information

MDPI

Article

NATCA YOLO-Based Small Object Detection for Aerial Images

Yicheng Zhu, Zhenhua Ai, Jinqiang Yan, Silong Li, Guowei Yang and Teng Yu *

College of Electronic Information, Qingdao University, Qingdao 266071, China; yiiicheng@163.com (Y.Z.); aizhenhua@inspur.com (Z.A.); yanjinqiangyjq@163.com (J.Y.); fourdragonsli@163.com (S.L.); ygw_ustb@163.com (G.Y.)

* Correspondence: yutenghit@foxmail.com

Abstract: The object detection model in UAV aerial image scenes faces challenges such as significant scale changes of certain objects and the presence of complex backgrounds. This paper aims to address the detection of small objects in aerial images using NATCA (neighborhood attention Transformer coordinate attention) YOLO. Specifically, the feature extraction network incorporates a neighborhood attention transformer (NAT) into the last layer to capture global context information and extract diverse features. Additionally, the feature fusion network (Neck) incorporates a coordinate attention (CA) module to capture channel information and longer-range positional information. Furthermore, the activation function in the original convolutional block is replaced with Meta-ACON. The NAT serves as the prediction layer in the new network, which is evaluated using the VisDrone2019-DET object detection dataset as a benchmark, and tested on the VisDrone2019-DET-test-dev dataset. To assess the performance of the NATCA YOLO model in detecting small objects in aerial images, other detection networks, such as Faster R-CNN, RetinaNet, and SSD, are employed for comparison on the test set. The results demonstrate that the NATCA YOLO detection achieves an average accuracy of 42%, which is a 2.9% improvement compared to the state-of-the-art detection network TPH-YOLOv5.

Keywords: NAT; CA; Meta-ACON; small object detection

check for updates

Citation: Zhu, Y.; Ai, Z.; Yan, J.; Li, S.; Yang, G.; Yu, T. NATCA YOLO-Based Small Object Detection for Aerial Images. *Information* **2024**, *15*, 414. https://doi.org/10.3390/info15070414

Academic Editor: Vincenzo Moscato

Received: 12 June 2024
Revised: 9 July 2024
Accepted: 10 July 2024
Published: 18 July 2024

1. Introduction

In recent years, the use of UAVs has become more widespread in various fields, such as aerial imaging, traffic monitoring, and agricultural production. As a result, the processing of data captured from these devices and the effective extraction of feature information have become crucial [1]. Object detection serves as the foundation for these vision tasks. Currently, the two main frameworks for object detection are Anchor Based and Anchor Free.

Anchor-Based frameworks utilize predefined anchors in an image to detect objects. This involves classifying and regressing these anchors through a one-stage detection network or a two-stage network. One-stage detection networks like SSD [2] and YOLO [3–5] are fast but have lower accuracy rates. Two-stage detectors such as Faster R-CNN [6,7] and Cascade R-CNN [8] incorporate a region proposal network (RPN) to generate approximately 2000 proposal frames, followed by classification and localization. Although slower, two-stage detectors offer higher accuracy.

Anchor-Free detection does not rely on predefined boxes and instead focuses on localizing and regressing a few key points. CenterNet [9] predicts the object based on its centroid and achieves detection by predicting both the centroid and bounding box size. CornerNet [10] transforms object detection into a key point detection problem, determining the object's position and size through the detection of key points. The object is represented by two key points (the upper left corner and the lower right corner of the bounding box), which are then grouped based on their associated distance to achieve object detection.

While these generalized object detection models perform well on traditional datasets like MSCOCO [11], PASCALVOC [12], and ImageNet [13], they exhibit poor detection performance on UAV-acquired datasets like VisDrone [14] and UAVDT [15]. These datasets contain a large number of small objects (less than 32 pixels in size) and significant amounts of complex background information, which pose challenges for accurate detection.

To address these challenges, DPNet [16] enhances the feature extraction ability of convolutional neural networks (CNNs) by incorporating global context (GC) and deformable convolution (DC) into the backbone network, based on ResNeXt [17] and DenseNet [18]. Additionally, scale normalization and image pyramids are employed for multi-scale training.

Despite the improvements achieved by the aforementioned traditional models, there is still no specific and effective solution or algorithmic structure for the detection of small objects in the presence of complex backgrounds and variable object sizes. This paper proposes the addition of a NAT module to the last layer of the feature extraction network, replacing the original detection head with NAT. The NAT module incorporates domain attention (neighborhood attention, NA) [19], which performs attention operations on the domain of each pixel. The key change is constraining the global attention mechanism to operate locally within a sliding window and introducing more local bias attributes. This modification enhances object detection performance.

Furthermore, the paper introduces the CA module to the feature fusion network, mitigating the loss of detail information caused by 2D pooling. Unlike the convolutional block attention module (CBAM), which extracts channel attention into two parallel 1D feature encodings, the CA module integrates spatial information from different coordinates to retain a wider range of location information. As the network's depth increases, the resolution of the image decreases, resulting in fewer features for small objects on the feature map's extensive receptive field, thereby reducing detection results. To address this, an additional detection head is added and fused with the shallow feature map. The prediction layer is also replaced with NAT, resulting in more effective detection of small objects.

2. Related Work

2.1. Generalized Small Object Detection Methods

Data enhancement-based approach for small object detection: In their work, Yu et al. [20] introduced a scale matching strategy during the data preprocessing stage. This strategy involves cropping objects of different sizes to reduce the gap between them, ensuring scale matching and avoiding the loss of information about small objects caused by scaling operations. To address the issues of sparse distribution and lack of diversity in small object detection, Kisantal et al. [21] proposed a method called copy enhancement. This method increases the number of small object samples in the dataset by copying and pasting them multiple times, thereby improving the detection performance of small objects. Building upon Kisantal's method, Chen et al. [22] proposed the Rrnet algorithm. This algorithm utilizes an adaptive resampling strategy for data enhancement and incorporates a pre-trained semantic segmentation network to replicate the contextual information of object images.

