remote sensing

MDPI

Article

SAFF-SSD: Self-Attention Combined Feature Fusion-Based SSD for Small Object Detection in Remote Sensing

Bihan Huo, Chenglong Li, Jianwei Zhang *, Yingjian Xue and Zhoujin Lin

School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing 210044, China; 20211215022@nuist.edu.cn (B.H.); 202212150015@nuist.edu.cn (C.L.); 20211215007@nuist.edu.cn (Y.X.); 20211215025@nuist.edu.cn (Z.L.)

* Correspondence: zhangjw@nuist.edu.cn

Abstract: SSD is a classical single-stage object detection algorithm, which predicts by generating different scales of feature maps on different convolutional layers. However, due to the problems of its insufficient non-linearity and the lack of semantic information in the shallow feature maps, as well as the fact that small objects contain few pixels, the detection accuracy of small objects is significantly worse than that of large- and medium-scale objects. Considering the above problems, we propose a novel object detector, self-attention combined feature fusion-based SSD for small object detection (SAFF-SSD), to boost the precision of small object detection. In this work, a novel self-attention module called the Local Lighted Transformer block (2L-Transformer) is proposed and is coupled with EfficientNetV2-S as our backbone for improved feature extraction. CSP-PAN topology is adopted as the detection neck to equip feature maps with both low-level object detail features and high-level semantic features, improving the accuracy of object detection and having a clear, noticeable and definitive effect on the detection of small targets. Simultaneously, we substitute the normalized Wasserstein distance (NWD) for the commonly used Intersection over Union (IoU), which alleviates the problem wherein the extensions of IoU-based metrics are very sensitive to the positional deviation of the small objects. The experiments illustrate the promising performance of our detector on many datasets, such as Pascal VOC 2007, TGRS-HRRSD and AI-TOD.

Keywords: convolutional neural network; feature fusion; transformer; EfficientNetV2-S; CSP-PAN; NWD metric; small object detection

Citation: Huo, B.; Li, C.; Zhang, J.; Xue, Y.; Lin, Z. SAFF-SSD: Self-Attention Combined Feature Fusion-Based SSD for Small Object Detection in Remote Sensing. *Remote Sens.* **2023**, *15*, 3027. https://doi.org/10.3390/rs15123027

Academic Editor: Arko Lucieer

Received: 27 April 2023
Revised: 1 June 2023
Accepted: 5 June 2023
Published: 9 June 2023

1. Introduction

The widespread use of object detection technology has increased with the advancement of deep learning technology in the military, national defense, transportation, industrial automation and other fields [1]. Small objects are poorly detected compared with objects of normal scale due to their low resolution, small coverage in the image, and inadequate feature expression. Therefore, the question of how to accurately recognize and locate small objects is a significant topic in the current computer vision field. The criteria for defining small objects in different scenarios vary, but there are two basic types of current defining methodologies: (1) definition based on absolute size, where objects of a size less than 32 pixels × 32 pixels are defined as small objects in the MS COCO dataset; and (2) definition based on relative size, where an object's imaging points are less than 80 pixels in a 256 pixels × 256 pixels image (i.e., a small object is defined as an object whose ratio of the number of pixels occupied by the object to the total number of pixels in the original image is less than 0.12%).

Existing convolutional neural network (CNN)-based [2] object detection methods can be divided into two categories, namely, two-stage object detection methods and one-stage object detection methods. With regard to the two-stage method, a region proposal network (RPN) and region of interest pooling (RoI pooling) are used first to obtain candidate regions. Then, classification and regression are performed on the candidate regions to obtain the final detection results, representing algorithms such as Fast R-CNN [3] and Faster R-CNN [4].

However, compared with the two-stage method, use of the one-stage object detection methods offers a more end-to-end strategy. The YOLO series is widely recognized as one of the most popular detection frameworks in industrial applications. This success is largely due to its exceptional balance between speed and accuracy. The series' pioneering works, YOLOv1–3 [5–7], have paved the way for one-stage detectors and been substantially improved over time. YOLOv4 [8] has been a game-changer in this regard, reorganizing the detection framework into separate parts (backbone, neck, and head) and introducing the bag-of-freebies and bag-of-specials to design a single-GPU training framework. Today, efficient detectors such as YOLOv5 [9], YOLOX [10], PPYOLOE [11], and YOLOv7 [12] are credible alternatives for deployment. Different-sized models are often obtained through scaling techniques. Aiming at the problem of large object scale gap, SSD [13] improves detection accuracy by setting multiple scales of feature maps for detection without losing real-time performance. Single-stage object detection models employ VGG-16 as their backbone network for feature extraction. To improve detection accuracy, several convolutional layers are added on top of the backbone network to produce a diverse set of feature maps. SSDs generate six of these, making them ideal for detecting smaller objects, while the deep feature maps contain more semantic information, making them suitable for detecting larger objects. For each input image, SSD utilizes the anchor concept introduced in Faster R-CNN by setting prior boxes with different scales and aspect ratios. These prior boxes serve as a reference for predicting the bounding boxes, thus reducing the complexity of training. Finally, SSD applies non-maximum suppression to generate the ultimate detection results. However, the problem of small object detection still arises due to the lack of semantic information derived from the shallow features and spatial information of deep feature maps. Hence, several techniques have been proposed by researchers to enhance the semantic information of the feature maps. ESSD [14] uses transposed convolution to upsample the feature map once more and then skips to connect feature maps of the same size. This improves situations where there is a difficulty in classifying due to shallow feature maps. FFESSD [15] presents a lightweight feature fusion module that integrates several outputs of the backbone and passes them through the feature pyramid again. This yields new features with stronger semantic information. ZHAO applied transposed convolution, dilated convolution and deformable convolution to improve SSD for multi-scale feature fusion in terms of receptive field and adaptive object shape [16]. Similar to the above, RSSD [17] also uses transposed convolution. However, it fuses the outputs of the various stages of the backbone. In addition, sharing parameters between feature maps becomes feasible when the channels of the fused feature maps exhibit consistency. Thereafter, in MDSSD [18], the deeper feature maps are considered to possess richer semantic information, so the improved model retains more of the backbone. This allows it to make jump connections over a larger span and finally use elaborate feature fusion modules.

This paper presents SAFF-SSD, an enhanced small object detection algorithm that is based on SSD and incorporates a transformer model and a multi-scale feature fusion strategy. As shown in Figure 1, comparisons with another SSD-based object detectors show that our method achieves the best accuracy on Pascal VOC2007. The structure is illustrated in Figure 2. To begin, more development is required for the VGG-16 backbone of the original SSD network's feature extraction efficiency and capability. Taking into account the fact that Transformer [19] has good global self-focus modeling capabilities, in our work we seek to combine Transformer and convolution in the network's backbone. To this end, we propose the Local Lighted Transformer block (2L-Transformer) and couple it with the lightweight neural network EfficientNetV2-S as the new backbone network, in order to achieve a balance between accuracy and efficiency. Second, SSD only predicts on a single-scale, shallow feature map, which cannot effectively integrate positional information with semantic features. Hence, SSD often results in significant false and missed detection rates, particularly for small items and objects of varying sizes. In order to resolve this issue, CSP-PAN is incorporated into the SSD. This is a feature fusion approach that improves the accuracy and stability of detection by integrating feature maps of different sizes and resolutions to

offer additional contextual information. With CSP-PAN, the model may leverage input from several layers concurrently to capture spatial and semantic information about the object more accurately. This can aid the model in detecting objects with varying sizes and resolutions and prevent it from being affected by resolution and scale discrepancies, resulting in a decrease in false and missing detection rates. Finally, the NWD metric [20] is utilized to overcome the sensitivity of the position variation of the microscopic objects and dramatically improve the detection performance.

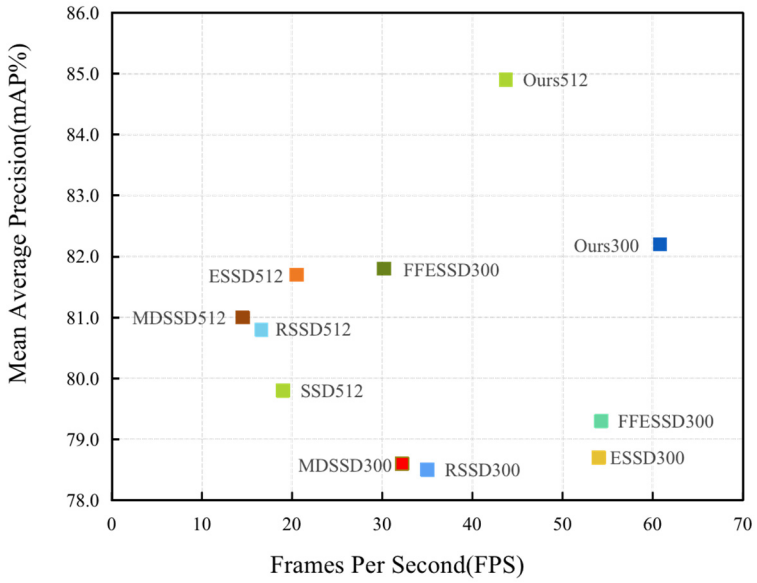

Figure 1. Comparison with another object detector based on SSD. Our method achieves optimal accuracy on Pascal VOC2007.

