remote sensing

MDPI

*Article*

# Detection of Military Targets on Ground and Sea by UAVs with Low-Altitude Oblique Perspective

**Bohan Zeng [1], Shan Gao [2], Yuelei Xu [1,*], Zhaoxiang Zhang [1], Fan Li [1] and Chenghang Wang [1]**

[1] Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710129, China; zengbohan@mail.nwpu.edu.cn (B.Z.); zhangzhaoxiang@nwpu.edu.cn (Z.Z.); lifan@mail.nwpu.edu.cn (F.L.); chwang811@mail.nwpu.edu.cn (C.W.)

[2] Wuhan National Laboratory for Optoelectronics, Huazhong Institute of Electro-Optics, Wuhan 430073, China; gaoshan427@outlook.com

* Correspondence: xuyuelei@nwpu.edu.cn

**Abstract:** Small-scale low-altitude unmanned aerial vehicles (UAVs) equipped with perception capability for military targets will become increasingly essential for strategic reconnaissance and stationary patrols in the future. To respond to challenges such as complex terrain and weather variations, as well as the deception and camouflage of military targets, this paper proposes a hybrid detection model that combines Convolutional Neural Network (CNN) and Transformer architecture in a decoupled manner. The proposed detector consists of the C-branch and the T-branch. In the C-branch, Multi-gradient Path Network (MgpNet) is introduced, inspired by the multi-gradient flow strategy, excelling in capturing the local feature information of an image. In the T-branch, RPFormer, a Region–Pixel two-stage attention mechanism, is proposed to aggregate the global feature information of the whole image. A feature fusion strategy is proposed to merge the feature layers of the two branches, further improving the detection accuracy. Furthermore, to better simulate real UAVs' reconnaissance environments, we construct a dataset of military targets in complex environments captured from an oblique perspective to evaluate the proposed detector. In ablation experiments, different fusion methods are validated, and the results demonstrate the effectiveness of the proposed fusion strategy. In comparative experiments, the proposed detector outperforms most advanced general detectors.

**Keywords:** unmanned aerial vehicle (UAV); object detection; military targets; feature fusion strategy; hybrid detection model

**Citation:** Zeng, B.; Gao, S.; Xu, Y.; Zhang, Z.; Li, F.; Wang, C. Detection of Military Targets on Ground and Sea by UAVs with Low-Altitude Oblique Perspective. *Remote Sens.* **2024**, *16*, 1288. https://doi.org/10.3390/rs16071288

Academic Editor: Farid Melgani

Received: 1 February 2024
Revised: 25 March 2024
Accepted: 3 April 2024
Published: 5 April 2024

## 1. Introduction

In recent years, the evolution of military technology and the emergence of diverse threats [1] have posed significant challenges to conventional security measures. In response to these challenges, deep learning, a potent artificial intelligence technology, has gained prominence in the national defense and security field [2]. It plays a pivotal role in tasks including critical target tracking [3], scene matching for autonomous navigation in restricted conditions [4], and threat identification [5]. Given maneuverability and rapid deployment capabilities [6], small-scale and low-altitude unmanned aerial vehicles (UAVs) are frequently employed in site protection, targeted patrols, and strategic reconnaissance missions [1] to ensure swift responses to potential threats.

When identifying threat targets, conventional manual methods are inefficient, demanding a great deal of human resources and time investments. Furthermore, the outcomes frequently face issues of subjectivity and arbitrariness. Some handcrafted image processing methods, such as histogram of oriented gradient [7] (HOG) and Deformable Part Model [8] (DPM), have demonstrated drawbacks, including poor robustness and sensitivity to scale variations. Sumari [9] employed Support Vector Machine (SVM) to recognize aerial military targets. They utilized SVM to learn the knowledge of 11 features, including the wing, engine, fuselage, and tail of military airplanes, achieving good accuracy on their datasets.

In contrast, object detection techniques offer a promising solution for accurately locating and classifying specific targets, thus alleviating the aforementioned limitations. Du [10] proposed a lightweight military target detector. They employed the coordinate attention module to design the backbone, reducing the parameters and computation. Additionally, the authors proposed power parameter loss by combining EIOU and focal loss, further enhancing detection accuracy and convergence speed. Jafarzadeh [11] implemented YOLO based on automated detection of tanks. They constructed a tank dataset and utilized data augmentation to address the scarcity of targets. They achieved satisfactory results in real-time military tank detection. Jacob [12] introduced CNN for classifying military vehicles in Synthetic Aperture Radar (SAR) images. Through transfer training on a VGG16-pretrained model, they achieved classification accuracy of 98. Yu [13] presented a military target detector based on YOLOv3. By introducing deformable convolution and dual-attention mechanisms, the authors designed ResNet50-D network, which effectively improved the accuracy and speed for military target detection, providing better technical support for battlefield situation analysis.