Small object detection method based on multi-scale learning: To achieve better small object detection, it is crucial to leverage both deep semantic information and shallow representational information. Simonyan et al. [23] addressed this challenge by proposing the use of multiple small convolution kernels instead of a large one for multi-scale object detection. This approach improves the detection performance of small objects while maintaining a lower computational cost. Yu et al. [24] tackled the problem of multi-size object detection by introducing dilated convolution, which increases the receptive field for multi-scale contextual information aggregation without sacrificing resolution. Dai et al. [25] proposed deformable convolution, which arranges sampling points irregularly to address multi-scale and rotating object detection. Cai et al. [26] presented a unified multi-scale deep convolutional neural network that utilizes an inverse convolutional layer to enhance the resolution of the feature map, significantly improving small object detection performance while reducing memory resource usage and computational cost. Liang et al. [27] proposed a deep feature pyramid network based on FPN, which enhances the semantic information of small objects through the feature pyramid structure and lateral connections. Cao et al. [28] introduced contextual information to SSD through a multilevel feature fusion network, achieving a better balance between detection speed and accuracy for small objects. Li et al. [29] addressed the limitation of SSD in fully fusing feature maps between different layers by proposing a feature fusion single multi-frame detector. This detector employs a lightweight feature fusion module to connect and fuse features from each layer, generating a feature map that is then used for detection. This approach improves the detection accuracy of small objects with minimal loss of detection speed.

2.2. Aerial Image Small Object Detection Method

2.2.1. Dense Small Objects, Uneven Distribution of Objects

Yang et al. [30] proposed a clustered detection (ClusDet) network. This network consists of three sub-networks: the cluster proposal sub-network (CPNet), the scale estimation sub-network (ScaleNet), and the dedicated detection sub-network (DetectNet). CPNet is responsible for object clustering and generates object cluster regions. ScaleNet estimates the scale of the object clusters, while DetectNet detects objects in the cluster regions normalized to each scale. The ClusDet network significantly improves the detection performance of dense and small objects and implicitly models a priori contextual information, thereby enhancing inference speed. However, the network structure is complex, and training the two sub-networks is time-consuming. Additionally, the fixed setting of the cluster region does not align with the varying aerial image shooting viewpoints in actual scenes.

Wang et al. [31] proposed a cluster region estimation network (CRENet) for the problem of an uneven distribution of small targets in aerial images and the target scale across a large scale, which is a detection network with coarse to fine detection, mainly including three parts: (1) coarse network (CNet) searches the image containing dense targets using a clustering algorithm and then calculates the difficulty value of each clustered region to mine the difficult region and remove the simple clustered region; (2) cropping module uses Gaussian scaling function to search for regions containing dense targets. (1) Coarse network (CNet) searches for regions with dense targets using a clustering algorithm and then calculates the difficulty value of each clustered region to excavate the difficult regions and remove the simple clustered regions. (2) The cropping module scales the difficult clustered regions with a Gaussian scaling function to reduce the differences between the target scales and then feeds the difficult clustered regions into the fine network. (3) Fine network (FNet) uses a softmax algorithm to reduce the differences between the target scales and then feeds the difficult clustered regions into the fine network. Then, the difficult clustered regions are fed into the fine network. (3) Fine network (FNet) fuses the detection results of each region of interest (ROI) and the global image with softmax. Compared with ClusDet and DMNet, CRENet concentrates its computational resources on the region containing dense targets and eliminates the simple clustering region, which improves detection efficiency. Because the clustering region has different sizes, the clustering algorithm is used directly instead of the network to predict the clustering region, and the problem of duplicating the anchor and clustering region can be avoided.

Deng et al. [32] proposed an end-to-end global–local self-adaptive network (GLSAN) for the problem of dense small targets and uneven distribution, which mainly consists of three parts: global–local detection network (GLDN), self-adaptive region selecting algorithm (SARSA), and local super-resolution network (LSRN). GLSAN integrates the global–local fusion strategy into a progressive size-changing network. GLSAN integrates the global–local fusion strategy into a progressively size-varying network to achieve higher precision detection. The local fine detector can adaptively detect the target region detected by the global coarse detector with high resolution by cropping the original image, SARSA can dynamically crop the dense region in the input image, and LSRN can be used to increase the size of the cropped image to provide more detailed information to improve the detection accuracy of the detector. However, GLSAN also suffers from the problem of over-parameterization, and the targets in the corners of the image are easily ignored, resulting in missed detection.

2.2.2. Very Small Object Detection in Large-Scale Scenes

Li et al. [33] proposed the density map-guided object detection network (DMNet), which comprises three components: a density map generation network for generating density maps, an image cropping network for segmenting the input map into foregrounds based on the density map, and an object detector for detecting objects using the generated foregrounds. Nevertheless, the determination of the number of clipping blocks and the density threshold heavily relies on the designer's expertise, and the quality of the density map directly impacts the clipping results, subsequently affecting the accuracy of object detection.

Yu et al. [34] introduced the scale match (SM) strategy to address the issue of scale mismatch between the network pre-training dataset and the dataset utilized for detection learning. The scale-matching strategy aims to adjust the object scale between these two datasets, thereby enhancing the feature representation for detecting very small objects. By effectively leveraging the scale information of these objects, it has the potential to enhance detection performance. Nevertheless, it is imperative to note that this scale-matching approach is currently limited to a basic image-level method and necessitates further refinement.

In order to solve the problem of very small target detection in aerial images, Xu et al. [35] proposed the dot distance (DotD) metric. The sensitivity of intersection over union (IOU) to offset is shown in the fact that a small relative movement between two bounding boxes will lead to a large change in IOU and a large number of positive samples are wrongly categorized as negative samples in RPN, which makes it difficult for the training to converge. At the same time, the NMS may treat some real prediction frames as redundant frames, which seriously affects the detection accuracy of the performance targets. DotD is the Euclidean distance between the centers of the two normalized bounding boxes, which is insensitive to small offsets and has better performance than the IOU series in detecting very small targets. Applying DotD to the region proposal network (RPN) and none-maximum suppression (NMS) modules can achieve better detection performance. DotD can also be easily inserted into many anchor-based detectors, and DotD can be used as a threshold for deciding positive and negative samples, which solves the problem that some positive samples will be regarded as negative samples under the IOU.