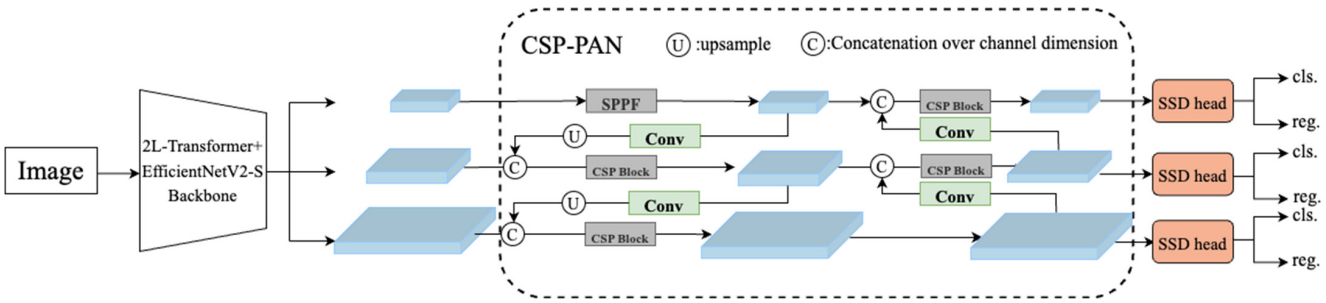

Figure 2. Architecture of our proposed method. This is a hybrid feature extraction architecture designed based on EfficientNet and Transformer, which cleverly uses the variant CSP module to improve the limitations of bottleneck network feature fusion. Finally, the prediction results are obtained using several convolutions as the head.

The contributions are summarized as follows:

1. We propose Local Lighted Transformer block (2L-Transformer), a new transformer module, and integrate it with EfficientNetV2-S as our backbone for improved feature extraction. On the AI-TOD dataset, an increase of 1.3% in *AP* is observed when we replace the original backbone with EfficientNetV2-S. When further embedding the proposed Transformer block in EfficientNetV2-S, we achieve an additional 1.8% increase in *AP*.
2. By optimizing the original bottleneck network on the PAN topology utilizing CSP as an internal building block, our experiments demonstrate that the optimized model is more effective in integrating multi-scale features.

3. We also evaluate the possibility of using the normalized Wasserstein distance (NWD) as an alternative similarity metric for small targets. Specifically, on the AI-TOD dataset, the proposed model with NWD has a 14.1% increase in *AP* compared with the standard *AP* baseline. Moreover, on the VOC 2007 dataset, our proposed multi-level NWD shows better results than the NWD metric.

The rest of the article is arranged as follows: some related works are reviewed in Section 2. Materials and methods are introduced in Section 3. Section 4 presents the results of our experiments and discusses their implications. Finally, Section 5 offers a concise summary of our findings and suggests directions for future research.

2. Related Work

2.1. Feature Pyramid

Pyramidal feature representation is a fundamental building block for solving multi-scale problems. SSD first tried to use pyramidal features for object detection. FPN [21] use lateral connections and top-down pathfinding to construct a feature pyramid with strong semantics. Further, PAN [22] adds bottom-up augmentation based on FPN to improve information flow. A bidirectional route is repeated several times by EfficientDet [23] to facilitate higher level feature fusion. In addition to designing the network manually, NAS-FPN [24] tries to find a stronger feature pyramid structure with the aid of a neural architectural search.

2.2. Transformers/Self-Attention Mechanism

Transformer is a self-attention-based neural network architecture proposed by Vaswani et al. [25] in 2017, where the self-attention enables Transformer to weigh the importance of various input sequence segments when creating the output. Nowadays, several versions of Transformer models are available, such as the BERT model for pre-trained language representations [26] and the GPT model for language production [27–29]. These have emerged as highly effective models in the field of natural language processing, outperforming previous models in various tasks, including machine translation, language modeling, and text classification. This has further brought Transformer into the focus of research. The Transformer has become one of the fundamental components of contemporary deep learning models. Self-attention layers have been utilized to supplement backbones [30–34] and head networks [35,36], enabling them to capture distant dependencies or diverse interactions. Moreover, the Transformer's encoder–decoder architecture has recently been employed for object detection and instance segmentation tasks [37–40].

2.3. Similarity Metrics

The Intersection over Union metric is commonly employed for measuring the similarity between bounding boxes. However, it has limitations in cases where the boxes do not overlap. To address this limitation Generalized *IoU* (*GIoU*) [41], which penalizes the smallest box converting bounding boxes, is proposed as a solution to this issue. The degradation from *GIoU* to *IoU* occurs in cases where one bounding box completely encloses another.

Consequently, it is suggested that *DIoU* [42] and *CIoU* [42] be used in order to overcome the constraints of *IoU* and *GIoU*. Yang et al. [43] also proposed a Gaussian Wasserstein distance (GWD) loss for oriented object detection by measuring the positional relationship of oriented bounding boxes. The goal of GWD is to address the oriented object identification border discontinuities and square-like difficulty. Moreover, Wang et al. have introduced the normalized Wasserstein distance (NWD) [20] as a solution to the sensitivity issue of Intersection over Union (*IoU*) in measuring the accuracy of object detection algorithms for small objects.

3. Materials and Methods

This section will introduce the proposed self-attention combined feature fusion-based SSD (SAFF-SSD) in detail. Figure 2 displays the SAFF-SSD architectural plan. Sections 3.1 and 3.2 introduce, respectively, the structures of the two essential constituents. The similarity metric is described in Section 3.3.

3.1. Self-Attention Mechanism

Transformer was initially introduced to address natural language processing (NLP) problems, but it has also been applied to the computer vision (CV) domain with good results on many tasks. In order to use Transformer with CV tasks, the self-attention mechanism of Transformer must first allow for interactions between different locations of the image, enabling the model to capture dependencies between different regions. Secondly, the multi-headed self-attention mechanism in Transformer can also process features from multiple channels simultaneously and combine them into a global representation. However, spatial structure information cannot be handled by the traditional Transformer. Even more seriously, image input will raise computational efficiency issues compared with text input. Therefore, applying Transformer models in the field of object detection remains a challenge.

In view of this, we aim to address this challenge by proposing a novel approach that combines the Transformer and convolution in the backbone network. This approach can improve computational efficiency while also leveraging the global self-attention modeling ability of Transformer. We use EfficientNetV2-S [44] as the base framework for the backbone. This is a lightweight and high performance convolutional neural network designed to solve the conflict between efficiency and accuracy and uses a module called "Squeeze-and-Excitation (SE)" [45]. Although it may improve detection accuracy, it is only an abbreviated enhancement or suppression of channels, lacks attention to spatial position, and is not stable across tasks or datasets. Therefore, SE is not real self-attention. Our proposed 2L-Transformer module truly combines the self-attention of the transformer with the spatial invariance of the convolution, and it has worked well in many subsequent experiments.

Let us first review the method of ViT [19]. This divides the image into 16 × 16 patches and each part is embedded into a vector. It then extracts features by standard transformer, which is the stacking of several multiheaded self-attention (MSA) modules, a multilayer perceptron module (MLP) and a layer norm (LN). We consider the module stacked L times in ViT as a block, named Transformer. The process can be computed as

$$\begin{gathered} z_0 = \left[x_{class};\, x_p^1 E;\, x_p^1 E;\, \ldots\, ;\, x_p^1 E\right] + E_{pos}, E \in R^{(P^2 \cdot C) \times D}, E_{pos} \in R^{(N+1) \times D} \\ z'_l = MSA(LN(z_{l-1})) + z_{l-1}, l = 1 \ldots L \\ z_l = MLP(LN(z'_l)) + z'_l, l = 1 \ldots L \\ Y = LN(z_L^0) \end{gathered} \tag{1}$$

However, this method produces a quadratic multiplication of computational expenses as the image resolution increases, and the constrained image resolution can have a negative effect on small object detection. Therefore, we present an improved module.

The Local Lighted Transformer block (2L-Transformer) is shown in Figure 3. To start with, we split the feature map into two parts according to the channels. Subsequently, one part does window local self-attention [46], which is to extract features within the window. For the other part, embedding is first implemented using dilation convolution, followed by global self-attention to interoperate information between windows. Next, we reshape the output of both branches to the original size, where the globally attended branch is downsampled and requires additional upsampling. Finally, we concatenate the output of the two branches according to the channel. At the same time, to preserve spatial location information and ensure valid gradients, we also design a shortcut from the input to the output [47]. After that, smoothing is performed using standard convolutional layers (Conv + BN + SiLu). Our module can be computed as

$$
\begin{aligned}
x_{loc},\ x_{glo} &= \mathrm{split}(x),\ W \times H \times 2C \rightarrow W \times H \times C,\ W \times H \times C \\
z_{glo} &= \left[x^1_{glo,p}E;\ \ldots\ ;\ x^1_{glo,p}E\right] + E_{glo,pos}, E \in R^{(P^2 \cdot C) \times D}, E_{pos} \in R^{N \times D} \\
Y_{glo} &= \mathrm{unreshape}\left(\mathrm{LN}\left(\mathrm{Transformer}\left(z_{glo}\right)\right)\right), \\
z_{loc} &= \mathrm{reshapeToWindow}(\mathrm{LN}(x_{loc})), W \times H \times D \rightarrow (p_w \cdot p_h) \times (win_w \cdot win_h) \times D \\
Y_{loc} &= \mathrm{unreshape}\left(\mathrm{LN}\left(\mathrm{Transformer}\left(z_{glo}\right)\right)\right), \\
Y &= \mathrm{Conv}\left(\mathrm{concat}\left[Y_{loc},\ Y_{glo},\ x_{loc},\ x_{glo}\right]\right)
\end{aligned}
\tag{2}
$$

