complex background image at night.

Abound object detectors were proposed to detect natural objects in some datasets, such as MS COCO [4], PASCAL VOC [5], and ImageNet [6]. However, the object detectors that perform well under natural images do not perform well when applied to UAV aerial images. This issue is caused by the characteristics of aerial UAV images that differ significantly from those of naturally collected images. Therefore, this work is dedicated to improving object detection performance in UAV aerial photography scenes.

The aerial photography characteristics of drones are remarkably similar to the hunting behavior of eagles. In particular, eagles often swoop down from high altitudes to hunt, which is similar to the drastic changes in the scale of the images captured by UAVs, as shown in Figure 2. In fact, the unique structure of the eagle's eye enables it to see not only far but also a wide range. This work combines the behavioral features of eagles to optimize the YOLO [3,7–10] and proposes an aerial image object detection method named Eagle-YOLO, which supports multiscale agile perception fusion to enable the accurate detection of UAV aerial images.

Figure 2. Schematic diagram of eagle hunting behavior.

The basic flow of the Eagle-YOLO model is shown in Figure 3. First, the input image goes through a data augmentation module to increase the robustness and diversity of the data and then through the backbone feature extraction network to extract the features. The backbone network of Eagle-YOLO is consistent with that of YOLOv5 as CSPDarknet53, so Eagle-YOLO could employ the pretraining weights obtained from YOLOv5's training on the ImageNet [6] dataset to speed up the model convergence. Due to the presence of a large number of small mutually occluded objects in the UAV aerial images and the low resolution and lack of semantic information of the small objects, the Eagle-YOLO model introduces a low-level feature map with rich details on small objects into the feature fusion network. The model also introduces an attention mechanism to enable the model to focus on detecting key object areas and capture semantic information. Eagle-YOLO uses Bi-FPN [11] as the feature fusion network architecture, and the extracted feature information at each level is weighted and fused to the final output for prediction.

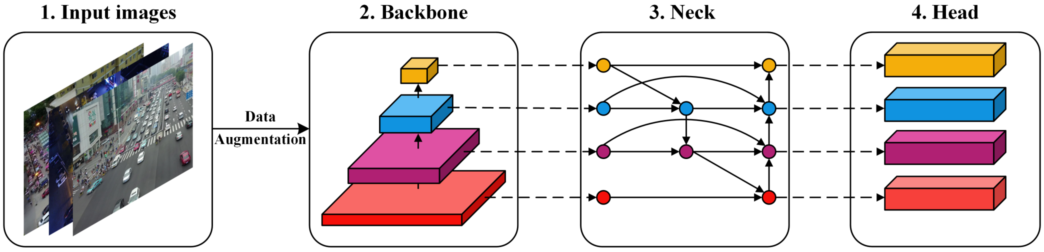

Figure 3. Overview of the basic procedure of the Eagle-YOLO model. Compared to YOLOv5, this model replaces the feature fusion network with a Bi-FPN and adds a large-size detection head.

The main contributions of this paper are summarized below:

- Inspired by the structure of an eagle's eye, Eagle-YOLO is proposed to integrate the LKAM and is enabled to focus on the object to be detected and capture the object's semantic information.
- Like the dramatic visual changes the eagles experience during swooping hunting, a Bi-FPN feature fusion network is introduced to capture the feature map containing more small object information and contextual semantic information to improve the detection performance and accuracy.
- Inspired by the sharp eyes of eagles, an Eagle-IoU loss is proposed to improve the original IoU loss of YOLOv5, which improves the detection accuracy and speeds up the convergence.

Finally, extensive experiments are conducted on the VisDrone2021-DET dataset, and the state-of-the-art YOLOv5x is employed as the benchmark for comparative analysis. The experimental results showed that Eagle-YOLO improved by 2.86% and 4.23% over

YOLOv5x in terms of the mAP and AP50, respectively. Additionally, Eagle-YOLO hds a better detection capability for complex backgrounds and unbalanced classes, which provides the effectiveness of object detection in UAV aerial image scenes.

2. Related Work

2.1. Usual Object Detection

Presently, object detection algorithms based on CNNs are divided into two main categories: anchor-based detectors and anchor-free detectors. Anchor-based detectors are mainly divided into two-stage detectors and one-stage detectors. Faster R-CNN [12] and Mask R-CNN [13] are the commonly used two-stage detectors. Mask R-CNN extends Faster R-CNN by adding a fully connected segmentation subnetwork in parallel for pixel-level object instance segmentation. A Swin Transformer [14,15] is one of the more recently popular one-stage detectors. It is a novel Transformer, called the hierarchical Transformer, which uses the idea of shifted windows to restrict the computation of self-attentiveness to nonoverlapping local windows while also allowing cross-window connections for efficiency. This allows information to be interacted between neighboring patches while reducing the computational complexity of the traditional Transformer. Recently, YOLOX has shown an outstanding processing speed and detection accuracy for the natural scene image dataset [9]. However, UAV aerial images are far from natural scene images, and these UAV images generally show the features of complex backgrounds and lots of small objects. Hence, directly using YOLOX models to detect objects in UAV aerial images will result in missed and false detections [16]. Anchor-free detectors mainly detect objects via classification and key point detection. RepPoint [17] is an anchor-free detector, which novelly proposes using point sets to represent the target's position. Unlike other one-stage detectors that use one-time regression and former classification to obtain the final object location and class, RepPoint provides finer-grained classification and easier localization. RepPoint uses two regressions and one classification, in which the classification and the last regression use irregular and deformable convolution. However, their detection performance is not suitable for drone aerial photography scenarios.