Wang et al. [36] proposed the normalized Wasserstein distance (NWD) in order to solve the problem of detecting very small targets in aerial images. Firstly, the bounding box is modeled as a two-dimensional Gaussian distribution, and then the NWD is used to measure the similarity of the Gaussian distribution. The biggest advantage of the NWD is that it can measure the similarity of the distribution even if the two bounding boxes do not overlap or contain each other. The biggest advantage of NWD is that the distribution similarity can be measured even if the two bounding boxes do not overlap or include each other. In addition, NWD is insensitive to targets with different scales, so it is suitable for measuring the similarity between very small targets. NWD can be used in both single-phase and multi-phase detectors, and it can be used instead of IOUs in label assignment, NMS, and loss function to solve the problem of sensitivity to positional shifts in IOUs. The problem of sensitivity to positional offset in IOU is solved.

3. Proposed Method

3.1. Overall Network Architecture

The NATCA YOLO network is specifically designed for small object detection in aerial images. The overall architecture of the network consists of three main parts: the feature extraction part (Backbone), the feature fusion part (Neck), and the prediction layer (Head). The Backbone is responsible for extracting features from the input image and generating a feature map. The Neck utilizes the features extracted at different levels in the Backbone stage to aggregate and reprocess them. The Head is responsible for detecting the location and category of the object by utilizing the Neck to aggregate the refined feature maps. The network structure of the NATCA YOLO is illustrated in Figure 1.

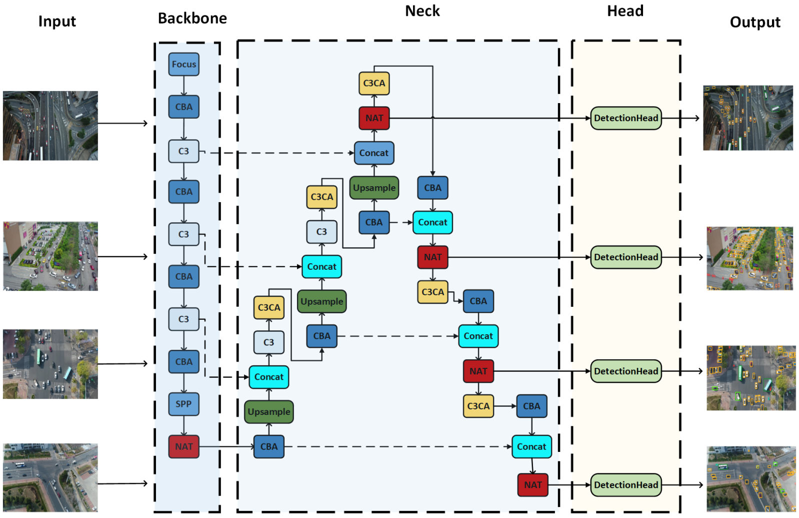


Figure 1. NATCA YOLO network architecture.

In the feature extraction layer of NATCA-YOLOv5, the Meta-ACON [37] activation function is used instead of the SiLU activation function in the original network. Additionally, behind the SPP layer, NAT is added and the features of different layers are fed into the Neck part. After three rounds of up-sampling, the NAT is employed instead of the prediction layer of the original network, enabling the detection of four different object sizes, ranging from smallest to largest.

3.2. Neighborhood Attention Converter

For visual tasks, such as object detection, the image resolution is typically much higher compared to classification tasks. In the case of aerial images, the resolution can be exceptionally high. However, when the image resolution is high, the self-attention (SA) mechanism in the vision Transformer (ViT) [38] introduces excessive complexity and computation, negatively impacting the model's performance in visual tasks. Convolutional networks primarily benefit from their inductive bias, whereas SA is a global operation. Although the multi-layer perceptron (MLP) layer provides some localization and translation invariance, other inductive biases need to be compensated for by a large amount of data. Consequently, the ViT model is less effective on the VisDrone dataset, which has limited samples.

The neighborhood attention Transformer (NAT) is a hierarchical vision converter based on NA. The structure of the NAT network is depicted in Figure 2.

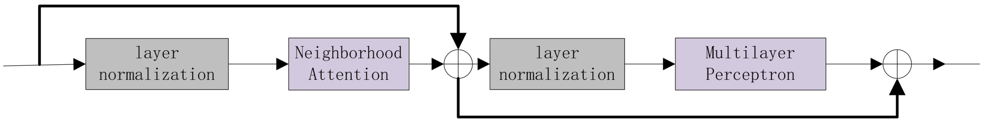


Figure 2. Neighborhood attention converter network architecture.

As a comparison to self-attention (SA), NAT adopts NA instead of SA, which restricts the global attention mechanism to local attention within a local window. This modification introduces more local bias properties and enhances NAT's performance, even surpassing CNN on small sample datasets like VisDrone. NAT also demonstrates superior feature extraction capabilities and preserves more global context information compared to ViT. Furthermore, NAT maintains the hierarchical pyramid structure, incorporating downsampling operations after each layer to reduce size by half. Unlike non-overlapping convolutions, NAT employs small kernel convolutions for embedding and downsampling. The simplicity and flexibility of neighborhood attention are illustrated in Figure 3.

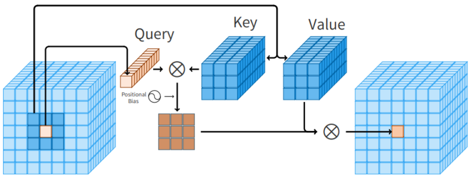


Figure 3. The architecture of neighborhood attention.

The selection of the key vector, Key, and the value vector, Value, from a neighborhood centered on the question vector, Query, is enforced by it. In comparison to SA, NA reduces computational cost and introduces a local inductive bias similar to convolution. When the neighborhood size reaches the maximum (i.e., the input image size), NA degrades to a neighborhood attention mechanism. Similarly, NA degrades to SA when the neighborhood size reaches its maximum (i.e., the size of the input image). NAT utilizes overlapping convolutions to downsample the feature mapping between different levels. NA restricts each pixel in the feature map to solely concentrate on its adjacent pixels. In this study, $\rho_{(i,j)}$ is used to denote the neighborhood of a pixel at (i,j), which is a fixed-length set of indexes of the closest pixel to (i,j). For size $L \times L$,$\|\rho(i,j)\| = L^2$, and thus the single pixel on the NA of a single pixel is defined as

$$NA(X_{i,j}) = softmax(\frac{Q_{i,j}K^T_{\rho_{(i,j)}} + B_{i,j}}{scale})V_{\rho(i,j)} \tag{1}$$

where, for the input vector X, the problem vector Q, the key vector K, and the value vector V are the linear projections of X, respectively; *scale* is the normalization scaling factor, and $B_{i,j}$ is the relative positional bias, which is added to each attention based on the relative position weights. This operation can be further extended to all pixels (i,j), resulting in a form of localized attention.