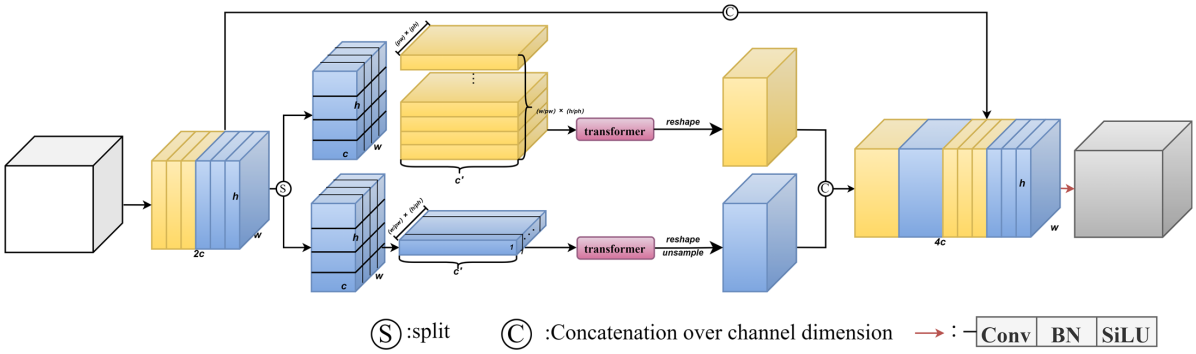

Figure 3. Architecture of our proposed Local Lighted Transformer block (2L-Transformer).

3.2. Feature Fusion Strategy

3.2.1. Spatial Pyramid Pooling-Fast

After the forward process consisting of standard convolution, each point on the feature map corresponds to a certain fixed small region of the original image. On scenes as complex as we can foresee—where small objects appear simultaneously with large ones—this fixed correspondence limits the model's capacity for detection. Therefore, in our work, inspired by the Spatial Pyramid Pooling (SPP) module proposed in *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition* by K. He [48], we add the Spatial Pyramid Pooling-Fast block after the backbone. As illustrated in Figure 4, we integrate local features by concatenating the maximum pooling outputs of different kernel sizes. Furthermore, doing so will enrich the expressiveness of the feature map. It is effective when there are large differences in object sizes, and it does not require additional parameters that need training. It is worth mentioning that stacking pooling layers with small kernel sizes gives exactly the same results and significantly reduces the computational cost when compared with the direct use of pooling layers with large kernel sizes.

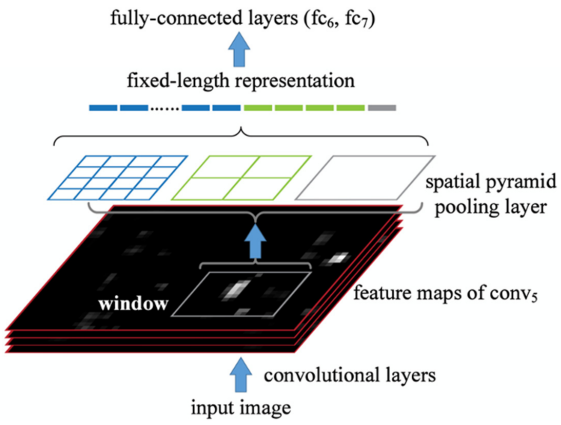

Figure 4. SPP-net for object detection [48]. Pooling features from arbitrary windows on feature maps. The feature maps are computed from the entire image, and the pooling is performed in candidate windows.

3.2.2. Path Aggregation Network

Figure 5a shows the detailed structure of the PANet proposed by Liu et al. To improve performance, it conducts path augmentation and aggregation. Specifically, the position features present in the shallow layers are gradually blurred or even lost as the feature extraction goes deeper. Considering this, bottom-up path augmentation is added, which will allow the combination of positional features at the shallow layers with semantic features at the deeper layers. PAN is an extension of FPN, which defines layers producing feature maps with the same spatial size as belonging to the same network stage, with each stage corresponding to a feature level. And PAN uses a set notation to denote the feature levels generated by FPN, specifically $\{P_1, P_2, P_3\}$. The enhancement path starts from the lowest level P_1 and progresses towards P_3, as illustrated in Figure 5a. From P_1 to P_3, the spatial size is progressively downsampled by a factor of 2. We use $\{N_1, N_2, N_3\}$ to denote the newly generated feature maps that correspond to $\{P_1, P_2, P_3\}$. Note that N_1 is just P_1, without any modification. Each building block acquires a higher resolution feature map Ni and a coarser feature map P_{i+1} through lateral concatenation and generates a new feature map N_{i+1}. Each feature map N_i is first passed through a 3×3 convolutional layer with step size 2 to reduce the space size. Then, each element of the feature map P_{i+1} and the downsampled map are joined laterally by summation. After obtaining the fused feature maps, a 3×3 convolutional layer is applied to generate the next feature map, N_{i+1}. This is an iterative process that terminates after approaching P_3. A rectified linear unit (ReLU) activation function follows each convolutional layer. Subsequently, the object is then detected on $\{N_1, N_2, N_3\}$.

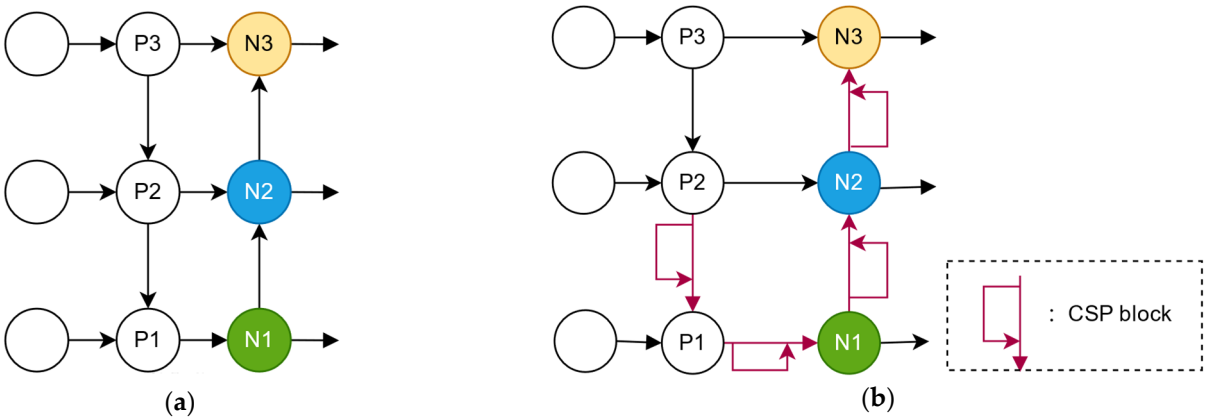

Figure 5. Illustrations of (**a**) PAN and (**b**) cross stage partial PAN (CSP-PAN), showing the flow of feature maps. The leftmost three circles represent the extracted features from the backbone. Then, starting from the last feature, the map flows from top to bottom (upsampling + fusion), and then from bottom to top (downsampling + fusion).

3.2.3. Cross Stage Partial-PAN

While adding the PAN module as the neck of the model (which is very lightweight compared with the backbone) does work, doing so can cause problems when we consider improving it. Generally, to improve the performance, increasing the depth and width of the network is considered the most straightforward approach. However, if we do this, issues such as parameter bloat, overfitting, reduced gradient stability, and difficulty in convergence will follow. One paper, which goes deeper with convolutions [49], reports the assembly of multiple convolution or pooling operations into modules and the building of a network. The networks designed in this way reached a much higher level at that time. Inspired by this, we expanded the standard convolutional layer into multiple branches in the neck part of the network. However, unlike inception [50], we did not use a 1×1 convolution to reduce the number of channels. Instead, we directly split the input into two

parts by channel. We have found that this has a minimal effect on the results while saving a considerable number of parameters.

The architecture of CSP-PAN is shown in Figure 5b. The modified CSP-PAN embeds the CSP-block [50], which has the structure shown in Figure 6a. In one CSP-block, the feature map in a stage is split into two parts through channel $X_0 = [X'_0, X''_0]$. Between X'_0 and X''_0, the former goes through one ConvBNSiLu layer, and is then directly connected to the end of the stage, and the latter goes through one ConvBNSiLu layer and some bottlenecks. The output of these parts will be concatenated and passed through another ConvBNSiLu layer, until it finally generates output X_U. Unlike the original CSP, we have found that concatenating each bottleneck layer to the output results in better performance.