2.2. UAV Aerial Photography Object Detection

Object detection in UAV images is more challenging than general object detection. Firstly, the background of UAV aerial imagery is complex and varied, and data augmentation is one of the common tools used to increase the robustness and diversity of the data, which allows the model to learn more and have better generalization capabilities. Currently, the most commonly used methods of data augmentation in traditional vision are adding noise, random cropping, flipping horizontally, flipping up and down, scaling, panning, and enhancing the image brightness and contrast. Advanced data augmentation includes Cutmix [18], Transmix [19], Mosaic [7], and others. The next big tricky challenge for object detection in UAV images is the presence of a large number of small objects that obscure each other. Multiscale feature learning is a common approach [20,21], and Wang et al. [22] proposed a Cascade mask generation framework that can quickly detect small objects by combining multiscale input methods. However, memory consumption and inference time are generally increased dramatically, which makes the model computationally expensive. Furthermore, the deep network has a large sensory field and strong semantic information, which lacks texture information of the object and is not conducive to detecting small objects. Lin et al. [23] proposed a Feature Pyramid Network (FPN). This top-down architecture gives each feature map layer a stronger ability to capture semantic information by fusing deep semantic information with shallow texture information. However, the bottom-up fusion approach has a single direction and lacks information fusion in a double direction. Although many variants of FPNs have emerged in recent years [24–29], no specific detection algorithm is proposed to improve small objects and their structure detection.

3. Methodology

Inspired by the characteristics of eagles, a novel object detection model based on the YOLOv5, named Eagle-YOLO, is proposed for UAV aerial image detection. As shown in Figure 4, the model architecture of Eagle-YOLO consists of three modules, namely the Backbone, Neck, and Head.

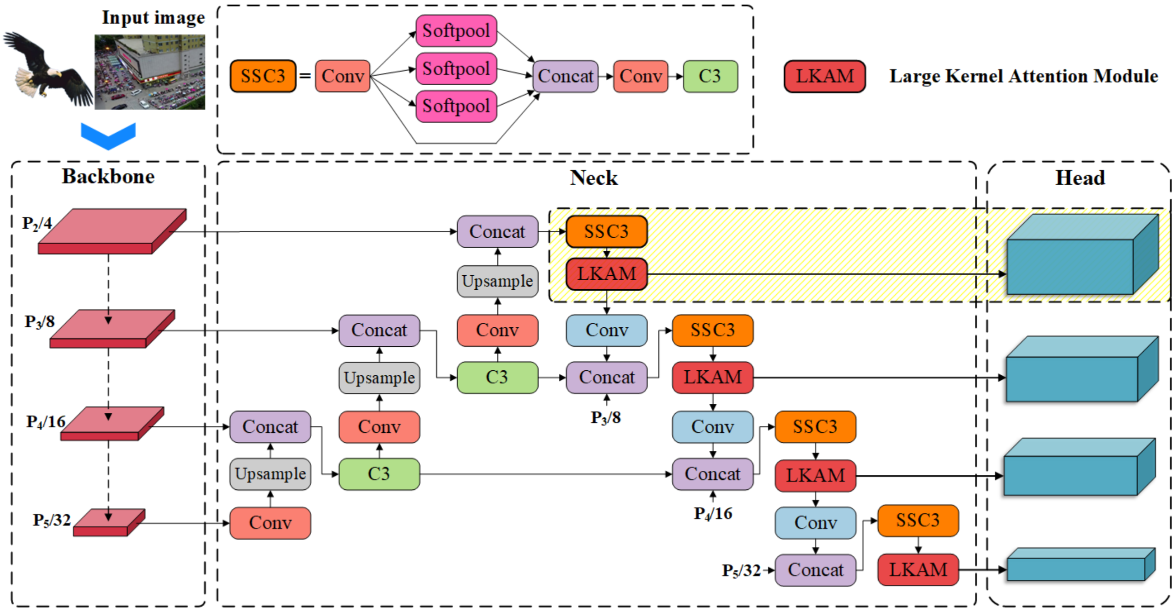

Figure 4. Diagram of the Eagle-YOLO network structure.

In the Eagle-YOLO network architecture, the first module is the Backbone for feature extraction at different levels, where $P_2/4, P_3/8, P_4/16, P_5/32$ represent images downsampled 4, 8, 16, and 32 times, respectively. The Backbone follows the original YOLOv5 CSPDarknet53 backbone network, and the training efficiency of the model is improved by using YOLOv5's initial pretraining weights. The second module is the Neck module for feature re-extraction and feature fusion, which is also a crucial part of the Eagle-YOLO.

The overall feature fusion architecture is based on Bi-FPN [11], which fuses features at different levels by weighting the values, which makes the feature fusion network more rational. The SSC3 submodule in the Neck module consists of SoftPool Spatial Pyramid Pooling (SSPP) and C3. The SSPP uses a SoftPool weighted by Softmax for pooling [30], which helps keep the features' expressiveness well. The attention mechanism LKAM is introduced to enable Eagle-YOLO to focus on essential object regions and capture semantic information for feature network fusion. The third module is the Head, in which a detection layer containing rich details on small objects is employed to improve the detection performance for small objects.

3.1. Attention Module

The visual attention module mimics the human visual mechanism by quickly previewing the global image and focusing more on the critical object area to obtain more detailed information. The basic form of the attention mechanism is shown in Equation (1). Due to the unique structure of eagle eyes [31], as shown in Figure 5a, eagle eyes possess a well-developed binocular vision and show acute visual capabilities. Therefore, we introduce the LKAM in this work to capture a larger field of perception and enhance the features [32]:

$$\text{Attention} = f(g(x), x) \tag{1}$$

where $g(x)$ is used to find the important regions, and $f(g(x), x)$ is used to make the network focus on the critical regions.

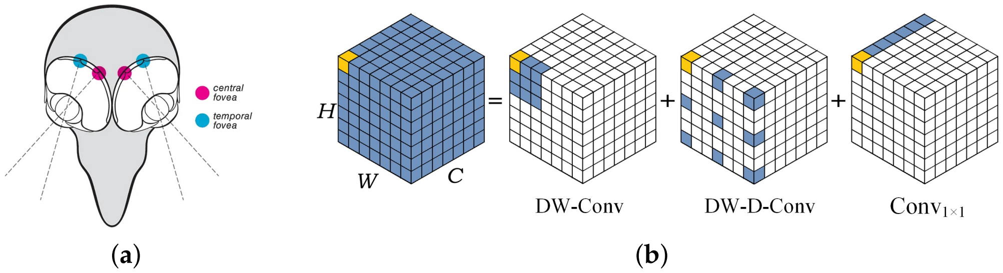

Figure 5. (**a**) Eagle eyes. (**b**) Decomposition diagram of large-kernel convolution.