When $\rho_{i,j}$ maps each pixel to all pixels, the

$$K_{\rho(i,j)} = K \tag{2}$$

$$V_{\rho(i,j)} = V \tag{3}$$

NA degenerates into SA with positional bias, and after removing the bias term, the expression of SA can be obtained as

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d}})V \tag{4}$$

In this study, Q represents the problem vector, K represents the key vector, V represents the value vector, and $\sqrt{d}$ is the scaling parameter. d represents the dimension of Q and K.

The operation depicted in Figure 3 is iteratively performed for every pixel in the feature map. For corner pixels that cannot be centered, the sense field is enlarged by expanding the neighborhood in order to maintain its small size. For example, for $L = 3$, each query will end with nine key−value pixels surrounding it (a 3 × 3 grid centered on the query). For corner pixels, the neighborhood is also a 3 × 3 grid, but the query is not in the center.

3.3. Coordinate Attention Mechanism

CNN is capable of effectively amalgamating feature information from various layers. However, as the network becomes more complex, pooling operations lead to the loss of intricate details between channels and feature positions. Following the convolution of the input feature map, each position encompasses information pertaining to a local region of the original image. The convolutional block attention module (CBAM) computes the maximum and average of multiple channels at each position to serve as weighting coefficients. Nevertheless, this approach solely considers local region information and fails to capture long-range dependencies. By employing the coordinate attention mechanism, channel attention is decomposed into two parallel one-dimensional feature encodings. Subsequently, these encodings contain directional information, enabling each feature map to acquire longer-range information along a spatial direction within the input feature maps. Consequently, the model can more accurately locate and identify object areas. The coordinate attention mechanism is illustrated in Figure 4.

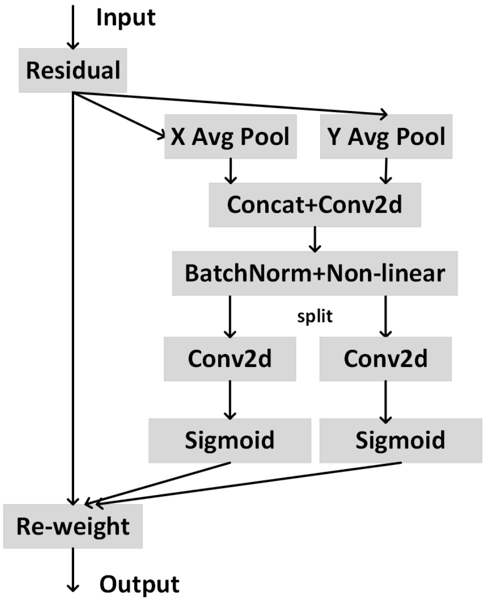


Figure 4. The architecture of coordinate attention.

To address the challenge of preserving position information while compressing it into channel information for channel attention, we propose a novel approach. Firstly, we perform one-dimensional global pooling on the two dimensions, allowing us to obtain longer dependency information in one direction while still retaining position information in the other direction. For input x, use $(H, 1)$ and $(1, W)$ to encode each channel along the horizontal and vertical coordinates, respectively, that is

$$z_c^h(h) = \frac{1}{w}\Sigma_{0<i<w} x_c(h, i) \tag{5}$$

$$z_c^w(h) = \frac{1}{H}\Sigma_{0<j<w} x_c(j, w) \tag{6}$$

where $z_c^h(h)$ denotes the feature vector after global pooling of spatial information on the channel along the horizontal direction, $z_c^w(h)$ denotes the feature vector after global pooling of spatial information on the channel along the vertical direction, x_c denotes the input feature vector, W denotes the width of the feature map, and H denotes the height of the feature map.

The two pooled vectors are concatenated along the spatial dimensions, followed by a 2D convolution to decrease the number of channels and model complexity. Subsequently, regularization techniques and nonlinear activation functions are applied, which is

$$f = \delta(F_1(\left[z^h, z^w\right])) \tag{7}$$

where f denotes the output feature map, δ denotes the nonlinear activation function, F_1 denotes the regularization, z^h denotes the vector encoded along the horizontal coordinates, and z^w denotes the vector encoded along the vertical coordinates.

f is divided into two tensors $(c/r, 1, H)$ and $(c/r, 1, w)$ along the spatial dimension, and then recovered to the same number of channels as the input x_c by convolution with the same number of channels, and finally weighted by *sigmoid* normalization, that is

$$g^h = \sigma(F_h(f^h)) \tag{8}$$

$$g^w = \sigma(F_w(f^w)) \tag{9}$$

where g^h denotes the horizontal component with the same number of channels as the input x_c, F_h denotes the horizontal regularization, f^h is the horizontal component of f, g^w denotes the vertical component with the same number of channels as the input x_c, F_w is the vertical component of f^w, and σ is the *sigmoid* activation function. Equating g^h and g^w as weights, attention along the horizontal and vertical directions is applied to the input tensor simultaneously to obtain coordinate attention as

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j) \tag{10}$$

where $y_c(i,j)$ denotes the coordinate attention at pixel (i,j), $g_c^h(i)$ denotes the equivalent weight in the horizontal direction, and $g_c^w(j)$ denotes the equivalent weight in the vertical direction.

In the deep feature space, the channel attention (CA) mechanism operates by employing two one-dimensional global pooling operations. This enables the network to acquire a broader sensory field and encode precise spatial location information. Consequently, the object region can be detected and localized with greater accuracy.

3.4. Activation Function

In order to enhance the model's generalization ability, we substitute the activation function in the convolution module of the original network with Meta-ACON [33], which allows for the adaptive activation of neurons. The ReLU activation function is essentially a MAX function, and its differentiable variant for smoothing maxima when considering only two inputs can be represented as follows. The differentiable variant of the smoothed maxima of ReLU is

$$ReLU(x) = MAX(0,x) == \begin{cases} o & x < 0 \\ x & x \geq 0 \end{cases} \tag{11}$$

It can be seen that the value and the gradient are both 0 in the part less than 0, while the derivative remains 1 in the part greater than 0

$$\begin{aligned} S_\beta(\eta_a(x), \eta_b(x)) =& (\eta_a(x) - \eta_b(x)) \\ & * \sigma[\beta(\eta_a(x) - \eta_b(x))] + \eta_b(x) \end{aligned} \tag{12}$$

where σ denotes the *sigmoid* function; β is a smoothing factor, and Smooth Maximum is the standard MAX function when β tends to infinity, and Smooth Maximum is the arithmetic mean when β is zero. $\eta_a(x)$and $\eta_b(x)$ denote linear functions.