Deploying detection algorithms on UAVs will significantly enhance the accuracy and efficiency of threat identification. However, detecting targets for low-altitude UAVs faces several challenges. Firstly, in scouting operations, UAVs may fly at diverse attitudes and angles, leading to drastic changes in the scale and background of the targets. Additionally, unlike remote sensing images captured from a top-down perspective at high altitudes, UAVs often monitor targets with a large oblique perspective for camouflage and concealment. Furthermore, potential threat targets are stealthy, disguised, and versatile, which further complicates recognition. Based on these challenges, the ideal dataset should meet two key requirements:

- The dataset is supposed to encompass a wide range of terrain and weather conditions, including both ground and water domains;
- The UAV's viewpoint has a large oblique perspective and altitude variations to match real scouting mission scenarios.

The performance of detectors heavily relies on thorough extraction of image information. RGB images exhibit a strong two-dimensional local structure [14] because neighboring pixels in adjacent regions are highly correlated. This property allows humans to comprehend high-level concepts even from a tiny portion of an image. CNN excels in capturing local feature information by effectively modeling pixel relationships within a defined kernel size. Moreover, the local receptive field and weight-sharing properties of the convolution kernel enable it to enjoy shift, rotation, and scale invariance. These inductive biases have enabled CNN-based detectors to achieve good performance over recent decades. Two-stage detector Faster R-CNN [15] first generated a vast number of predefined anchors and then regressed the positions of positive ones to obtain detection results. One-stage detectors such as the YOLO series [16–21] and SSD [22] leverage a single CNN to facilitate end-to-end detection and directly generate detection results. However, CNN-based detectors pay little attention to global information modeling.

By contrast, Transformer, characterized by a global self-attention mechanism, can establish long-range dependencies between remote pixels in an image. It provides better generalization and larger model capacity compared to CNN, but it necessitates pretraining on large datasets before transfer training on custom datasets. ViT [23] introduced the Transformer architecture to computer vision, achieving results comparable to CNN in classification tasks. DETR [24] was a pioneer in applying an encoder–decoder Transformer architecture in detection tasks. However, a global attention mechanism requires more computing resources and longer training time. In order to improve the efficiency of the Transformer block, subsequent works attempted to reduce the computational scope of self-attention, such as Deformable DETR [25], Swin Transformer [26], CSWin Transformer [27], and PVT [28]. Nonetheless, the weight matrix derived by self-attention between tokens, which describes the relevance of different areas in an image, is unable to model relationships between pixels within one token scope, implying a deficiency in extracting local information.

CNN and Transformer both have strengths and weaknesses. CNN is proficient in local perception but lacks global modeling capability, while Transformer excels in establishing long-range dependencies and gathering global information but lacks ability regarding extraction of localized features. Many studies have explored hybrid structures. CvT [14] proposed convolutional projection to replace the linear aspect in each Transformer block. It abandoned position embedding due to a built-in local context structure of convolutions, enabling the simplification of the process. BoTNet [29] achieved remarkable results in object detection tasks by replacing the convolution layer with a multi-head self-attention block in the final three bottlenecks of ResNet [30]. However, these works were only preliminary attempts for hybrid models as they merely sequentially coupled convolution and Transformer modules in turn without fully leveraging their respective strengths.

For the above design ideas and application scenarios, this article explores how to combine the feature modeling process of CNN and Transformer in a decoupled manner, aiming to leverage the merits of both and improve the performance of UAV object detection from a large oblique perspective. Our contributions are as follows:

1. A hybrid detection model that combines CNN and Transformer architectures is proposed for detecting military targets on ground and at sea. The detector incorporates T-branch Region–Pixel two-stage Transformer (RPFormer) and C-branch Multi-gradient Path Network (MgpNet) in a decoupled manner.
2. RPFormer, an efficient Transformer architecture consisting of Region-stage and Pixel-stage attention mechanisms, is proposed to gather the global information of an entire image. Additionally, MgpNet, based on Elan [21] block and inspired by the multi-gradient flow strategy [31], is introduced using local feature extraction.
3. A feature fusion strategy for hybrid models in channel dimensions is proposed. It fuses feature maps through three steps: Cross-Concatenation Interaction, Global Context Embedding, and Cross-Channel Interaction.
4. An object detection dataset regarding military background is constructed. The dataset consists of air-to-ground and air-to-sea scenarios and includes common combat units in a real scenario. All the images are captured from a large oblique perspective from 10 to 45 degrees at low altitude.