The core idea of the LKAM is to generate an attention map through large-kernel convolution to obtain important object areas, the essential expression of which is Equations (2) and (3). However, large-kernel convolution usually leads to significant computational overhead and parameter drawbacks. Therefore, the LKAM decomposes the large-kernel convolution into a depth-wise convolution with local spatial properties(DW-Conv), a depth-wise dilation convolution with long-range spatial properties (DW-D-Conv), and a channel convolution (Conv$_{1\times1}$), as shown in Figure 5b. By the decomposition, the LKAM can generate attention maps with a small computational cost:

$$g(x) = \mathrm{LargeKernelConv}(x) \tag{2}$$

$$f(g(x), x) = g(x) \otimes x \tag{3}$$

where $x \in \mathbb{R}^{C\times H\times W}$ denotes the input feature map and $\otimes$ denotes the product of the elements.

As shown in Figure 6, the input data for the LKAM are feature tensors $\mathbf{x} \in \mathbb{R}^{C\times H\times W}$.

Firstly, the original feature map $\mathbf{x}$ is input to the DW-Conv submodule, which generates a three-dimensional spatial local attention feature denoted as $\mathbf{F}_\mathbf{l} \in \mathbb{R}^{C\times H\times W}$ from a convolution of kernel size of 5 × 5 and padding of 2. $\mathbf{F}_\mathbf{l}$ is then passed through the DW-D-Conv submodule, which generates a three-dimensional spatial long-range dependence attention feature denoted as $\mathbf{F}_\mathbf{lr} \in \mathbb{R}^{C\times H\times W}$ from a convolution of kernel size of 7 × 7, padding of 9, and dilation of 3. Then, $\mathbf{F}_\mathbf{lr}$ is fed into a Conv$_{1\times1}$ submodule consisting of a convolution of kernel size of 1 × 1 to learn the channel adaptability and obtain a three-dimensional attention map containing all of the above features. Finally, this submodule generates a spatial adaptability feature $\mathbf{F} \in \mathbb{R}^{C\times H\times W}$ and then multiplies it by the original input tensor $\mathbf{x}$ to obtain the refined features. The above derivation process is given in Equations (4)–(7):

$$\mathbf{F}_\mathbf{l} = \text{DW-Conv}_{5\times5}(\mathbf{x}) \tag{4}$$

$$\mathbf{F}_\mathbf{lr} = \text{DW-D-Conv}_{7\times7}(\mathbf{F}_\mathbf{l}) \tag{5}$$

$$\mathbf{F} = \mathrm{Conv}_{1\times1}(\mathbf{F}_\mathbf{lr}) \tag{6}$$

$$\mathrm{Attention} = \mathbf{F} \otimes \mathbf{x} \tag{7}$$

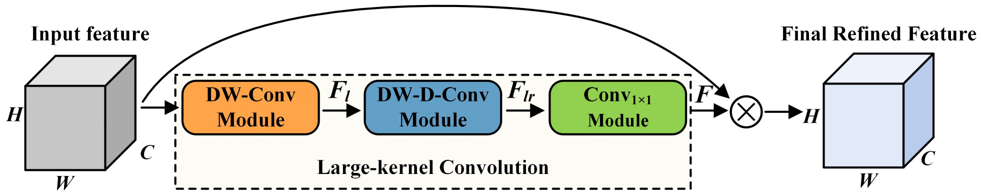

Figure 6. The overview of LKAM module.

3.2. Context Based Feature Fusion

In order to make the model have a better context feature fusion mechanism, Eagle-YOLO uses Bi-FPN [11] as the feature fusion network. In the original feature network fusion

PANet, features of different scales have the same weights for feature fusion. However, the differences in the information of different scale features are ignored during the feature fusion. Therefore, when it comes to feature fusion at different scales, Bi-FPN adds a learnable weight to enable the network to learn the contribution of features at different scales. At the same time, residual connection operations have been added to enhance the representation ability of features. So, the Bi-FPN employed in this work will make the feature fusion network more reasonable. Eagle-YOLO introduces a low-level feature map with rich spatial geometric detail information into the feature fusion network to improve the detection accuracy of small objects.

The schematic diagram of Bi-FPN is shown in Figure 7. In the schema, $\mathrm{P}_\mathrm{n}^\mathrm{in}(\mathrm{n} = 2,3,4,5)$ represents the feature map extracted by the Backbone with a scale of 2^x times downsampling; $w_n(\mathrm{n} = 1,2,3)$ denotes the learnable weight parameters; $\mathrm{P}_\mathrm{n}^\mathrm{mid}(\mathrm{n} = 3,4)$ represents the feature map under model feature re-extraction with a scale of 2^x times downsampling, and $\mathrm{P}_\mathrm{n}^\mathrm{out}(\mathrm{n} = 2,3,4,5)$ denotes the feature map downsampled at a scale of 2^x times. We take $\mathrm{P}_4^\mathrm{out}$ as a example. As shown in Equations (8) and (9), $\mathrm{P}_4^\mathrm{out}$ is a weighted fusion of P_4^in, $\mathrm{P}_4^\mathrm{mid}$, $\mathrm{P}_3^\mathrm{out}$. Then, $\mathrm{P}_3^\mathrm{out}$ is an input to the LKAM to capture the contextual semantic information and incorporate this information into the feature fusion network.