$S_\beta(0,x)$ is the ReLU in the smooth form based on which the activate or not (ACON) series of activation functions, of which ACON-C is one of the most widespread forms, is

$$\begin{aligned} ACON - C(x) =& S_\beta(p_1 x, p_2 x) = (p_1 - p_2)x \\ & * \sigma[\beta(p_1 - p_2)x] + p_2 x \end{aligned} \tag{13}$$

Meta-ACON switches the activation to activate or not as the switching factor β controls it to be non-linear or linear. The adaptive function for calculating channel-wise β is

$$\beta_c = \sigma W_1 W_2 \sum_{h=1}^{H} \sum_{w=1}^{W} x_{c,h,w} \quad (14)$$

where H denotes the height of the feature map, W denotes the width of the feature map, $x_{c,h,w}$ denotes the number of input feature vectors of c, height of h and width of w, σ is the sigmoid activation function, and $W1(c, c/r)$ and $W2(c/r, c)$ are used to save the parameter counts, with the number of channels being c, and the scaling parameter r.

Meta-ACON-C's β is learned channel-wise, and a 1 × 1 convolution was used to let each channel share a weight. However, 1 × 1 convolution increases the number of parameters, so a scaling parameter r(r = 16 by default) is added and two 1 × 1 convolutions are performed, reducing the number of parameters and allowing each channel to share a weight.

4. Experimental Results and Analysis

4.1. Activation Function

The VisDrone2019-DET benchmark dataset comprises 10,209 still images captured by a UAV platform in diverse scenarios. The dataset is divided into a training set (6471 images), a validation set (548 images), a test set (1610 images), and a test challenge set (1580 images). UAV-captured images exhibit significant scale variations, complex background interference, and predominantly small objects (less than 32 pixels) with variable viewpoints. Furthermore, there are noticeable differences even for the same object due to these factors. The dataset contains 10 categories of data, with an imbalanced distribution, where pedestrians and cars constitute a larger portion of the data. The data distribution of VisDrone2019-DET is illustrated in Figure 5. In this paper, all the models are trained on the training set and evaluated on the test set.

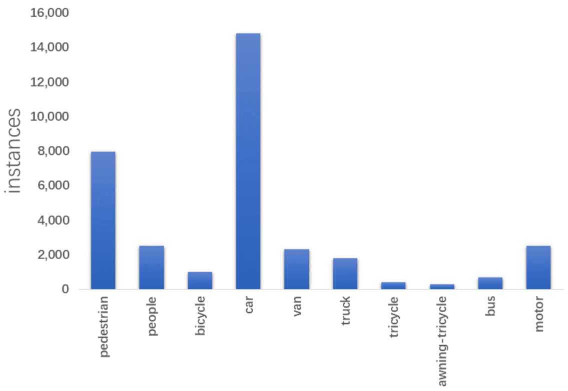


Figure 5. VisDrone2019-DET data distribution.

4.2. Evaluation Indicators

Similar to the evaluation metrics employed in MS COCO [11], this paper assesses the performance of the detection algorithms using the mean accuracy (mAP). The mAP is determined by averaging the results obtained at 10 intersection and intersection over union (IOU) thresholds, ranging from 0.50 to 0.95 with a uniform step size of 0.05 across all categories. It serves as the primary metric for evaluating the performance of the algorithm in object detection.

4.3. Contrast Test and Ablation Test

The algorithm presented in this paper is implemented using the PyTorch deep learning framework and YOLOv5. The NVIDIA GeForce RTX 2080Ti 11GB graphics card is used for model training. Stochastic gradient descent (SGD) is employed as the optimizer with a weight decay of 0.0005. The initial training includes three rounds of warm-up training, where the momentum of SGD is set to 0.8. The learning rate is updated using a one-dimensional linear difference during warm-up. After warm-up, a cosine annealing function is used to reduce the learning rate, starting from an initial value of 0.02 and reaching a minimum value of 0.2 × 0.01. The model is trained for 100 rounds.

To evaluate the effectiveness of the NATCA YOLO model in small object detection tasks in aerial images, comparative testing is conducted using Faster R-CNN, RetinaNet [39], SSD, FCOS [40], DETR [41], and TPH-YOLOv5 [42] detection networks on the test set. The mean average precision (mAP) is used as the main evaluation metric, and the comparison results are shown in Table 1.

Table 1. Comparison results of the average accuracy of each model.

Model	mAP%
Faster R-CNN	19.32
RetinaNet	21.02
SSD	20.36
FCOS	21.14
DETRZ	38.02
TPH-YOLOv5	39.10
NATCA-YOLO(ours)	42.00

From Table 1, it can be observed that the NATCA-YOLOv5 model exhibits stronger generalization ability compared to traditional CNN models.

To analyze the impact of each module on the NATCA-YOLOv5 model, ablation experiments are performed to assess the effects of the coordinate attention module, coordinate attention mechanism and Meta-ACON activation function. The average accuracy results for each model are presented in Tables 2 and 3.

Table 2. Results of ablation experiments with average accuracy for each model.