## 2. Related Work

### 2.1. CNN-Based Detectors

Benefiting from end-to-end paradigms to directly acquire target positions, one-stage detectors frequently exhibit superior inference speed. YOLO [16] divided one image into multiple grids, and each grid was responsible for predicting objects whose ground truth centroids fell within it. Based on YOLO, a series of improved algorithms have been proposed, such as SSD [22], YOLOv3 [18], and YOLOv7 [21]. SSD utilized multi-level feature maps to better deal with multi-scale issues of targets. RetinaNet [32] introduced focal loss, which effectively alleviated the imbalance problem of positive and negative samples. Two-stage detectors such as Fast R-CNN [33] and Faster-RCNN [15] operated in two steps. Firstly, anchor generation generated dense predefined candidate boxes on input images. This process may adopt methods like Region Proposal Network. Subsequently, each candidate box went through classification and position regression. While it showed excellent accuracy, it suffered from relatively slower inference speed. Building upon generalized CNN-based detectors, many efforts have been proposed for practical UAV detection tasks. Wang [34] proposed an improved YOLO detector for maritime ship detection. They improved the accuracy of the detector by introducing convolutional kernels of different shapes for tiny objects. Xiong [35] presented an enhanced YOLOv5 model for accurate litchi fruit recognition. The model incorporated Bi-directional Feature Pyramid Network for feature fusion and utilized Slicing Aided Hyper Inference for tiny target detection. The experimental results demonstrated higher AP compared to the original method.

Hou [36] introduced YOLOX-Pro for efficient landslide detection, especially in intricate mixed scenarios. By integrating VariFocal loss, they tackled sample distribution challenges, and the model demonstrated heightened capabilities in detecting landslides in complex environments. However, CNN-based detectors exhibit limitations in capturing global context due to their inherent local attention characteristics. A UAV-based object detection model necessitates consideration of not only the target attributes but also the contextual background environment for achieving enhanced and reliable detection outcomes.

### *2.2. Transformer-Based Detectors*

Transformer has gained popularity in the vision field due to excellent parallelization and global attention mechanisms. In detection tasks, Transformer can be implemented in two ways. One is adopting a complete encoder–decoder architecture: after preprocessing steps like patch embedding and position encoding, image tokens traverse cascaded encoders for feature extraction. The decoders receive Key, Value, and Query tensors and then conduct self-attention computations. Key and Value derive from the encoder, while Query is the semantic and location information carrier of the objects from initialization generation. DETR [24], as the pioneering work, viewed the detection task as a set prediction problem, eliminating tedious preprocessing and postprocessing steps like anchor generation and non-maximum suppression. It employed a bipartite matching strategy to produce detection results. However, the global attention mechanism results in slow convergence. Additionally, the limited resolution of feature maps faces challenges in scenarios with sharp scale variations. Subsequent works improved DETR from different aspects, such as Deformable DETR [25], Up-DETR [37], and DAB-DETR [38]. For instance, Deformable DETR proposed a deformable attention module. Each pixel paid attention to only a set of sampled points within its own neighborhood. The positions of these sampled points were learnable rather than fixed, enabling a locally sparse and efficient attention mechanism.

Another paradigm is designing an efficient Transformer encoder to replace the backbone of existing detection baselines like Mask-RCNN [39]. Transformer encoders, such as Pyramid Vision Transformer [28], served as generic backbones for feature extraction. However, a global attention mechanism means longer training time. To alleviate the problem, researchers introduced sparse attention to limited attention calculations in a local scope. Swin Transformer [26] divided the feature map into non-overlapping windows, limiting the attention computation scope. Shifted Windows Multi-Head Self-Attention was proposed to enable information interactions. CSWin Transformer [27] and CrossFormer [40] modified the window shape to axial strips and dilated windows, respectively. Zhao [41] designed Res-SwinTransformer with a Local Contrast Attention Network named RSLCANet for infrared small-target detection in aerial remote sensing. With fewer parameters, RSLCANet was suitable for practical deployment in applications like car navigation, crop monitoring, and infrared warning. In UAV object detection, accurately perceiving the location of targets is crucial. Transformers rely on position embeddings to introduce relative positions, while CNNs inherently excel at handling positional information in images. Therefore, integrating CNNs into detectors is necessary, particularly for small targets.