$$\mathrm{P}_4^\mathrm{out} = \mathrm{Conv}\left(\mathrm{w}_1^{'} \times \mathrm{P}_4^\mathrm{in} + \mathrm{w}_2^{'} \times \mathrm{P}_4^\mathrm{mid} + \mathrm{w}_3^{'} \times \mathrm{Resize}\left(\mathrm{P}_3^\mathrm{out}\right)\right) \tag{8}$$

$$w_n^{'} = \frac{w_n}{w_1 + w_2 + w_3 + \varepsilon} \tag{9}$$

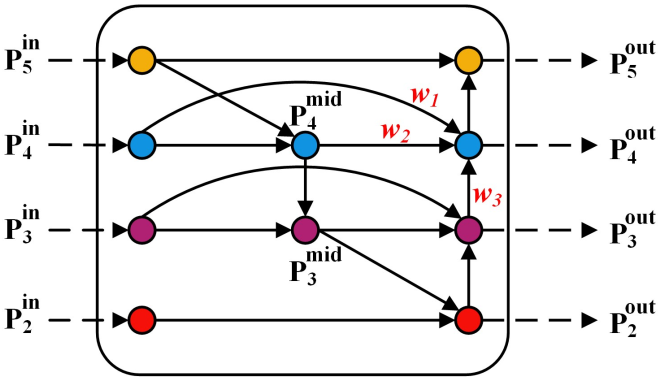
$\mathbf{P}_5^\mathbf{in}$
$\mathbf{P}_4^\mathbf{in}$
$\mathbf{P}_3^\mathbf{in}$
$\mathbf{P}_2^\mathbf{in}$
$\mathbf{P}_4^\mathbf{mid}$
$\mathbf{P}_3^\mathbf{mid}$
w_1
w_2
w_3
$\mathbf{P}_5^\mathbf{out}$
$\mathbf{P}_4^\mathbf{out}$
$\mathbf{P}_3^\mathbf{out}$
$\mathbf{P}_2^\mathbf{out}$

Figure 7. Schematic diagram of Bi-FPN.

3.3. Eagle-IoU Loss

Like an eagle's eyes, whose sharpness makes it possible to perceive small objects at high altitudes and in complex scenarios, the loss function in the Eagle-YOLO should be improved in terms of its sense ability. Therefore, we propose Eagle-IoU loss; this loss function focuses on the angle between the center point of the truth bounding box and the predicted bounding box. In addition, we combine *L1* loss and Alpha-IoU [33] to balance the contribution of each part. The essential IoU loss (L) is represented by Equations (10) and (11):

$$L = 1 - \mathrm{IoU} + R\left(B, B^{GT}\right) \tag{10}$$

$$\mathrm{IoU} = \frac{|B \cap B^{GT}|}{|B \cup B^{GT}|} \tag{11}$$

where B represents the predicted bounding box; B^{GT} denotes the ground truth bounding box, and IoU represents the ratio of the intersection and union of B and B^{GT}. $R\left(B, B^{GT}\right)$ is the penalty term in the essential Loss function, consisting of B and B^{GT}.

As a specific model, we propose a novel penalty term of Eagle-IoU loss in this work, denoted as R_{Eagle}. This penalty term integrates three parts: the Angle–Distance cost (R_a),

$L1$ cost (R_{L1}), and Shape cost (R_s). Firstly, we calculate the angle penalty strength (P_{Angle}) for the Angle–Distance cost (R_a), as shown in Equation (12):

$$P_{Angle} = 1 - 2 * \sin^2\left(\theta - \frac{\pi}{4}\right) \tag{12}$$

where θ is the angle between the center point of B and B^{GT}, as shown in Figure 8. If $0 \leq \theta \leq \pi/4$, then B and B^{GT} keep approaching the same horizontal line; otherwise, $\pi/4 < \theta \leq \pi/2$, then B and B^{GT} keep approaching the same vertical line. When $\theta = \pi/4$, the value P_{Angle} is the largest. Then, we calculate the value of the Angle–Distance cost (R_a), as shown in Equation (13):

$$R_a = \left(1 - e^{-\gamma\rho_w}\right) + \left(1 - e^{-\gamma\rho_h}\right) \tag{13}$$

where c_w and c_h are the x and y distances from the centers of B and B^{GT}, respectively. ρ_w and ρ_h are in Equations (14)–(16):

$$\rho_w = \left(\frac{c_w}{w}\right)^2 \tag{14}$$

$$\rho_h = \left(\frac{c_h}{h}\right)^2 \tag{15}$$

$$\gamma = 2 - P_{Angle} \tag{16}$$

where w and h are the length and width of the smallest rectangle enclosing B and B^{GT}, respectively; c_w and c_h are the horizontal distance and vertical distance between the center point of B and B^{GT}, respectively; and γ is the evaluation coefficient of the angle which is claculated to make sure that the value of P_{Angle} is the largest and the cost of R_a is the largest.

Secondly, the $L1$ cost(R_{L1}) is computed with Equation (17):

$$R_{L1} = \frac{|x - x^{gt}|}{w} + \frac{|y - y^{gt}|}{h} \tag{17}$$

where x , x^{gt} and y , y^{gt} are the x and y coordinates of the center points of B and B^{GT}, respectively.

Then, the Shape cost(R_s) is calculated with Equation (18):

$$R_s = \left(1 - e^{-\omega_w}\right)^3 + \left(1 - e^{-\omega_h}\right)^3 \tag{18}$$

where w is the length of B and h is the width of B. ω_w and ω_h are in Equations (19) and (20):

$$\omega_w = \frac{|w - w^{gt}|}{max(w, w^{gt})} \tag{19}$$

$$\omega_h = \frac{|h - h^{gt}|}{max(h, h^{gt})} \tag{20}$$

where w^{gt} is the length of B^{GT} and h^{gt} is the width of B^{GT}.

Finally, we establish the penalty term R_{Eagle} and $L_{Eagle-IoU}$, as shown in Equations (21) and (22):

$$R_{Eagle} = {R_a}^{\alpha_1} + {R_{L1}}^{\alpha_2} + {R_s}^{\alpha_3} \tag{21}$$

$$L_{Eagle-IoU} = 1 - \mathrm{IoU}^{\alpha} + R_{Eagle} \tag{22}$$

Extensive experiments conducted with the VisDrone2021-DET dataset showed that the model obtained the best performance when we set all the parameters $\alpha, \alpha_1, \alpha_2, \alpha_3$ as 3.