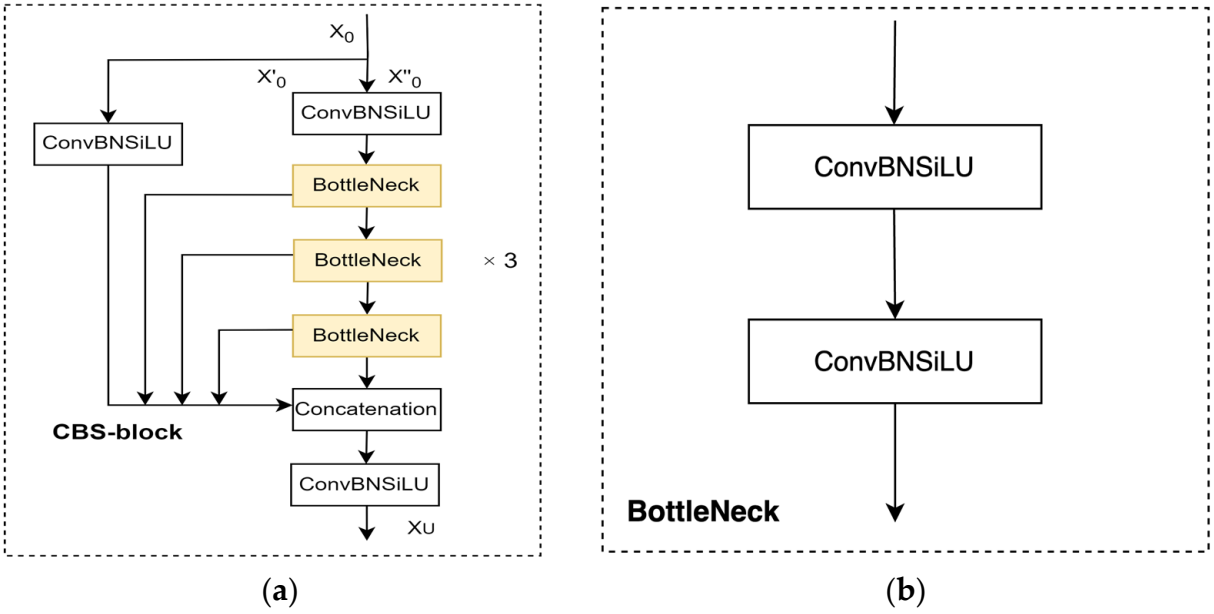

Figure 6. Illustrations of (**a**) CSP-block and (**b**) bottleneck. Unlike the original CSP structure, each output of the bottleneck network on the main branch in the variant CSP structure will be concatenated together.

3.3. Similarity Metric

IoU is actually the Jaccard similarity coefficient for computing similarity of two limited sample sets. According to this fact, in [20], Wang proposed a novel metric for small objects using Wasserstein distance, which can better reflect the distance between distributions even if they have no overlap. Compared with *IoU*, this metric has better properties for measuring the similarity between small objects. Normalized Wasserstein distance is a distance metric that utilizes the Wasserstein distance, derived from Optimal Transport theory, to measure the distance between probability distributions. When comparing two 2D Gaussian distributions denoted by $\mu_1 = \mathcal{N}(m_1, \Sigma_1)$ and $\mu_2 = \mathcal{N}(m_2, \Sigma_2)$, the second order Wasserstein distance can be used to quantify their distance from one another as follows:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \mathrm{Tr}(\Sigma_1 + \Sigma_2 - 2(\Sigma_2^{1/2}\Sigma_1\Sigma_2^{1/2})^{1/2}) \tag{3}$$

and it can be simplified as:

$$W_2^2(\mu_1, \mu_2) = \|m_1 - m_2\|_2^2 + \left\|\Sigma_1^{\frac{1}{2}} - \Sigma_2^{\frac{1}{2}}\right\|_F^2, \tag{4}$$

where $\|\cdot\|_F$ is the Frobenius norm.

Furthermore, for Gaussian distributions $\mathcal{N}_a$ and $\mathcal{N}_b$, which are modeled from bounding boxes $A = (cx_a, cy_a, w_a, h_a)$ and, $B = (cx_b, cy_b, w_b, h_b)$, respectively, Equation (4) can be simplified as:

$$W_2^2(\mu_1, \mu_2) = \left\| \left(\left[cx_a, cy_a, \frac{w_a}{2}, \frac{h_a}{2} \right]^T \right), \left[cx_b, cy_b, \frac{w_b}{2}, \frac{h_b}{2} \right]^T \right\|_2^2 \quad (5)$$

An exponential form normalization is applied to Wasserstein distance, resulting in a new metric known as normalized Wasserstein distance (NWD):

$$NWD(\mathcal{N}_a, \mathcal{N}_b) = \exp\left(-\frac{\sqrt{W_2^2(\mathcal{N}_a, \mathcal{N}_b)}}{C} \right), \quad (6)$$

where C is a constant that has a strong connection to the dataset.

In addition, the NWD metric has a single hyperparameter C that is strongly dependent on the dataset, making it difficult for model training in multi-scale object detection tasks. To address this issue, we expanded the normalized Wasserstein distance (NWD) to multi-level NWD. The process of setting the hyperparameters for the multi-level NWD is shown in Figure 7. First, we categorized objects in the training set into three groups based on their length, corresponding to the three detection heads in object detection. Next, we computed the NWD metric for various scales based on the origin of the predicted bounding boxes. This direct approach has been effective in bringing the NWD performance in multi-scale object detection tasks closer to that of *IoU*.

Figure 7. The process of setting the hyperparameters for the multi-level NWD. In the clustering process, we cluster objects based on their length and width in the training set and use the average length of objects within the same cluster as the hyperparameter. For example, in the image, cluster C_1 corresponds to larger objects.

4. Results

4.1. DataSet

Three publicly available object-detection datasets are used to evaluate the proposed methods in the experiments. Some examples, the Pascal VOC 2007 [51], TGRS-HRRSD [52], and AI-DOT [53], are shown in Figure 8.

The first dataset is Pascal VOC 2007 which is a benchmark dataset for object detection and has been extensively used for evaluating object detection models. It was introduced

as part of the PASCAL Visual Object Classes Challenge in 2007 and has since become a standard benchmark for evaluating the performance of computer vision algorithms. The dataset consists of over 5000 images, each of which is annotated with object class labels and bounding box coordinates for up to 20 object categories, including people, cars, and animals.

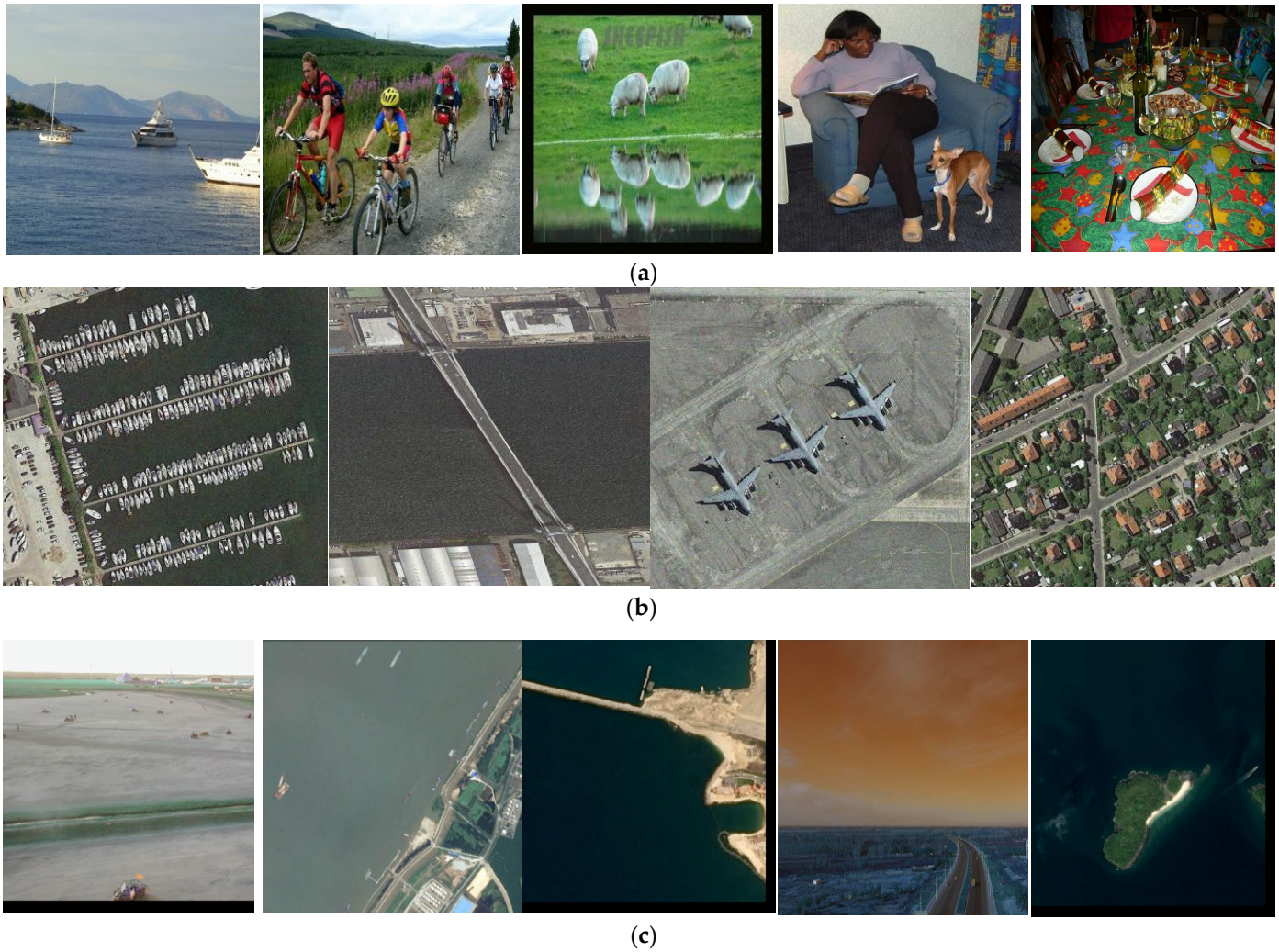

Figure 8. The figure showcases example images from three experimental datasets, with examples from (**a**) Pascal VOC 2007, (**b**) TGRS-HRRSD, and (**c**) AI-TOD displayed in the top, middle, and bottom rows, respectively.