### *2.3. Hybrid Detectors*

The CNN–Transformer hybrid detector is designed to thoroughly exploit the strengths of both components, while how to efficiently integrate CNN and Transformer feature extraction modules is crucial for enhancing detector performance. In most hybrid detectors, CNN and Transformer modules are alternately employed and tightly integrated into a single network in a coupled manner. The Transformer module as the backbone initially performs global correlation modeling from the input image, and, subsequently, its output is used as the input for the CNN-based neck to conduct feature integration and scale adjustment. Lu [42] proposed a hybrid model for UAV image detection. It utilizes CSWinFormer as the backbone network to extract features, followed by a CNN-based neck network. A slicing-based inference method is introduced to enhance small-object detection accuracy. Ye [43] introduced a hybrid detector for low-altitude UAVs. They proposed an attention-enhanced Transformer block as the backbone to extract global information, and a CNN-based bottleneck is utilized to control the computation load. Zhao proposed YOLO-Vit [44] for UAV-based infrared vehicle detection, tackling issues like complex backgrounds and small targets causing high false alarms. They combined YOLOv7 with MobileViT for efficient feature extraction and designed C3-PANet neural network with the CARAFE upsampling method to enhance recognition accuracy. Ren proposed HAM-Transformer [45] Net for improving small-object detection in complex scenes of remote sensing images using UAVs. The approach combined a convolutional network with an adaptive multi-scaled Transformer, effectively extracting detailed features and hierarchical information. All of these hybrid detectors achieve good results, but they have not fully utilized the advantages of the fusion model.

Compared to the aforementioned series form, the parallel hybrid model, in which both the CNN and Transformer operate independently as separate branches, with each being responsible for extracting features from the input image, is a strategy worth considering. The parallel form is a decoupled design paradigm, facilitating the selection of the same depth and size feature maps to create a comprehensive representation according to the task requirements. This pattern follows the design philosophy of high cohesion and low coupling. Deng [46] introduced CTNet, aiming to enhance the classification capabilities of high-resolution remote sensing scenes. CTNet combined a CNN and Vision Transformer in parallel form, optimized using a joint loss function to enhance intraclass aggregation. This method demonstrated outstanding performance in classification tasks. We introduce the parallel paradigm into the detection task and redesign the Transformer branch. Furthermore, we propose a feature fusion module to eliminate the feature gap between the two branches.

## 3. Method

In this section, we first systematically introduce the overall architecture of the proposed detection model, as illustrated in Figure 1. Proposed hybrid detection model combines CNN and Transformer architecture in parallel form, which consists of C-branch and T-branch. We will then provide a detailed description of associated technologies:

- In Section 3.1, the core block Region-Pixel two-stage attention mechanism in RPformer is illustrated in Figure 2. RPformer, an efficient Transformer Network as depicted in Figure 3, is proposed as T-branch to gather information in global scope;
- In Section 3.2, the core block of MgpNet, efficient layer aggregation network's (Elan) block details, is illustrated in Figure 4. Inspired by multi-gradient flow strategy [31], as shown in Figure 5, a Multi-gradient Path Network (MgpNet), as depicted in Figure 6, is introduced as C-branch for extracting local information;
- In Section 3.3, a feature fusion strategy for the hybrid model is designed as depicted in Figure 7. The fusion module combines feature maps of the same scale through three steps in turn: Cross-Concatenation Interaction, Global Context Embedding, and Cross-Channel Interaction.

### *3.1. Region–Pixel Two-Stage Attention Mechanism*

Self-attention mechanism in Transformer makes it establish relationships among distant pixels, allowing for greater model capacity. However, it incurs larger computational load and slower convergence. The challenge lies in how to efficiently calculate attention between pixels. To tackle this issue, many handcrafted sparse attention patterns, such as limiting attention in local windows [26], axial stripes [27], and dilated windows [40], have been proposed to reduce computational loads of global attention mechanism. However, these approaches artificially design the attention scopes so Queries pay attention to Key–Value pairs only in these manually specified regions. On the one hand, the fixed shape and size of the region result in all Queries attending to the same set of Key–Value pairs; it is easy to cause the model to fall into suboptimal solutions, and, on the other hand, too many artificial settings make the model less robust.