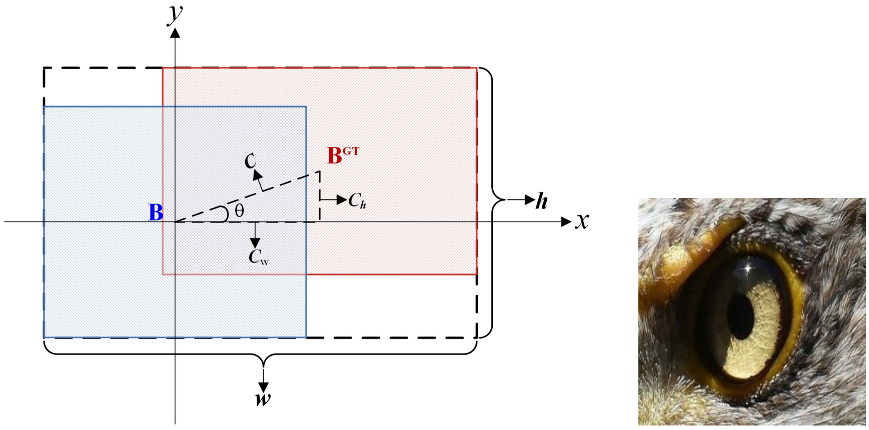

Figure 8. Calculation of the Eagle-IoU, the ground truth bounding box, and the predicted bounding box.

4. Experiments

The Eagle-YOLO was evaluated with the VisDrone2021-DET dataset [34]. The experimental environment was equipped with an NVIDIA RTX3090 (24G) GPU card, CUDA version 11.1, CUDNN version 8.2.1, python 3.8, and Pytorch 1.10.1.

4.1. Dataset

VisDrone2021-DET is an object-detection dataset with a drone perspective. In the VisDrone2021-DET dataset, there are 6471 images in the training set, 548 in the validation set, and 1580 in the test set. The dataset has ten categories: pedestrian, people, bicycle, car, van, truck, tricycle, bus, motor, and awning tricycle. The VisDrone2021-DET dataset includes several difficulties: (1) the size of the objects in images is highly variable; (2) some categories have a high degree of similarity (e.g., people and pedestrians); (3) most of the objects are too small (less than 32 pixels) to detect, and in addition, there are occlusions between the objects; and (4) data distribution is not uniform enough. For example, the numbers for cars and pedestrians are more than 65% of the total number of objects, while the number of other categories is small. The quantity and proportion are summarized in Table 1.

Table 1. The proportion and number of labels for each category.

Category	**Ped.** [1]	**Awn.** [2]	**People**	**Bicycle**	**Car**	**Bus**	**Van**	**Truck**	**Tricycle**	**Motor**
Quantity	79,335	3246	27,059	10,480	144,866	5926	24,956	12,875	4812	29,646
Proportion [%]	23.12	0.95	7.88	3.05	42.21	1.73	7.27	3.75	1.4	8.64

[1] Ped. is pedestrian; [2] Awn. is awning-tricycle.

4.2. Evaluation Metrics

In this work, we employed the mean Average Precision (mAP) and Average Precision 50 (AP50) to verify the model's performance. The mAP considers both precision and recall, which are calculated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{23}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{24}$$

where TP represents True Positive, FP represents False Positive, and FN represents False Negative. For object detection, when the Intersection Over Union (IoU) exceeds a threshold, the predicted result is considered a TP. Otherwise, the predicted result is regarded as an

FP. If the ground truth bounding box does not have a matching box, it is viewed as a FN. The Average Precision (AP) is the area of the Precision–Recall curve plotted against the coordinate axis circumference, which is calculated as shown in Equation (25). The AP50 represents the AP with a threshold value of 0.5. The mAP represents the average value of the AP under the IoU threshold in the range [0.5, 0.95], with a step size of 0.05, as shown in Equation (26):

$$AP = \int_0^1 P(R)dR \tag{25}$$

where P represents Precision, R represents Recall, and $P(R)$ represents the relationship function between P and R:

$$mAP = \frac{AP_{50} + AP_{55} + \ldots + AP_{90} + AP_{95}}{10} \tag{26}$$

4.3. Implementation Details

Experiment prep. YOLOv5 is available in four different model sizes, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. These models' mAP values on the COCO dataset increase sequentially, and the size of these models increases accordingly. In this work, we used the mAP and AP50 to evaluate the performance of these four models on the VisDrone2021-DET dataset, and the results are shown in Figure 9. Based on the experimental results, it is clear that YOLOv5x outperformed the other versions in both the mAP and AP50 metrics at all sizes. To achieve a better performance, in this work, we used the pretrained weights and backbone network of YOLOv5x, which not only has a better detection accuracy but also significantly reduces the training time of the model.

Training phase. We trained the model with a fixed image size of 1600 × 1600 and set two as the batch size. We used the VisDrone2021-DET-train dataset for 300 epochs of training, and the first two stages were used for warmup. We used an adam optimizer for training, with 3×10^{-4} as the initial learning rate, as shown in Table 2.

Test phase. We tested the model on the VisDrone2021-DET-test dataset with the same size of images as the training phase.

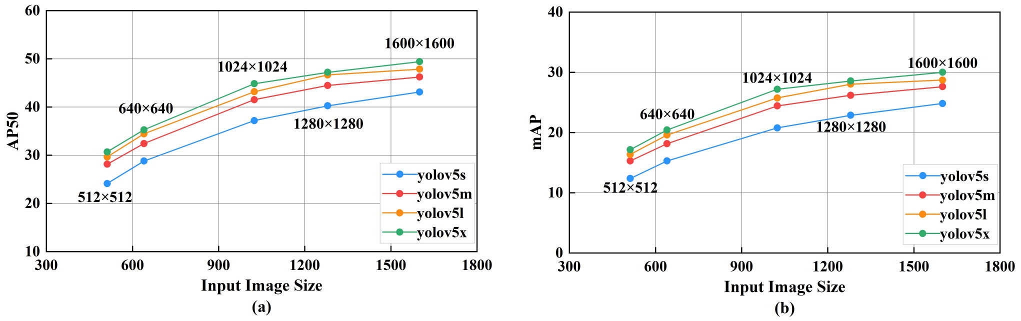

Figure 9. Comparison of performance metrics on VisDrone2021-DET dataset for each YOLOv5 model under different input sizes. (**a**) Evaluation metric is AP50. (**b**) Evaluation metric is mAP.