The second dataset is TGRS-HRRSD, which was published by the University of Chinese Academy of Sciences in 2019. The spatial resolution of the 21,761 image samples in the TGRS-HRRSD, which ranges from 0.15 m to 1.2 m, were collected from Google Earth and Baidu Map. There are 55,740 object instances total, divided across 13 object categories. The TGRS-HRRSD dataset contains occlusion levels and truncation rates, which are key factors in evaluating the performance of small object detection models and makes it a more accurate measure of small object detection model performance in challenging real-world scenarios.

The third dataset is AI-TOD. This is a recent addition to the family of object detection datasets, with a focus on small objects. The dataset comprises 4280 images with an average absolute size of instances of only 12.8 pixels, which is much smaller than other datasets such as PASCAL VOC (156.6 pixels), MS COCO (99.5 pixels) [54], or DOTA (55.3 pixels) [55], and which makes it one of the smallest object detection datasets. The dataset presents a challenging task for object detection models due to the extremely small object size and limited number of pixels

available for each object instance. Therefore, it serves as a valuable benchmark for evaluating the performance of object detection models when detecting small objects.

4.2. Experimental Setup

To compare the performance of the proposed method, we evaluate it against several algorithms. Among these, SSD [13], YOLOv3 [7], YOLOv4 [8], RetinaNet [56], YOLOv6-M [57] are one-stage methods. In addition, the FoveaBox [58] and YOLOv8n [59] are anchor-free methods. Some algorithms of remote sensing are EFNet [60] and HRCNN-regression [52]. Due to the unavailability of public codes, the main sources for various experimental results of domain algorithms are already published references.

In the object detection experiments, we use the following default hyper-parameters: the optimizer is SGD and a step decay learning rate scheduling strategy is used with an initial learning rate of 0.01, multiplied by a factor of 0.1 at 50 and 100 epochs. The architecture uses a single GPU to perform training with a batch size of 8 due to GPU memory limitations. The momentum is set to 0.95 with the weight decay set to 0.0005. All experiments are run with an NVIDIA RTX 3080ti GPU.

4.3. Evaluation Metric

The average precision (*AP*) and its derivative metrics are adopted to quantitatively evaluate the proposed method. The *AP* is a comprehensive metric in the task of object detection and is based on the precision and recall of Equations (7) and (8).

$$precision = \frac{TP}{TP + FP}, \tag{7}$$

$$recall = \frac{TP}{TP + FN}, \tag{8}$$

where the terms *TP*, *FP*, and *FN* are true positives, false positives, and false negatives, respectively. The terms *TP*, *FP*, and *FN* are calculated from the Intersection over Union (*IoU*) between the bounding boxes of ground-truth and the bounding boxes of prediction as follows:

$$IoU = \frac{B_{pred} \cap B_{gt}}{B_{pred} \cup B_{gt}}, \tag{9}$$

where B_{pred} denotes the bounding box of prediction, and B_{gt} is the bounding box of ground-truth. The standard HRRSD metrics include AP, $AP_{0.5}$, $AP_{0.75}$, AP_s, AP_m, and AP_l. AP denote AP at IoU = 0.50:0.05:0.95 (average over IoU thresholds), $AP_{0.5}$ denotes AP at IoU = 0.50 (equally to PASCAL VOC metric). $AP_{0.75}$ denotes AP at IoU = 0.75 (very strict metric). AP_s denotes AP for small objects whose areas are smaller than 32^2. AP_m denotes AP for small objects whose areas are between 32^2 and 96^2. AP_l denotes AP for large objects whose areas are bigger than 96^2. In addition, AP_{vt}, AP_t, AP_s, AP_m are APs for very tiny, tiny, and small medium scales, respectively. For the detection of multi-category objects, the AP usually denotes mean average precision (mAP) which is obtained by the average of different category APs.

4.4. Results and Analysis

The details of the experimental findings are presented in this section. The quantitative results of the comparative methods are shown in Tables 1–3, and qualitative results are illustrated in Figures 9–11. We apply *mAP* to be evaluation metric, which is a widely used standard in the field of object detection. Therefore, it allows direct comparison of the performance of different models or mechanisms.

Table 1. Results of comparative approaches for object detection on Pascal VOC 2007. In natural scenes, our algorithm yields performances that are almost on par with the state-of-the-art one-stage algorithm yolov6. Nevertheless, our approach still boasts highly satisfactory results in this context.

Method	Backbone	Input Size	*mAP*
SSD300	VGG-16	300 × 300	77.2
SSD512	VGG-16	512 × 512	79.8
ESSD [14]	VGG-16	300 × 300	78.7
ESSD [14]	VGG-16	512 × 512	81.7
RSSD [17]	VGG-16	300 × 300	78.5
RSSD [17]	VGG-16	512 × 512	80.8
FFESSD [15]	VGG-16	300 × 300	79.1
FFESSD [15]	VGG-16	512 × 512	80.9
MDSSD [18]	VGG-16	300 × 300	78.6
MDSSD [18]	VGG-16	512 × 512	81.0
YOLOv6-M [57]	EfficientRep	640 × 640	86.0
YOLOv8n [59]	Darknet	640 × 640	80.2
DETR [39]	Transformer Encoder	640 × 640	82.6
Ours300	VGG-16	300 × 300	79.9
Ours512	VGG-16	512 × 512	82.4
Ours300	2L-Transformer+EfficientNetV2-S	300 × 300	82.2
Ours512	2L-Transformer+EfficientNetV2-S	512 × 512	84.9

Table 2. Results of comparative approaches for object detection on TGRS-HRRSD. Although many algorithms can detect most of the targets in TGRS-HRRSD based on $AP_{0.5}$, our approach stands out by achieving superior overall performance and demonstrating particular strengths in detecting small targets (AP_S).

Method	Backbone	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_s	AP_m	AP_l
HRCNN-regression [52]	AlexNet	/	51.4	/	/	/	/
YOLO-v4	CSPDarknet-53	59.9	89.7	68.3	11.2	50.1	58.2
YOLOv3-608	Darknet-53	59.4	89.0	66.8	10.6	48.3	55.6
RetinaNet	ResNet-50	59.6	89.3	67.7	10.9	49.2	57.4
EFNet [60]	ResNet-50	62.2	90.7	71.7	12.4	53.5	60.2
SSD512	VGG-16	52.7	87.3	57.4	9.5	42.1	51.7
FoveaBox	Resnet-50	61.8	90.4	70.6	18.4	51.8	58.7
SSD512	EfficientNetV2-S	53.4	88.0	62.1	9.6	43.5	53.8
YOLOv6-M [57]	EfficientRep	62.9	91.4	71.2	18.3	54.0	61.2
YOLOv8n [59]	Darknet	61.3	90.4	70.7	18.0	53.1	59.3
DETR [39]	Transformer Encoder	58.8	88.5	69.5	14.5	53.5	61.8
Ours512	2L-Transformer+ EfficientNetV2-S	63.2	91.1	72.2	20.4	53.8	60.6

Table 3. Results of comparative approaches for object detection on AI-TOD. Unlike object detection in daily life, detecting objects in aerial images is more challenging and so fully demonstrates the advantages of our approach.

Method	Backbone	*AP*	$AP_{0.5}$	$AP_{0.75}$	AP_{vt}	AP_t	AP_s	AP_m
SSD512 [13]	ResNet-50	7.0	21.7	2.8	1.0	4.7	11.5	13.5
TridentNet [61]	ResNet-50	7.5	20.9	3.6	1.0	5.8	12.6	14.0
FoveaBox [58]	ResNet-50	8.1	19.8	5.1	0.9	5.8	13.4	15.9
RepPoints [62]	ResNet-50	9.2	23.6	5.3	2.5	9.2	12.9	14.4
FCOS [63]	ResNet-50	9.8	24.1	5.9	1.4	8.0	15.1	17.4
M-CenterNet [53]	DLA-34	14.5	40.7	6.4	6.1	15.0	19.4	20.4
RetinaNet	ResNet-50	4.7	13.6	2.1	2.0	5.4	6.3	7.6
ATSS [64]	ResNet-50	12.8	30.6	8.5	1.9	11.6	19.5	29.2
Faster R-CNN	ResNet-50	11.1	26.3	7.6	0.0	7.2	23.3	33.6
Cascade R-CNN [65]	ResNet-50	13.8	30.8	10.5	0.0	10.6	25.5	36.6
DetectorRS [66]	ResNet-50	14.8	32.8	11.4	0.0	10.8	28.3	38.0
SSD512	EfficientNetV2-S	6.9	21.4	2.7	0.8	4.7	11.0	13.5
YOLOv6-M [57]	EfficientRep	16.3	37.9	12.1	2.6	10.2	30.5	40.2
YOLOv8n [59]	Darknet	14.9	32.8	11.6	2.0	9.8	26.4	33.6
DETR [39]	Transformer Encoder	10.6	26.4	7.4	0.0	6.7	20.5	35.2
Ours	2L-Transformer+ EfficientNetV2-S	21.1	49.9	14.9	7.0	20.8	30.1	38.8

Figure 9. Qualitative examples of small object scene detection on Pascal VOC 2007. (**a**) Qualitative example of image 1. (**b**) Qualitative example of image 2. (**c**) Qualitative example of image 3. (**d**) Qualitative example of image 4. Each row, from left to right, represents the image that needs to be detected with labels, the SSD algorithm's detection result, and our proposed algorithm's detection result, respectively.