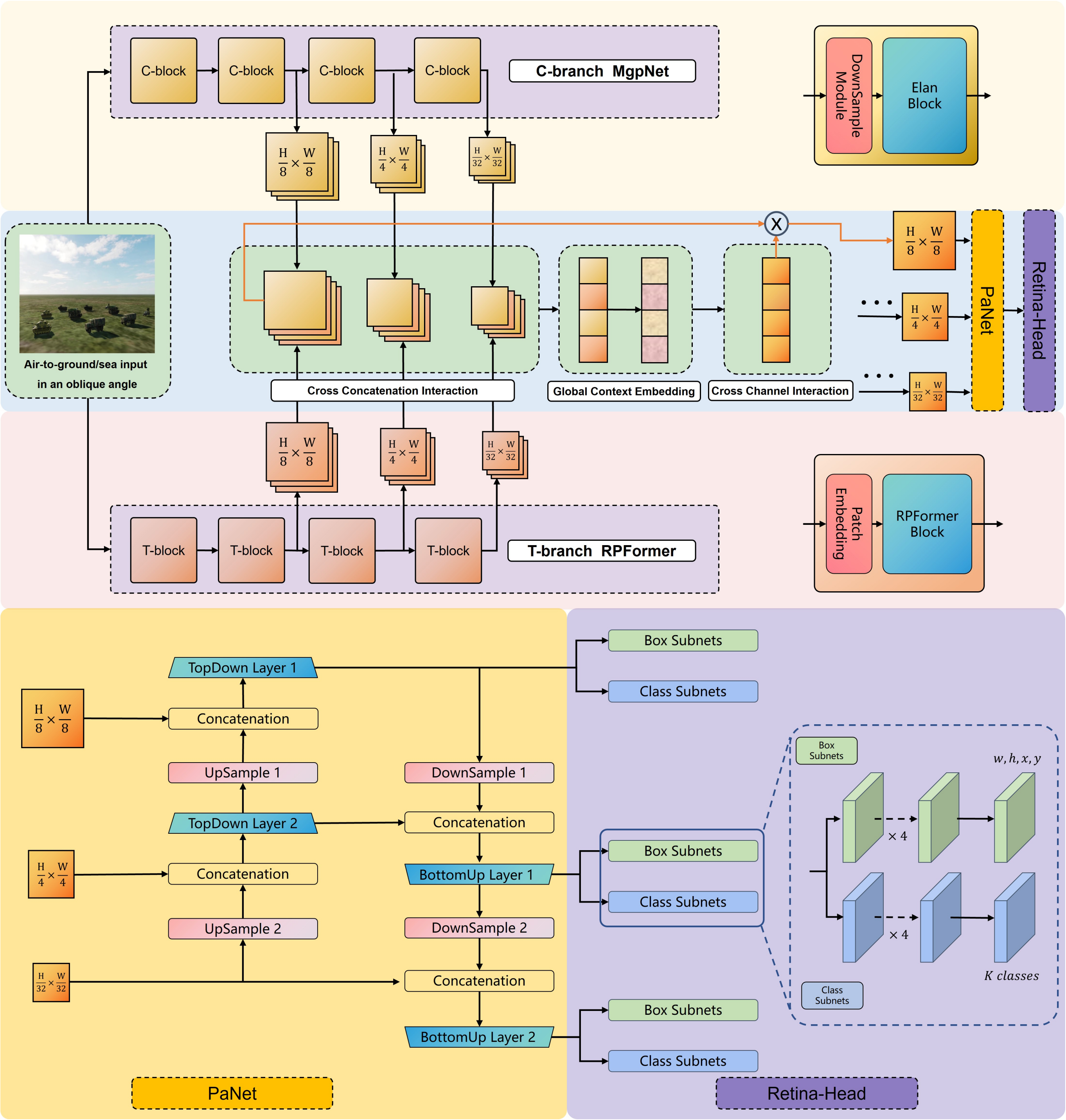

**Figure 1.** Framework of the proposed CNN–Transformer hybrid detection model.

In the original global attention mechanism, the weight matrix obtained by multiplying Q and K is an approximate sparse matrix after softmax operation, which indicates that there are some distant regions with weak correlations. There is no necessity to expend extra computational resources for these regions. Consequently, we aim at improving attention process so that pixels in certain regions focus only on those highly correlated with them. This should be a dynamic process that is determined by the specific data distribution rather than being manually configured. Proposed Region–Pixel two-stage attention mechanism helps to search the most relevant Key–Value pairs in an objective and data-driven pattern.

### 3.1.1. Region–Pixel Transformer Block

The core idea of proposed block is how to eliminate irrelevant Key–Value pairs and identify specific regions that each given region should focus on. As depicted in Figure 2, the Region–Pixel Transformer (RPFormer) comprises Region-stage attention and Pixel-stage attention. The former stage is tasked with identifying regions with the highest relevance, while the latter stage calculates self-attention within these identified regions. For illustrative purposes, we set the number of heads to one, and the same principle applies to the multi-head form.

- Region-stage attention

In the Region-stage attention, cosine similarity is employed to measure the correlation between feature vectors in two distinct regions. Moreover, we utilize convolutional projection instead of linear projection to obtain Query and Key–Value pairs, effectively reducing the computational load in this relevant region choice step. The details are provided below.