Table 2. Training parameters.

Parameter	Value
Image size	1600 × 1600
Batch size	2
Epochs	300
Warmup	2
Optimizer	adam
Initial learning rate	3×10^{-4}

5. Experimental Results

In this section, we evaluate the Eagle-YOLO model on VisDrone2021-DET-test dataset. In addition, we compare the performance of Eagle-YOLO with the baseline model YOLOv5x and other object detection algorithms. Table 3 reports the mAP of 10 categories in the VisDrone2021 dataset and the overall mAP and AP50. The experimental results showed that our model was far superior to the detection algorithm submitted by the VisDrone team, both in the overall detection performance and in the detection performance of a single category. Compared with CornerNet, which had the best overall performance among the detection algorithms submitted by the VisDrone team, Eagle-YOLO improved the overall AP50 and mAP by 15.5% and 19.52%, respectively. Regarding the bus category where the improvement was most significant, the mAP increased by 30.97%. Compared to the baseline model YOLOv5x, Eagle-YOLO improved the overall mAP and AP50 by 2.86% and 4.23%, respectively. Then, the three categories with the most significant improvement were truck, van, and bus, which improved by 4.64%, 4.13%, and 3.56%. All other categories of the mAP were also higher than the baseline model. After the above analysis, the Eagle-YOLO model outperformed the baseline model YOLOv5x overall.

Table 3. Comparison of the individual methods in the VisDrone2021-DET test.

Method	mAP	AP50	Ped.	People	Bicycle	Car	Van	Truck	Tricycle	Awn.	Bus	Motor
RetinaNet * [35]	11.81	21.37	9.91	2.92	1.32	28.99	17.82	11.35	10.93	8.02	22.21	7.03
RefineDet * [36]	14.9	28.76	14.9	3.67	2.02	30.14	16.33	18.13	9.03	10.25	21.93	8.38
Cascade-RCNN * [37]	16.09	31.91	16.28	6.16	4.18	37.29	20.38	17.11	14.48	12.37	24.31	14.85
FPN * [23]	16.51	32.2	15.69	5.02	4.93	38.47	20.82	18.82	15.03	10.84	26.72	12.83
Light-RCNN * [38]	16.53	32.78	17.02	4.83	5.73	32.29	22.12	18.39	16.63	11.91	29.02	11.93
CornerNet * [39]	17.41	34.12	20.43	6.55	4.56	40.94	20.23	20.54	14.03	9.25	24.39	12.1
YOLOv4 [7]	29.91	49.03	23.02	13.08	12.23	58.06	38.23	40.12	21.02	19.38	50.93	23.01
TPH-YOLOv5 [40]	30.94	50.69	23.80	13.43	12.74	59.06	40.38	42.77	19.31	19.87	53.33	24.71
YOLOv5x	30.05	49.41	23.1	13.1	12.4	58.2	38.3	40.3	21.3	19.7	51.8	23
Eagle-YOLO	**32.91**	**53.64**	**25.19**	**15.08**	**14.0**	**60.24**	**42.43**	**44.94**	**24.03**	**21.4**	**55.36**	**26.41**

* indicates the detection algorithm submitted by the VisDrone team. The best result is presented in bold.

6. Discussion

In this section, we firstly analyze the experimental results of Eagle-YOLO versus the baseline model YOLOv5x in Table 2; then, we perform ablation experiments on Eagle-YOLO by using the VisDrone2021-DET-test dataset, and finally, we visualize the model predictions.

It is worth noting that although truck and bus accounted for only 3.75% and 1.75% of the dataset, the model in this paper still had a good detection effect regarding truck and bus, with a mAP of 44.94% and 55.36%, respectively. Compared with the baseline model, the mAP of truck and bus improved by 4.64% and 3.56%, respectively. This indicated that the Eagle-YOLO model had a good feature learning ability and robustness for unbalanced data. For the tricycle and awning-tricycle categories, the mAP improvement was the lowest at only 1.6% and 1.7%, respectively. The performance of Eagle-YOLO in terms of detecting people and cars was not good enough. Maybe the reasons were the small size of the pixels and the mutual occlusion of the objects in these two categories.

Then, we validated the effect of each component of the Eagle-YOLO model by using the VisDrone2021-DET-test dataset. As shown in Table 4, the ablation study for each component was conducted by using the VisDrone2021-DET-test dataset. We employed YOLOv5x as the baseline model with a fixed input size of 1024 × 1024 for the images, and P2 represents images downsampling four times. After we added the P2 detection layer to YOLOv5x, the total number of layers increased from 444 to 525, and GFLOPs increased from 204.2 to 237.7. At the same time, the model's detection performance for small objects improved significantly. As shown in Table 3, the mAP and AP50 improved by 1.53% and 3.15%, respectively.

Table 4. Ablation studies in the VisDrone2021-DET test.

P2	Bi-FPN	LKAM	Eagle-IoU Loss	mAP (%)	AP50 (%)
-	-	-	-	27.12	44.56
✓				28.65 (↑1.53)	47.71 (↑3.15)
	✓			27.27 (↑0.15)	45.64 (↑1.08)
		✓		27.50 (↑0.38)	46.14 (↑1.58)
			✓	27.84 (↑0.72)	46.33 (↑1.77)
✓	✓			28.93 (↑1.81)	48.12 (↑3.56)

To evaluate the effectiveness of Bi-FPN in feature fusion, we replaced FPN with Bi-FPN in the model; then, the mAP and AP50 improved by 0.15% and 1.08%, respectively. Although the improvement was not significant, the improvement was significant when we combined the Bi-FPN feature fusion network and P2 head layer in the model. After adding the LKAM, the mAP of Eagle-YOLO increased only 0.38% and AP50 increased 1.58%.