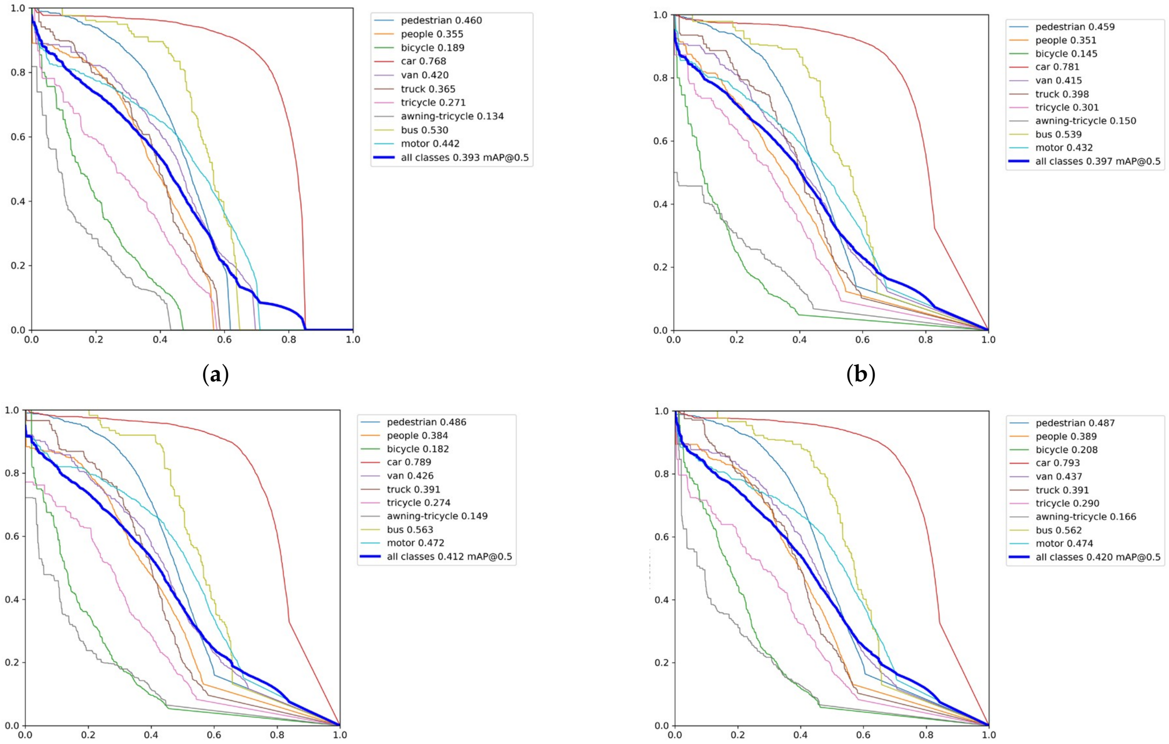
Model	mAP%
YOLOv5	28.9
YOLOv5+NAT	39.3
YOLOv5+NAT+CBAM	39.7
YOLOv5+NAT+CA	41.2

Table 3. Results of ablation experiments with isolation model.

Model	mAP%
NATCA-YOLO	42.0
Our-NAT	39.3 (−2.7)
Our-ACON	41.2 (−0.8)
Our-CA	40.6 (−1.4)

As shown in Table 2, YOLOv5 serves as the basic model structure without the NAT module and coordinate attention, using the Meta-ACON [33] activation function in the convolution block. Incorporating the NAT module into the last layer of the Backbone and replacing the prediction layer of the original network with the NAT module leads to a 10.4% improvement in AP metrics compared to YOLOv5. The addition of the coordinate attention module (CA) improves AP metrics by 1.5% compared to the CBAM module. Replacing the activation function of the convolutional layer in the original network with the Meta-ACON activation function further boosts the AP metric to 42%. These experiments demonstrate that the integration of the NAT module and coordinate attention, along with the use of the Meta-ACON activation function, enhances the accuracy and generalization ability of the model in small object detection tasks on a limited aerial image dataset.

The accuracy vs. recall (PR) curve for the model predictions is illustrated in Figure 6.



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

(**c**) (**d**)

Figure 6. Accuracy vs. recall curves for model predictions (the y-axis is precision and the x-axis is recall). (**a**) PR Curves for YOLOv5 + NAT; (**b**) PR Curves For YOLOv5 + NAT + CBAM; (**c**) PR Curves for YOLOv5 + NAT + CA; (**d**) PR Curves for YOLOv5 + NAT + CA NATCA YOLO.

As illustrated in Figure 6, the addition of the NAT module to the YOLOv5 network resulted in AP metrics of 39.3% for all categories, with 76.8% for cars and 46% for pedestrians. This performance is considered good, especially considering the small sample dataset. The inclusion of the convolutional block attention module did not significantly improve the overall performance, with an average accuracy of 39.7%. However, the addition of coordinate attention in the deep network allowed for information exchange not only between different channels but also within the same channels, as well as obtaining longer-range information through position encoding. This led to an improvement in the AP metrics for all categories, reaching 41.2%, with 78.9% for cars and 48.6% for pedestrians, representing a 2.1% and 2.6% improvement compared to Figure 6a

To further enhance the detection accuracy, the activation function in the original network's convolutional block was replaced with Meta-ACON, resulting in the achievement of AP metrics of 42% for NATCA YOLO in all categories. This improvement is particularly significant for objects with fewer samples and similar categories, such as bicycles, tricycles, and awning tricycles.

We split the Visdrone dataset, sampled without repetition, into three parts, Fold1, Fold2, and Fold3, taking turns using two of them as the training data and one as the test data and conducting unrepeated experiments in different models.

As shown in Table 4, model3 gives the best results, so we follow the results from this training on the test set.

Table 4. k-fold cross-validation results for NATCA-YOLO.

Model	Model 1	Model 2	Model 3
mAP%	47.5	48.3	**51.3**

Figure 7 clearly shows the detection instance performance of several models on the same dataset. For complex background images, small target images, dense target images, and images with uneven target distribution, such as the circular obstacle in the middle of the first column of images (marked with a red circle in the figure), VistrongerDet, ViT-YOLO, and TPH-YOLOv5 models have all undergone false detections. The model proposed in this paper achieved accurate detection results.