After downsampled module patch embedding or merging, feature map $X \in R^{C\times H\times W}$ is divided into $N^2$ regions by reshaping $X$ to $X_r \in R^{N^2\times \frac{HW}{N^2}\times C}$. These regions are non-overlapping, and each one contains the same number of feature vectors.

Next, specific methods are utilized to generate the Query tensor ($Q$) and Key–Value pair tensor ($K$ and $V$) from the partitioned feature map $X_r$ for computing self-attention. The initial Transformer [47] module adopts linear projection transformation:

$$\begin{aligned} Q &= X_r \cdot W_Q \\ K &= X_r \cdot W_K \\ V &= X_r \cdot W_V \end{aligned} \tag{1}$$

The weight matrices $W_Q$, $W_K$, and $W_V$ represent the linear transformations for $Q$, $K$, and $V$, respectively. However, CvT [14] introduced regular convolutional projection first to replace linear transformation and achieved additional modeling of local spatial context. Therefore, we employ Separable Convolution [48] with kernel size 3 to replace regular convolution, aiming to further minimize the computational burden and achieve additional local spatial context:

$$\begin{aligned} Q &= Pointwise \cdot conv\big(Batchnorm\big(Depthwise \cdot conv\big(X_r'\big)\big)\big) \\ K &= Pointwise \cdot conv\big(Batchnorm\big(Depthwise \cdot conv\big(X_r'\big)\big)\big) \\ V &= Pointwise \cdot conv\big(Batchnorm\big(Depthwise \cdot conv\big(X_r'\big)\big)\big) \end{aligned} \tag{2}$$

Separable Convolution consists of depthwise separable convolution and spatial separable convolution as in Equation (2). The BatchNorm Layer [49] is ultilized to accelerate the training process and improve the model's generalization.

Then measure the correlation between two regions. Assume there are feature vectors $a_1, a_2, \ldots, a_n$ in region $A$ and $b_1, b_2, \ldots, b_n$ in region $B$ . Correlation between feature vectors in two regions can be measured as follows:

$$cossim(a_i, b_j) = \frac{a_i^T b_j}{\|a_i\|_2 \|b_j\|_2} \tag{3}$$

When value is close to zero, it indicates that the correlation between $a_i$ and $b_j$ is weak. The region correlation is collectively described by all vectors in each region:

$$\begin{aligned} cossim(A,B) &= cossim((a_1,a_{2,\dots},a_n),(b_1,b_{2,\dots},b_n)) \\ &= cossim((a_1),(b_1,b_{2,\dots},b_n)) + \cdots + cossim((a_n),(b_1,b_{2,\cdots},,b_n)) \\ &= \left(\frac{a_1^T b_1}{\|a_1\|_2\|b_1\|_2} + \cdots + \frac{a_1^T b_n}{\|a_1\|_2\|b_n\|_2}\right) + \cdots + \left(\frac{a_n^T b_1}{\|a_n\|_2\|b_1\|_2} + \cdots + \frac{a_n^T b_n}{\|a_n\|_2\|b_n\|_2}\right) \\ &= (\hat{a}_1 + \cdots + \hat{a}_n)^T\left(\hat{b}_1 + \cdots + \hat{b}_n\right) \end{aligned} \tag{4}$$

where $\|\cdot\|$ is $l_2$ norm and $\hat{a}_i$ is unit vector. To parallelize the computation, channel normalization is employed to obtain normalized feature map $Q^{'}$ and $K^{'} \in R^{N^2 \times \frac{HW}{N^2} \times C}$, followed by summation over the region to derive Region-level Query $Q^{'}_{sum}$ and Key $K^{'}_{sum} \in R^{N^2 \times C}$. Then, a region-level weight matrix $W \in R^{N^2 \times N^2}$ is constructed by matrix multiplication between $Q^{'}_{sum}$ and the transposed $K^{'}_{sum}$:

$$W = Q^{'}_{sum}(K^{'}_{sum})^T \tag{5}$$

$W$ is a square matrix designed to quantify correlation among $N^2$ regions. For $i$th row in $W$, it quantifies the correlation between the region $i$th and the other $N^2$ regions, including itself $W[i,i] = 1$. We set a hyperparameter to retain $K$ elements with the largest value in each row besides $W[i,i]$ since the angle between it and itself is always 0 degrees. $I \in R^{N^2 \times K}$ records the indices of these regions:

$$I = KIndex(W) \tag{6}$$

The $i$th row in $I$ contains $K$ most relevant region's indices for the $i$th region.