When we employed Eagle-IoU loss, the mAP and AP50 increased by 0.72% and 1.77%, respectively, which showed that the Eagle-IoU loss provided a stronger robustness in fewer categories and boxes containing noise. At the same time, the introduction of the $L1$ cost (R_{L1}) effectively avoided the problem of unstable gradients and slow convergence during the early training of the Euclidean distance.

Based on the ablation experiments discussed above, the Eagle-YOLO showed a significant improvement compared to the state-of-the-art baseline model.

To further verify the effectiveness of the Eagle-YOLO model, we employed the Gradient Class Activation Mapping (Grad-CAM [41]) method to interpret our model's predicted results visually. Grad-CAM is a backpropagation-based Explainable AI (XAI [42,43]) method that displays important areas of interest to the model by generating thermal maps. As shown in Figure 10 , the baseline model YOLOv5x showed a poor detection performance for small objects at longer distances despite having a good detection performance for close-range objects. However, the proposed model offers an improved effect when it comes to detecting larger objects at a short distance and small objects at a long distance. At the same time, compared with the baseline model, Eagle-YOLO is more accurate at positioning the object.

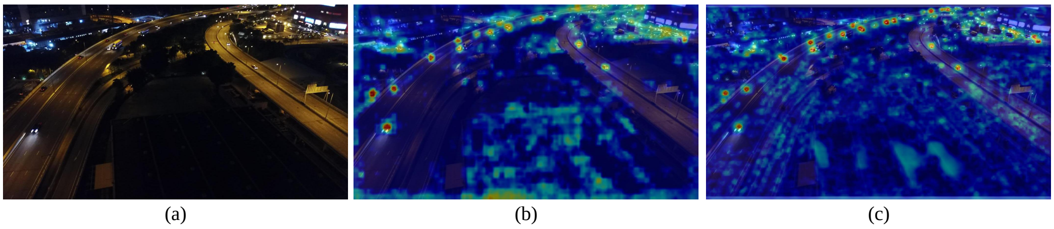
(a) (b) (c)

Figure 10. Schematic diagram of the model visualization of the Grad-CAM technique. (**a**) Original image. (**b**) Visualization of the YOLOv5x model. (**c**) Visualization of the Eagle-YOLO model.

Finally, we listed some typical scenarios in the VisDrone2021-DET-test dataset to visualize the effect of Eagle-YOLO, which is shown in Figure 11. The first row contains the raw images, the second row shows the results of the YOLOv5x model, and the third row shows the results of the Eagle-YOLO model. The green oval box indicates the object area that was not detected by YOLOv5x, and the red oval box indicates the improved part of our proposed model compared with YOLOv5x. The results showed that Eagle-YOLO is quite good at detecting small objects in dense, overlapping, and complex environments, such as low light at night. In addition, Eagle-YOLO is not only sensitive to small objects but also has a better detection performance for small numbers of categories.

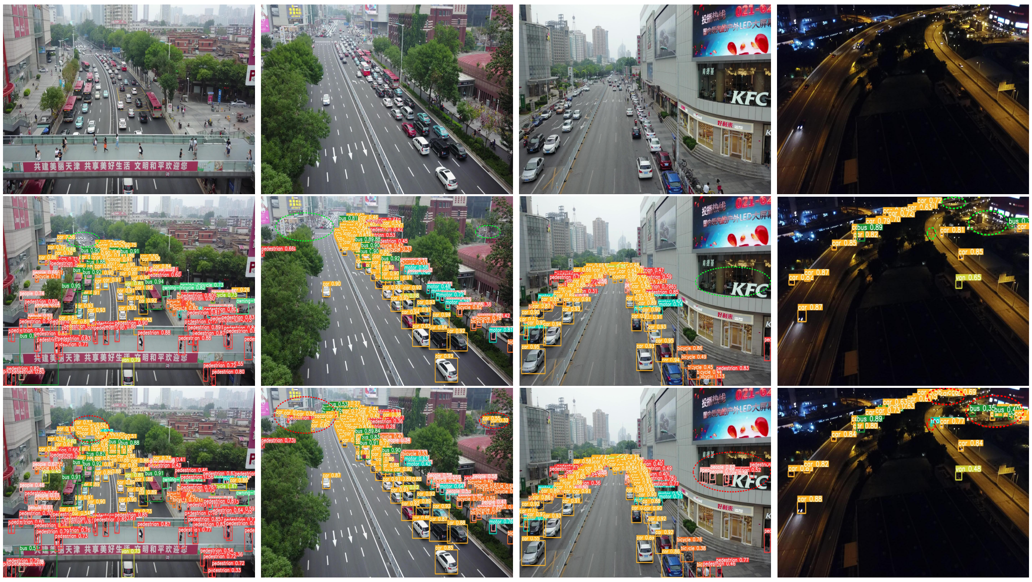

Figure 11. Some test results on the VisDrone2021-DET test. We selected some representative images for the display of the test results. The first row is the original image, the second row is the detection effect of YOLOv5x, and the third row is the detection result of Eagle-YOLO.

7. Conclusions and Future Work

To improve the object-detection accuracy in UAV aerial image scenes, we propose an Eagle-YOLO model inspired by the characteristics of eagles' eyes in this work. The model is based on YOLOv5 and is inspired by the hunting behavior of eagles. First, we introduced the Large Kernel Attention Module (LKAM) to make the model pay more attention to the detected regions and capture contextual information. Then, we combined the LKAM and the modified BiFPN feature fusion network to perform a contextual feature fusion to improve the detection accuracy when it comes to small objects in UAV aerial images. Finally, we propose a novel IoU loss named Eagle-IoU loss, which significantly improved the detection accuracy and model convergence speed. Compared to the baseline model YOLOv5x using the VisDrone2021-DET dataset, the experiments showed that Eagle-YOLO improved the mAP and AP50 by 2.86% and 4.23%, respectively.

In the future, we will continue to improve the structure of the model to lighten the network and increase the detection speed of the network without degrading its detection accuracy.