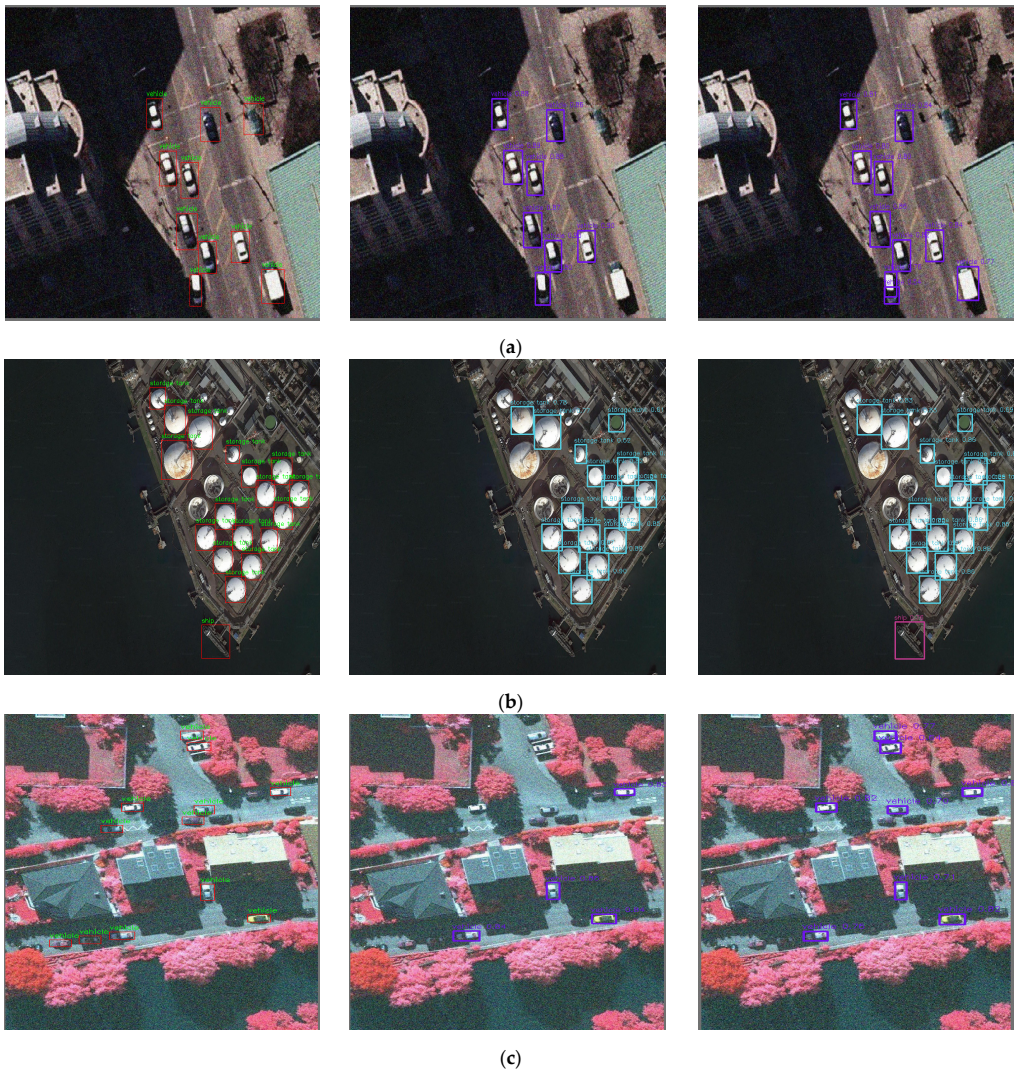

Figure 10. Qualitative examples of small object scene detection on TGRS-HRRSD. (**a**) Qualitative example of image 1. (**b**) Qualitative example of image 2. (**c**) Qualitative example of image 3. Each row, from left to right, represents the image that needs to be detected with labels, the SSD algorithm's detection result, and our proposed algorithm's detection result, respectively.

For the Pascal VOC 2007, based on Table 1, it can be observed that when the input size is 512 × 512, our method outperforms the classical SSD algorithm (baseline) with 5.1% improvement in *mAP*, and also outperforms other improved SSD methods such as FSSD, RSSD, DSSD, and MDSSD. Although we design SAFF-SSD for small object detection, it still achieves good results in VOC, outperforming YOLOv8n [59] and second only to YOLOv6-M [57]. This proves that the proposed SAFF-SSD is a promising method.

For the TGRS-HRRSD, it can be observed, based on Table 2, that our method achieves the best results on the TGRS-HRRSD dataset, outperforming methods such as RetinaNet [56] and YOLOv4 [8], and, with regard to *AP*s, beating the second-best, FoveaBox [58], by 2.0%, showing the advantage for small object detection.

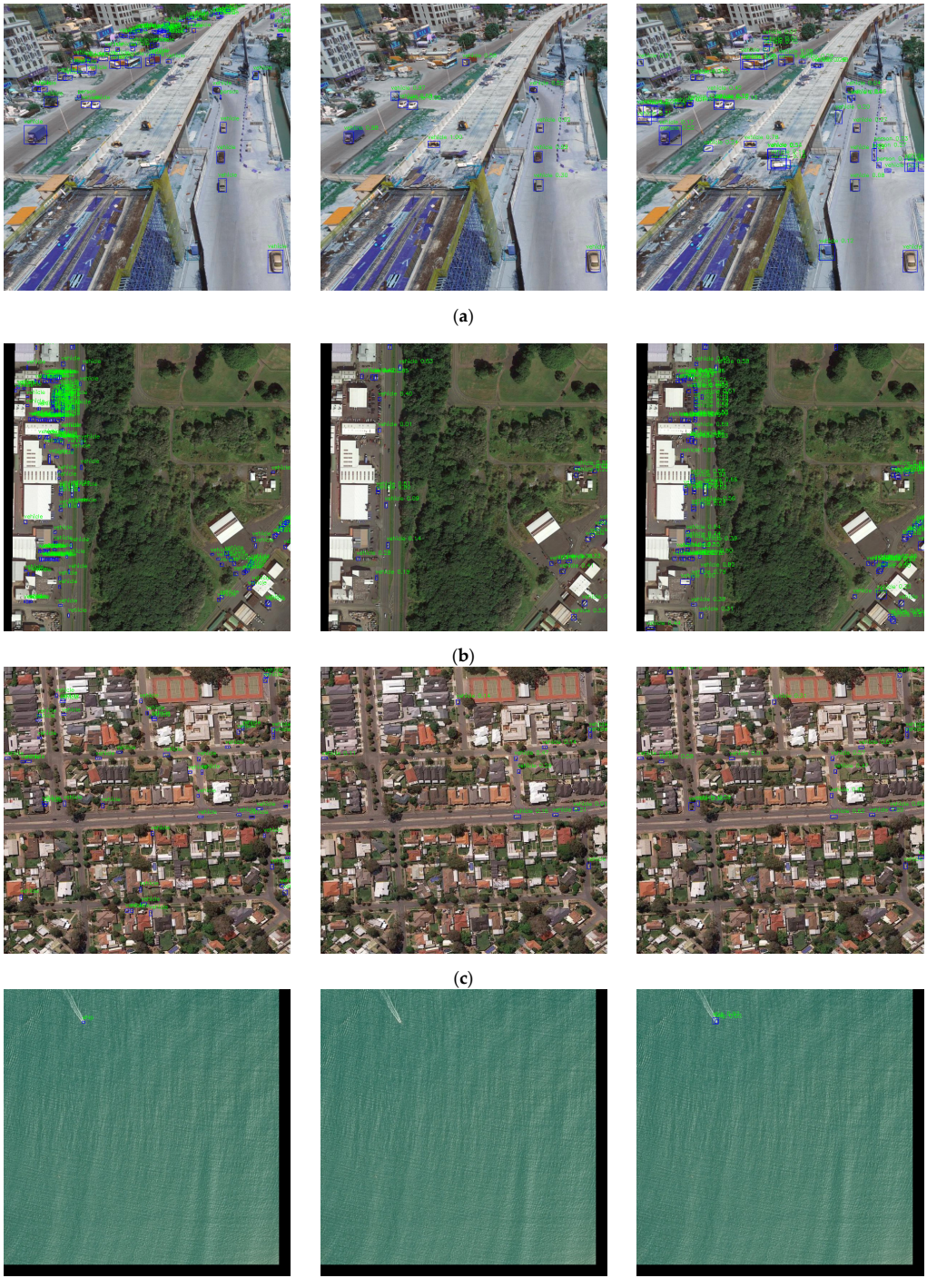

Figure 11. Qualitative examples of small object scene detection on AI-TOD. (**a**) Qualitative example of image 1. (**b**) Qualitative example of image 2. (**c**) Qualitative example of image 3. (**d**) Qualitative example of image 4. Each row, from left to right, represents the image that needs to be detected with labels, the SSD algorithm's detection result, and our proposed algorithm's detection result, respectively.

For the AI-TOD, and based on Table 3, it becomes apparent that AP_{vt} of current state-of-the-art detectors tends towards zero, which indicates that they are not suitable for producing satisfactory results on tiny objects. In addition, our proposed detectors surpass the AP_t metric of SSD [13], TridentNet [61], FoveaBox [58], RepPoints [62], FCOS [63] and M-CenterNet [53] by 16. 1%, 15.0%, 15. 0%, 11.6% and 12.8%, respectively. Notably, our

proposed method achieves state-of-the-art performance (21.1% *AP*) on the AI-TOD dataset among one-stage detectors, particularly in cases where objects are extremely small. This indicates a significant improvement in performance compared with existing models.