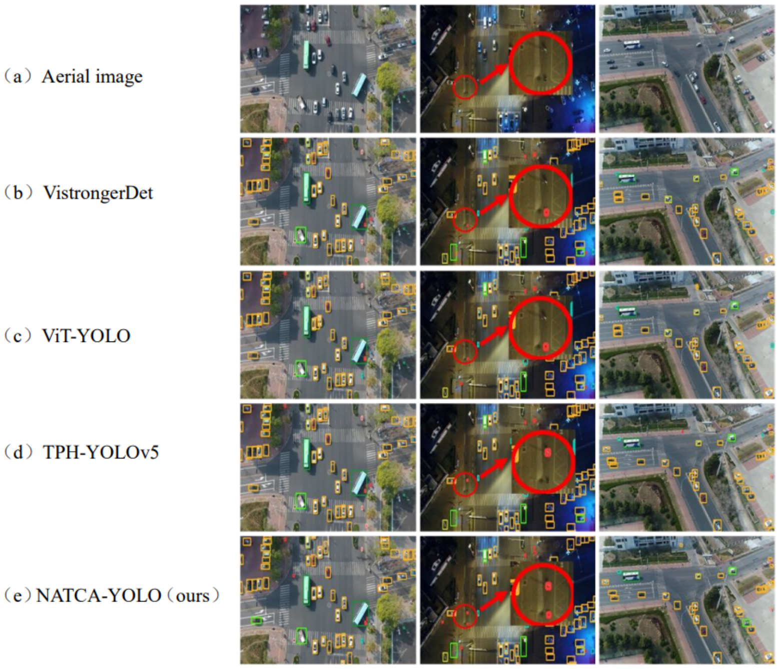


Figure 7. Comparison of detection performance of different models on the VisDrone Net dataset.

The detection results of NATCA YOLO in various scenarios are presented in Figure 7.

Table 5 shows the comparison of the quantitative results of different aerial photography small target detection models on the VisDrone dataset, and compared with other mainstream aerial photography small target detection models, the method proposed in this paper has better detection performance.

Table 5. Quantitative comparison of detection results of different models on VisDrone dataset.

Model	mAP%	AP50%
VistrongerDet [43]	33.72	56.42
ViT-YOLO [44]	38.6	62.81
TPH-YOLOv5	39.18	63.3
NATCA-YOLO(ours)	42	66.27

From Figure 8, it is evident that the model proposed in this paper effectively detects dense small objects in Figure 8a, eliminates the interference of the rectangular green background in Figure 8b, and accurately detects objects of different scales in Figure 8c. In summary, the NATCA YOLO model demonstrates accurate localization and identification of objects, even in the presence of challenges such as complex background interference, small objects, and a wide range of object sizes.

We analyze the inference speed of our method and other methods for different input sizes and on the same platform. Our inference speed is measured in "milliseconds". We test it on the same platform NVIDIA GeForce 2080 Ti. In addition, we set the size of the input images to small (1 × 3 × 320 × 320), medium (1 × 3 × 640 × 640), and large (1 × 3 × 960 × 960). The final results are shown in Table 6. The results show that our model improves the detection accuracy while the detection speed is well maintained.

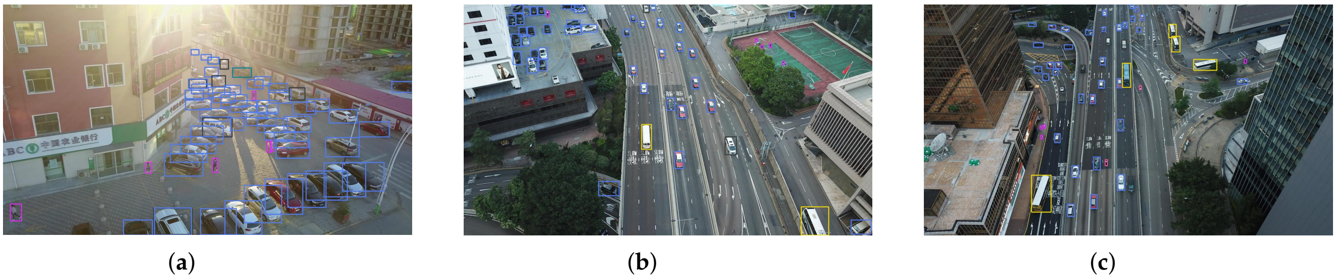


Figure 8. Detection results of NATCA YOLO in different scenarios. (**a**) Small intensive objects; (**b**) complex context; (**c**) multiple object sizes.

Table 6. Quantitative comparison results of the inference efficiency among different methods on different input sizes (320 × 320, 640 × 640, 960 × 960).

Model	320 × 320	640 × 640	960 × 960
ViT-YOLO	12.2 ms (100 FPS)	16.5 ms (73 FPS)	31.1 ms (32 FPS)
TPH-YOLOv5	8.4 ms (124 FPS)	12.5 ms (97 FPS)	26 ms (45 FPS)
NATCA-YOLO(ours)	9.3 ms (109 FPS)	14.6 ms (83 FPS)	28.1 ms (37 FPS)

5. Limitation and Future Work

Although the proposed neighborhood attention Transformer aerial small target detection method in this paper reflects a more excellent performance in the small target detection task but the method in this paper is not ideal in foggy weather. Similar target detections affect similar targets due to more similar features, resulting in poor detection. Therefore, it is hoped that in future work we will try to improve the network prediction classifier to solve the problem of similar target detection difficulty through multi-level label prediction. Additionally, the experimental results obtained in the computer environment may differ when the model is deployed on the UAV due to limited computational and space resources. Therefore, further research should focus on improving the algorithm to increase the detection accuracy of similar objects and designing a lightweight network structure to reduce model complexity while maintaining high detection accuracy.

6. Conclusions

This paper investigates deep learning-based small object detection for aerial images, analyzing the basic principles and limitations of general object detection models and existing aerial object detection models. Furthermore, it examines the characteristics of aerial images, such as the presence of numerous small objects, a wide range of object sizes, and complex backgrounds. By comparing existing aerial object detection algorithms, a new network architecture called NATCA is proposed based on the YOLOv5 model. To address the issue of poor detection accuracy of small objects, a neighborhood attention converter is added, and feature information is extracted from a shallower layer of the network for prediction. Additionally, coordinate attention is incorporated to eliminate interference from complex background information and retain global and contextual information. The activation function of the original network is also improved to enhance the model's generalization ability. Experimental results demonstrate that the proposed network model exhibits stronger visual information, feature extraction ability, and generalization ability compared with the generalized object model, resulting in significantly improved detection accuracy. UAVs offer advantages such as low cost, high flexibility, and simple operation, and their application scenarios are more diverse compared to aerial remote sensing technology. However, directly applying generalized object detection models to UAVs is less effective due to the diverse viewpoints of aerial images. Therefore, studying object detection algorithms specifically applicable to UAVs is of great significance for their applications.