Author Contributions: Conceptualization, L.L. (Lyuchao Liao); Formal analysis, L.L. (Lyuchao Liao) and F.Z.; Writing—original draft, L.L. (Linsen Luo); Writing—review & editing, J.S. and Y.L.; Funding acquisition, Z.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China OF FUNDER grant number 41471333, 61304199. This research was funded by Fujian Provincial Department of Science and Technology OF FUNDER grant number 2021Y4019, 2020D002, 2020L3014, 2019I0019. This research was funded by Fujian University of Technology OF FUNDER grant number KF-J21012. This research was funded by Shenzhen Science and Technology Innovation Program OF FUNDER grant number JCYJ20220530160408019. This research was funded by Basic and Applied Basic Research Foundation of Guangdong Province OF FUNDER grant number 2023A1515011915. This research was funded by the Key Research and Development Project of Hunan Province of China OF FUNDER grant number 2022GK2020. This research was funded by Hunan Natural Science Foundation of China OF FUNDER grant number 2022JJ30171.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: http://aiskyeye.com/.

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

References

1. Wang, D.; Wang, J.G.; Xu, K. Deep learning for object detection, classification and tracking in industry applications. *Sensors* **2021**, *21*, 7349. [CrossRef] [PubMed]
2. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
3. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
4. Lin, T.Y.; 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: 13th European Conference, Zurich, Switzerland, 6–12 September 2014, Proceedings, Part V 13*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
5. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
6. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
7. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
8. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
9. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
10. Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Laughing, A.; Hogan, A.; Hajek, J.; Diaconu, L.; Marc, Y.; et al. *ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 Models AWS Supervise. ly and YouTube Integrations*; Zenodo: Geneva, Switzerland, 2021. [CrossRef]
11. 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.
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 2969239–2969250. [CrossRef] [PubMed]
13. 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.
14. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 19–25 June 2021; pp. 10012–10022.
15. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin transformer v2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019.
16. Wang, X.; He, N.; Hong, C.; Wang, Q.; Chen, M. Improved YOLOX-X based UAV aerial photography object detection algorithm. *Res. Sq.* **2022**. [CrossRef]
17. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October– 2 November 2019; pp. 9657–9666.
18. Yun, S.; Han, D.; Oh, S.J.; Chun, S.; Choe, J.; Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6023–6032.
19. Chen, J.N.; Sun, S.; He, J.; Torr, P.H.; Yuille, A.; Bai, S. Transmix: Attend to mix for vision transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12135–12144.
20. Tong, K.; Wu, Y.; Zhou, F. Recent advances in small object detection based on deep learning: A review. *Image Vis. Comput.* **2020**, *97*, 103910. [CrossRef]
21. Liu, Y.; Sun, P.; Wergeles, N.; Shang, Y. A survey and performance evaluation of deep learning methods for small object detection. *Expert Syst. Appl.* **2021**, *172*, 114602. [CrossRef]
22. Wang, G.; Xiong, Z.; Liu, D.; Luo, C. Cascade mask generation framework for fast small object detection. In Proceedings of the 2018 IEEE International Conference on Multimedia and Expo (ICME), San Diego, CA, USA, 23–27 July 2018; pp. 1–6.
23. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2117–2125.
24. 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, Waikoloa, HI, USA, 3–8 January 2022; pp. 1160–1168.
25. Zhang, T.; Zhang, X.; Ke, X. Quad-FPN: A novel quad feature pyramid network for SAR ship detection. *Remote Sens.* **2021**, *13*, 2771. [CrossRef]
26. Wang, C.; Zhong, C. Adaptive feature pyramid networks for object detection. *IEEE Access* **2021**, *9*, 107024–107032. [CrossRef]
27. Zhao, G.; Ge, W.; Yu, Y. GraphFPN: Graph feature pyramid network for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2763–2772.
28. Zuo, Z.; Tong, X.; Wei, J.; Su, S.; Wu, P.; Guo, R.; Sun, B. AFFPN: Attention Fusion Feature Pyramid Network for Small Infrared Target Detection. *Remote Sens.* **2022**, *14*, 3412. [CrossRef]

29. Zhu, L.; Lee, F.; Cai, J.; Yu, H.; Chen, Q. An improved feature pyramid network for object detection. *Neurocomputing* **2022**, *483*, 127–139. [CrossRef]
30. Stergiou, A.; Poppe, R.; Kalliatakis, G. Refining activation downsampling with SoftPool. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10357–10366.
31. Duan, H.; Xu, X.; Deng, Y.; Zeng, Z. Unmanned aerial vehicle recognition of maritime small-target based on biological eagle-eye vision adaptation mechanism. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 3368–3382. [CrossRef]
32. Guo, M.H.; Lu, C.Z.; Liu, Z.N.; Cheng, M.M.; Hu, S.M. Visual attention network. *arXiv* **2022**, arXiv:2202.09741.
33. He, J.; Erfani, S.; Ma, X.; Bailey, J.; Chi, Y.; Hua, X.S. α-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 20230–20242.
34. Cao, Y.; He, Z.; Wang, L.; Wang, W.; Yuan, Y.; Zhang, D.; Zhang, J.; Zhu, P.; Van Gool, L.; Han, J.; et al. VisDrone-DET2021: The vision meets drone object detection challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2847–2854.
35. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
36. Liu, S.; Huang, D.; Wang, Y. Receptive field block net for accurate and fast object detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 385–400.
37. Cai, Z.; Vasconcelos, N. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
38. Li, Z.; Peng, C.; Yu, G.; Zhang, X.; Deng, Y.; Sun, J. Light-head r-cnn: In defense of two-stage object detector. *arXiv* **2017**, arXiv:1711.07264.
39. Law, H.; Deng, J. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 734–750.
40. 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.
41. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
42. Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; Samek, W. Explainable AI methods-a brief overview. In Proceedings of the xxAI-Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, Vienna, Austria, 18 July 2020; Revised and Extended Papers; Springer: Berlin/Heidelberg, Germany, 2022; pp. 13–38.
43. Sun, J.; Lapuschkin, S.; Samek, W.; Binder, A. Explain and improve: LRP-inference fine-tuning for image captioning models. *Inf. Fusion* **2022**, *77*, 233–246. [CrossRef]

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