Figures 9–11 show visual comparison of the detection performance of the SSD algorithm and the proposed algorithm. Each row, from top to bottom, represents the image that needs to be detected with labels, the SSD algorithm's detection result, and our proposed algorithm's detection result, respectively. From these figures, it can be seen that SSD is very prone to miss detection in images with complex scenes and dense objects. The proposed algorithm uses a transformer block to enhance the features and focus more on the spatial location information of small objects, and combines feature fusion to strengthen the semantic information of the feature map, thus effectively locating and detecting dense small object objects and reducing the phenomenon of missed detection, thus proving the superiority of the proposed algorithm in detecting dense small objects in a high-resolution image.

4.5. Ablation Study

4.5.1. Influence of Local Lighted Transformer Module

To demonstrate the efficacy of the proposed algorithm, we have chosen to perform ablation experiments on several datasets, such as AI-TOD. This also helps us to further analyze the degree of impact of each module on the detection performance. Among these datasets, AI-TOD is designed for evaluating the performance of the challenging task of small object detection. The average absolute size of the instances in this dataset is only 12.8 pixels, which makes it appropriate for performance testing of small object detectors.

We propose an efficient, lightweight, local self-attention 2L-Transformer module and have replaced some of the modules in the backbone network EfficientNetV2-S. To provide a more intuitive and clear analysis of the effect of the proposed attention module, a heat map is generated and visualized for a scene with small objects, when the input image size is set to 512 $\times$ 512. The 2L-Transformer module's detecting impact is shown in Figure 12. It is evident that the area of interest is more accurately objected to the small object, which effectively avoids the missed detection and improves the detection accuracy of small objects. Table 4 presents the results of the ablation experiment, indicating a significant improvement in the feature extraction ability of EfficientNetV2-S with the proposed attention module. First, we use EfficientNetV2-S as the backbone of SSD, and its detection accuracy gains 1.3% *mAP* improvement. It can be seen that using an improved design for the backbone does work. We then proceed to replace some of the modules in the EfficientNetV2-S with 2L-Transformer and end up with a nearly 3.1% improvement in detection accuracy compared with the original SSD. This is due to the good performance of the 2L-Transformer module in global modeling and context capture. To a certain extent, it addresses the limitations of the receptive field and compensates for the lack of features specific to small objects. However, EfficientNet is considered a lightweight backbone, and increasing the number of modules does not guarantee their superiority. Therefore, in the end, we replaced the backbone with the most commonly used ResNet50, and the experiment proved that the backbone with 2L-Transformer had better feature extraction ability than ResNet50, resulting in an improvement of nearly 1% in the final results. Moreover, the parameter counts of the backbone networks are comparable and all within 30 M.

Figure 12. Local Lighted Transformer block visualization results. (**a**) Heat map of image 1. (**b**) Heat map of image 2. (**c**) Heat map of image 3. The heatmap focuses on the regions with targets. However, it is worth noting that, while almost all targets are marked, the most intense heat is not always near the large objects in the image, but rather on small objects or strong feature areas of the large objects, such as the legs of a sheep.

Table 4. Effectiveness of Local Lighted Transformer block visualization results. Several backbone feature extractors were compared, including VGG-16, which was featured in the original SSD, ResNet-50 which is now widely utilized, the native version of EfficientNet, and the final mixed architecture version.

	Backbone	*AP*	$AP_{0.5}$
AI-TOD	VGG-16	7.0	21.7
	ResNet-50	9.1	26.2
	EfficientNetV2-S	8.3	24.4
	2L-Transformer + EfficientNetV2-S	10.1	27.5
VOC-07	VGG-16	60.7	79.8
	ResNet-50	62.1	81.6
	EfficientNetV2-S	61.5	80.3
	2L-Transformer + EfficientNetV2-S	62.5	82.8
TGRS-HRRSD	VGG-16	52.7	87.3
	ResNet-50	58.8	88.9
	EfficientNetV2-S	58.7	88.4
	2L-Transformer + EfficientNetV2-S	59.6	89.3

4.5.2. Influence of Different Feature Fusion Modules

To further investigate the impact of the feature fusion module on small object detection, we completely trained three variants of the model on two datasets (AI-TOD and VOC-07). By comparing the effect of the models on the test set, we can analyze the role and applicability scenarios of CSP-PAN and SPP-Fast. As expected, the experiments prove that the modules we used are effective.

- Spatial Pyramid Pooling-Fast

SPP-Fast uses maximum pooling for the inputs with different kernel sizes, then concatenates by channel. There are two main starting points for this: firstly, this facilitates the processing of objects of different sizes; and secondly, we aim to improve the robustness and generalization of the model. Nevertheless, adding the module only after the backbone may lead to weak prediction accuracy. The reasons are clear. Firstly, the output of this module is a small-sized feature map for predicting large objects, and the module has a limited range of influence; meanwhile, pooling may result in the loss of certain crucial information, for example, leading in a reduction of spatial resolution or a lack of location information. Therefore, some improvements are still needed.

- Cross Stage Partial-PAN

To verify the superiority of the CSP-PAN feature fusion module, we ensured that other components were the same and selected different necks for feature fusion. The baseline was the native neck of the SSD model which extracted feature maps from different stages, passed them through several convolutional layers, and maintained their sizes. Some details can be found in Table 5. The experimental results showed that CSP-PAN consistently outperformed FPN. Even for the structurally similar PFPN, CSP-PAN still achieved the best performance on all three datasets, with an improvement of nearly 0.5 points.

Table 5. Effectiveness of different feature fusion modules. Through comparison, CSP-PAN structure is not overly complex yet still manages to achieve an almost 0.5-point improvement.

Dataset/Component	SSD_Neck ($AP_{0.5}$)	FPN ($AP_{0.5}$)	PAN ($AP_{0.5}$)	CSP-PAN ($AP_{0.5}$)
AI-TOD	27.5	29.8	31.8	32.4
VOC-07	82.8	83.5	84.3	84.9
TGRS-HRRSD	89.3	90.2	90.5	90.6

CSP-PAN combines the location information in the shallow layer with the semantic information in the deep layer, thus achieving feature fusion and ultimately improving the capability of feature representation. As is to be expected, with this cross-layer feature fusion, the model is more likely to make a correct classification and localization of small objects. Specifically, Table 6 shows that the proposed method achieves a boost of 1.3% *mAP* on the VOC test set and nearly 4.7% *mAP* on the AI-TOD test set.

Table 6. The contribution of SPPF in feature fusion. Although it is a plug-and-play multi-scale sampling component that can be used for feature fusion, it is not suitable for standalone usage.

Dataset/Component	SPP-Fast	CSP-PAN	$AP_{0.5}$
AI-TOD			27.5
	√		27.3
		√	32.2
	√	√	32.4
VOC-07			82.8
	√		82.4
		√	84.1
	√	√	84.9
TGRS-HRRSD			89.3
	√		89.4
		√	90.4
	√	√	90.6

As analyzed above, adding SPP-Fast after the backbone without consideration will have negative effects, contrary to our expectations, because SPP-Fast has a limited range of influence and may cause information loss. If we couple CSP-PAN with SPP-Fast, the effect will be completely different, not only will the impact range be extended, but the lost information may also be filled from the cross-layer connection. Experiments show that the detection effect is optimal in the model wherein CSP-PAN is combined with SPP-Fast: 2.1% *mAP* improvement on the VOC test set and 4.9% *mAP* improvement on the AI-TOD test set.

4.5.3. Influence of Normalized Wasserstein Distance

Table 7 displays the effectiveness of NWD metric, 7.1% *AP* improvement is achieved with the NWD-based positive and negative sample assignment compared with the baseline method. This indicates that the small object sensitivity to *IoU* severely affects the assignment of labels, and this confounding error has a negative impact on the training. However, the NWD-based assignment strategy we adopt greatly improves the assignment quality and when we apply NWD to both assigning and loss modules, the best performance of 49.9% $AP_{0.5}$ is achieved.

As observed from the aforementioned table, the NWD values for objects of larger proportions undergo a significant reduction in the VOC-07 dataset, leading to a decline of almost 6% in *mAP*. In order to maintain the admirable descriptive features of NWD for small-sized objects, while simultaneously confronting its limitations in recognizing objects of diverse scales, we employed the multi-level NWD methodology, as delineated in the ensuing Table 8. In instances where the prevalent size of the objects in the dataset was small, NWD yielded better outcomes in comparison with the traditional *IoU* technique. Nonetheless, for entities exhibiting a varied range of proportions, normalizing NWD may impede the regression of bounding boxes, leading to distortion and reduced precision that may ultimately impair the model's training. The said predicament can be circumvented successfully by employing the multi-level NWD technique, which facilitates closer alignment of the NWD yield to that of *IoU* in diverse scale contexts, minimizing the disparity to a mere 20% of its original value.

Table 7. Effectiveness of normalized Wasserstein distance. We modified the existing models as previously described and conducted comprehensive tests on multiple datasets to explore NWD's efficacy in terms of matching positive and negative samples, model training, and validation.

Dataset	Method	Assigning	Loss	*AP*	$AP_{0.5}$
AI-TOD	Baseline			12.8	32.4
	NMD	√		18.9	46.2
			√	13.4	36.4
		√	√	21.1	49.9
VOC-07	Baseline			64.8	84.9
	NMD	√		62.2	80.2
			√	60.7	78.8
		√	√	60.5	78.6
TGRS-HRRSD	Baseline			62.5	90.6
	NMD	√		62.4	90.4
			√	63.0	91.1
		√	√	62.4	90.5

Table 8. Comparison of NWD and multi-level NWD methods in VOC-07. The multi-level NWD effectively addresses the challenge of multi-scale target detection in NWD, yielding results comparable to those achieved with *IoU*.