- Pixel-stage attention

In the Pixel-stage attention step, we calculate the attention between the Query and the Key–Value pairs in the $K$ regions preserved in the Region-stage attention step. However, the index numbers differ in each row. To optimize model's parallelization, we aggregate the Key–Value pair's tensors:

$$\begin{aligned} K_r &= f_{torch.gather}(K,I) \\ V_r &= f_{torch.gather}(V,I) \end{aligned} \tag{7}$$

where $K_r, V_r \in R^{K\frac{HW}{N^2} \times C}$ are gathered tensor. We can then calculate attention via gathered Key–Value pairs:

$$O = SelfAttention(Q, K_r, V_r) \tag{8}$$

The calculation formula for spatial attention is as follows:

$$SelfAttention(Q, K_r, V_r) = softmax(\frac{QK_r^T}{\sqrt{C}})V_r \tag{9}$$

$\sqrt{C}$ is the scalar factor to avoid gradient vanishing [47]. The final output of the feature map $O \in R^{\frac{HW}{N^2} \times C}$. For example, Figure 2 shows attention result between the Query in the last region and the Key–Value pairs in 2nd, 5th, and 6th regions. The Key–Value pairs' indices are obtained in Region-stage attention step.

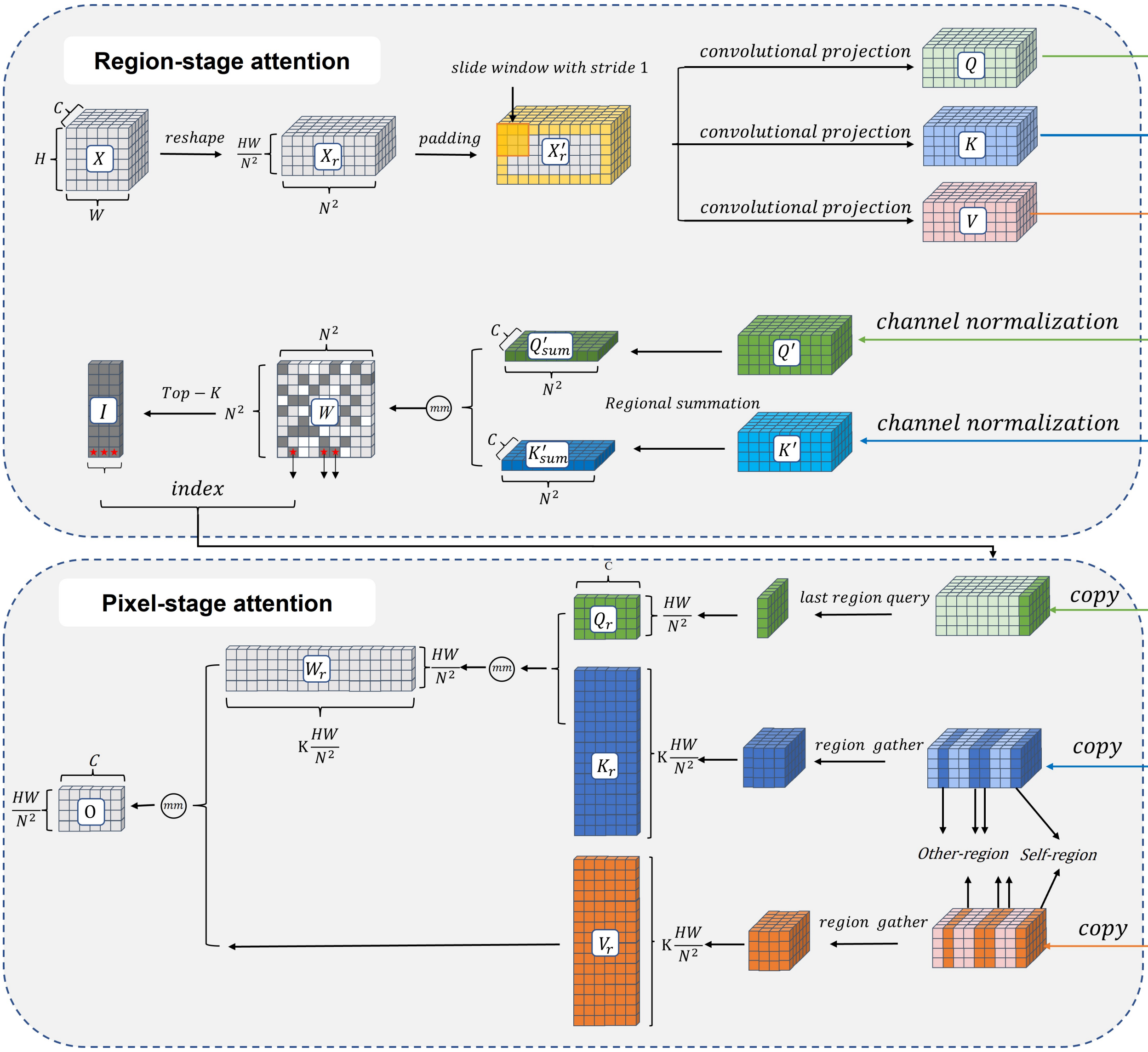

**Figure 2.** Region–Pixel two-stage attention mechanism of RPFormer block. The top and bottom parts are Region-stage attention and Pixel-stage attention steps, respectively. The red stars in Region-stage attention step represent the areas with the highest relevance to be retained. The arrows indicate transferring the indices of these areas to Pixel-stage attention step for sparse attention calculation.