Author Contributions: Conceptualization, Y.Z.; Resources, Z.A., J.Y., S.L. and G.Y.; Supervision, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

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

Abbreviations

The following abbreviations are used in this manuscript:

NAT	Neighborhood Attention Transformer
SA	Self-Attention
CBAM	Convolutional Block Attention Module
CA	Coordinate Attention

References

1. Liu, F.; Wu, Z.; Yang, A.; Han, X. Adaptive UAV object detection based on multi-scale feature fusion. *J. Opt.* **2020** *40*, 133–142.
2. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Amsterdam, The Netherlands, 2016; pp. 21–37.
3. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once:Unified,real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 779–788.
4. Puyi, S.; Hong, C.; Haobo, G. Improved UAV object detection algorithm for YOLOv5s. *Comput. Eng. Appl.* **2023**, *59*, 108–116.
5. Qi, J.; Wu, L.; Lu, F.; Shi, H.; Xu, F. UAV object detection based on improved YOLOv4 algorithm. *J. Weapons Equip. Eng.* **2022**, *43*, 210–217.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN:Towards real-time object detection with region proposal networks. *Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. Jun, S.; Pei, G.; Zhili, X. Faster-RCNN for car model identification analysis. *J. Chongqing Univ.* **2017**, *40*, 32–36.
8. 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; IEEE: Piscataway, NJ, USA, 2018; pp. 6154–6162.
9. Duan, K.; Bai, S.; Xie, L.; Qi, H.; Huang, Q.; Tian, Q. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27–28 October 2019; IEEE/CVF: Piscataway, NJ, USA, 2019; pp. 6569–6578.
10. 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; Springer: Berlin/Heidelberg, Germany, 2018; pp. 734–750.
11. 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 Proceedings of the European Conference on Computer Vision, Zürich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
12. 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]
13. 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; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.

14. Du, D.; Zhu, P.; Wen, L.; Bian, X.; Lin, H.; Hu, Q.; Peng, T.; Zheng, J.; Wang, X.; Zhang, Y.; et al. VisDrone-DET2019: The vision meets drone object detection in image challenge results. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019; IEEE/CVF: Piscataway, NJ, USA, 2019.
15. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer vision (ECCV), Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 370–386.
16. Zhou, Q.; Shi, H.; Xiang, W.; Kang, B.; Wu, X.; Latecki, L.J. DPNet: Dual-Path Network for Real-time Object Detection with Lightweight Attention. *arXiv* **2022**, arXiv:abs/2209.13933.
17. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
18. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
19. Hassani, A.; Walton, S.; Li, J.; Li, S.; Shi, H. Neighborhood attention transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 6185–6194.
20. Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; Han, Z. Scale match for tiny person detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 13–19 June 2020; pp. 1257–1265.
21. Kisantal, M.; Wojna, Z.; Murawski, J.; Naruniec, J.; Cho, K. Ugmentation for small object detection. *arXiv* **2019**, arXiv:1902.07296.
22. Chen, C.; Zhang, Y.; Lv, Q.; Wei, S.; Wang, X.; Sun, X.; Dong, J. Rrnet: A hybrid detector for object detection in drone-captured images. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019.
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
24. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. *arXiv* **2015**, arXiv:1511.07122.
25. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 764–773.
26. Cai, Z.; Fan, Q.; Feris, R.S.; Vasconcelos, N. A unified multi-scale deep convolutional neural network for fast object detection. In Proceedings of the IEEE International Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part IV 14; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 354–370.
27. Liang, Z.; Shao, J.; Zhang, D.; Gao, L. Small object detection using deep feature pyramid networks. In Proceedings of the Advances in Multimedia Information Processing-PCM 2018: 19th Pacific-Rim Conference on Multimedia, Hefei, China, 21–22 September 2018; Proceedings, Part III 19; Springer International Publishing: Berlin/Heidelberg, Germany, 2018; pp. 554–564.
28. Cao, G.; Xie, X.; Yang, W.; Liao, Q.; Shi, G.; Wu, J. 0 Feature-fused SSD: Fast detection for small objects. In Proceedings of the Ninth International Conference on Graphic and Image Processing (ICGIP 2017), Qingdao, China, 14–16 October 2018; Volume 10615, pp. 381–388.
29. Li, Z.; Zhou, F. FSSD: Feature fusion single shot multibox detector. *arXiv* **2017**, arXiv:1712.00960.
30. Yang, F.; Fan, H.; Chu, P.; Blasch, E.; Ling, H. Clustered object detection in aerial images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8311–8320
31. Wang, Y.; Yang, Y.; Zhao, X. Object detection using clustering algorithm adaptive searching regions in aerial images. In Proceedings of the Computer Vision—ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; Proceedings, Part IV 16; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 651–664.
32. Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; Qin, H. A global-local self-adaptive network for drone-view object detection. *IEEE Trans. Image Process.* **2020**, *30*, 1556–1569. [CrossRef] [PubMed]
33. Li, C.; Yang, T.; Zhu, S.; Chen, C.; Guan, S. Density map guided object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 190–191.
34. Jiang, N.; Yu, X.; Peng, X.; Gong, Y. SM+: Refined scale match for tiny person detection. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 1815–1819.
35. Xu, C.; Wang, J.; Yang, W.; Yu, L. Dot distance for tiny object detection in aerial images. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 1192–1201.
36. Xu, C.; Wang, J.; Yang, W.; Yu, H.; Yu, L.; Xia, G.S. A normalized Gaussian Wasserstein distance for tiny object detection. *arXiv* **2021**, arXiv:2110.13389.
37. Ma, N.; Zhang, X.; Liu, M.; Sun, J. Ctivate or not: Learning customized activation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 8032–8042.
38. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
39. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 2980–2988.
40. Tian, Z.; Shen, C.; Chen, H.; He, T. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE/CVF: Piscataway, NJ, USA, 2019; pp. 9627–9636.

41. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
42. 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; IEEE: Piscataway, NJ, USA, 2021; pp. 2778–2788.
43. Wan, J.; Zhang, B.; Zhao, Y.; Du, Y.; Tong, Z. Vistrongerdet: Stronger visual information for object detection in visdrone images. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 10–17 October 2021; pp. 2820–2829.
44. Zhang, Z.; Lu, X.; Cao, G.; Yang, Y.; Jiao, L.; Liu, F. ViT-YOLO: Transformer-based YOLO for object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2799–2808.

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.