Method	*AP*	$AP_{0.5}$
IoU	64.8	84.9
NWD	60.7	78.8
Multi-level NWD	62.6	84.1

5. Conclusions

In this paper, we have proposed SAFF-SSD: an improved small object detection algorithm, which aims to address the issue of low small object detection precision in SSD and which has an obvious advantage over the previous work and the currently presentative methods. One of the main contributions is that we propose Local Lighted Transformer block (2L-Transformer) as a novel self-attention module, which is embedded in the lightweight EfficientNetV2-S as our backbone. Experimentally, we have shown that the proposed 2L-Transformer increases the effectiveness of EfficientNetV2-S for superior feature extraction. In addition, our detector can leverage feature maps of different scales by using CSP-PAN to improve the accuracy and stability of detecting small objects. This structure can achieve better performance than even the most advanced neck networks. Finally, we used a large number of reliable experiments to verify the superiority of normalized Wasserstein distance (NWD) in small object detection and slightly extend it to improve the problem of handling multi-scale targets. Moving forward, our future research efforts will be directed towards enhancing the real-time performance of small object detection.

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

Funding: This research was supported by the National Natural Science Foundation of China (No. 62076137).

Data Availability Statement: All data included in this study are available upon request by contact with the corresponding author.

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

References

1. 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]
2. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-Based Learning Applied to Document Recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
3. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
5. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
6. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
7. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
9. Glenn, J. YOLOv5 Release v6.1. 2022. Available online: https://github.com/ultralytics/yolov5/releases/tag/v6.1 (accessed on 10 June 2020).
10. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430.
11. Xu, S.; Wang, X.; Lv, W.; Chang, Q.; Cui, C.; Deng, K.; Wang, G.; Dang, Q.; Wei, S.; Du, Y.; et al. Pp-yoloe: An evolved version of yolo. *arXiv* **2022**, arXiv:2203.16250.
12. 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.
13. Liu, W.; Anguelov, D.; Erhan, D. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 11–14 October 2016.
14. Leng, J.; Liu, Y. An enhanced SSD with feature fusion and visual reasoning for object detection. *Neural Comput. Appl.* **2019**, *31*, 6549–6558. [CrossRef]
15. Shi, W.; Bao, S.; Tan, D. FFESSD: An Accurate and Efficient Single-Shot Detector for Target Detection. *Appl. Sci.* **2019**, *9*, 4276. [CrossRef]
16. Zhao, Q.; Zhou, D.; Zhai, J. SSD small target detection algorithm based on deconvolution and feature fusion. *CAAI Trans. Intell. Syst.* **2020**, *15*, 310–316.
17. Jeong, J.; Park, H.; Kwak, N. Enhancement of SSD by Concatenating Feature Maps for Object Detection. In Proceedings of the British Machine Vision Conference, London, UK, 4–7 September 2017.
18. Cui, L.; Ma, R.; Lv, P.; Jiang, X.; Gao, Z.; Zhou, B.; Xu, M. MDSSD: Multi-scale deconvolutional single shot detector for small objects. *Sci. China (Inf. Sci.)* **2020**, *63*, 94–96. [CrossRef]
19. 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 $\times$ 16 Words: Transformers for Image Recognition at Scale 2021. *arXiv* **2020**, arXiv:2010.11929.
20. Xu, C.; Wang, J.; Yang, W.; Yu, H.; Yu, L.; Xia, G.S. Detecting tiny objects in aerial images: A normalized Wasserstein distance and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2022**, *190*, 79–93. [CrossRef]
21. Lin, T.-Y.; Dollar, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
22. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–20 June 2018; pp. 8759–8768.
23. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 10778–10787.
24. Ghiasi, G.; Lin, T.-Y.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7029–7038.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Long Beach, CA, USA, 4–9 December 2017.
26. Jacob, D.; Ming, C.; Kenton, L.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL), Minneapolis, MN, USA, 2–7 June 2019.
27. Alec, R.; Karthik, N.; Tim, S.; Ilya, S. Improving Language Understanding with Unsupervised Learning. *Tech. Rep.* **2018**, *4*.
28. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.

29. Tom, B.; Benjamin, M.; Nick, R.; Melanie, S.; Jared, K.; Prafulla, D.; Arvind, N.; Pranav, S.; Girish, S.; Amanda, A.; et al. Language models are few-shot learners. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020.
30. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-Local Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–20 June 2018; pp. 7794–7803.
31. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. GCNet: Non-Local Networks Meet Squeeze-Excitation Networks and Beyond. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1971–1980.
32. Bello, I.; Zoph, B.; Le, Q.; Vaswani, A.; Shlens, J. Attention Augmented Convolutional Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3285–3294.
33. Yin, M.; Yao, Z.; Cao, Y.; Li, X.; Zhang, Z.; Lin, S.; Hu, H. Disentangled non-local neural networks. In Proceedings of the European Conference on Computer Vision (ECCV), Online, 23–28 August 2020.
34. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3141–3149.
35. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation Networks for Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–20 June 2018; pp. 3588–3597.
36. Gu, J.; Hu, H.; Wang, L.; Wei, Y.; Dai, J. Learning Region Features for Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 381–395.
37. 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 (ECCV), Glasgow, UK, 23–28 August 2020; Proceedings, Part I 16. Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 213–229.
38. Cheng, C.; Fangyun, W.; Han, H. Relationnet++: Bridging visual representations for object detection via transformer decoder. In Proceedings of the Thirty-Fourth Annual Conference on Neural Information Processing Systems (NeurIPS), Vancouver, BC, Canada, 6–12 December 2020.
39. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable {detr}: Deformable transformers for end-to-end object detection. In Proceedings of the 2021 International Conference on Learning Representations, Online, 3–7 May 2021.
40. Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 14449–14458.
41. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 658–666.
42. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-iou loss: Faster and better learning for bounding box regression. In Proceedings of the 2020 AAAI Conference on Artifificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020; pp. 12993–13000.
43. Yang, X.; Yan, J.; Ming, Q.; Wang, W.; Zhang, X.; Tian, Q. Rethinking rotated object detection with gaussian Wasserstein distance loss. In Proceedings of the 2021 International Conference on Machine Learning (ICML), Online, 18–24 July 2021.
44. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. In Proceedings of the 2021 International Conference on Machine Learning (ICML), Online, 18–24 July 2021.
45. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks 2019. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7132–7141.
46. 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 2021 IEEE/CVF International Conference on Computer Vision (CVPR), Online, 19–25 June 2021; pp. 10012–10022.
47. Mehta, S.; Rastegari, M. MobileViT: Light-Weight, General-Purpose, and Mobile-Friendly Vision Transformer. *arXiv* **2022**, arXiv:2110.02178.
48. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1904–1916. [CrossRef]
49. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
50. Wang, C.Y.; Liao, H.Y.M.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
51. Everingham, M.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]

52. Zhang, Y.; Yuan, Y.; Feng, Y.; Lu, X. Hierarchical and Robust Convolutional Neural Network for Very High-Resolution Remote Sensing Object Detection. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5535–5548. [CrossRef]
53. Wang, J.; Yang, W.; Guo, H.; Zhang, R.; Xia, G.-S. Tiny Object Detection in Aerial Images. In Proceedings of the 2021 26th International Conference on Pattern Recognition (ICPR), Taichung, Taiwan, 18–21 July 2021; pp. 3791–3798.
54. 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 (ECCV), Zurich, Switzerland, 5–12 September 2014; pp. 740–755.
55. Xia, G.-S.; Bai, X.; Ding, J.; Zhu, Z.; Belongie, S.; Luo, J.; Datcu, M.; Pelillo, M.; Zhang, L. DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–20 June 2018; pp. 3974–3983.
56. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal Loss for Dense Object Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 318–327. [CrossRef] [PubMed]
57. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A one-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
58. Kong, T.; Sun, F.; Liu, H.; Jiang, Y.; Li, L.; Shi, J. FoveaBox: Beyound Anchor-Based Object Detection. *IEEE Trans. Image Process.* **2020**, *29*, 7389–7398. [CrossRef]
59. YOLO by Ultralytics (Version 8.0.0). Available online: https://github.com/ultralytics/ultralytics (accessed on 17 April 2023).
60. Liu, K.; Huang, J.; Li, X. Eagle-Eye-Inspired Attention for Object Detection in Remote Sensing. *Remote Sens.* **2022**, *14*, 1743. [CrossRef]
61. Li, Y.; Chen, Y.; Wang, N.; Zhang, Z.-X. Scale-Aware Trident Networks for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6053–6062.
62. Yang, Z.; Liu, S.; Hu, H.; Wang, L.; Lin, S. RepPoints: Point Set Representation for Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9656–9665.
63. Tian, Z.; Shen, C.; Chen, H.; He, T. FCOS: Fully Convolutional One-Stage Object Detection. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9626–9635.
64. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the Gap Between Anchor-Based and Anchor-Free Detection via Adaptive Training Sample Selection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 9756–9765.
65. Cai, Z.; Vasconcelos, N. Cascade R-CNN: Delving into High Quality Object Detection. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6154–6162.
66. Qiao, S.; Chen, L.-C.; Yuille, A. DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Online, 19–25 June 2021; pp. 10208–10219.