### 3.1.2. Structure

The proposed Region–Pixel Transformer is employed as the T-branch. It is a hierarchical structure consisting of four stages as depicted in Figure 3a. In the first stage, patch embedding is used as downsampling module to reduce spatial resolution while increasing the number of channels. In other three stages, the downsampling module is replaced with patch merging operation. Each stage is composed of $N_i$ consecutive blocks as illustrated in Figure 3b. Each block includes a depth-wise convolution for implicit relative position encoding, followed by a Region–Pixel two-stage attention module and a 2-layer multilayer perceptron (MLP) module. LayerNorm [50] and residual connection [30] techniques are employed. After carefully assessing the tradeoff between model complexity and efficiency, we set the depth in each stage, denoted by the number of Region–Pixel Transformer blocks as [4, 4, 18, 4]. The expansion ratio for MLP is 4. The channel width and head dimension are set to 64 and 32; the region size and $K$ are defined as 7 and 3.

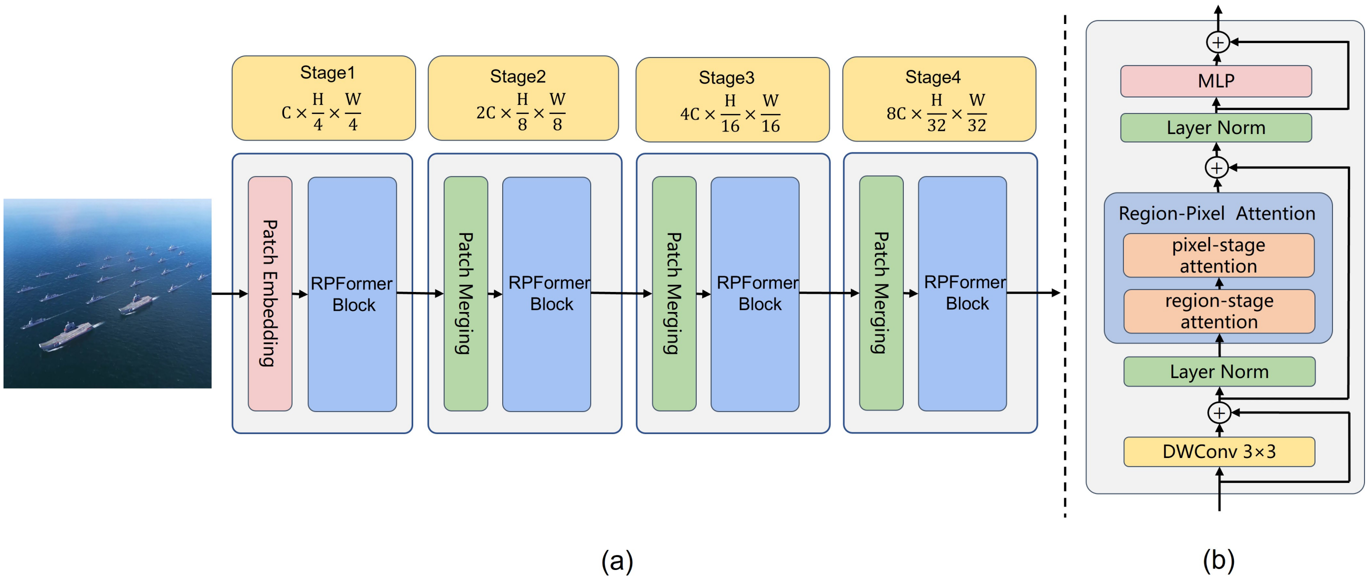

**Figure 3.** (**a**) The overall architecture of the RPFormer; (**b**) details of a RPFormer block.

## *3.2. Multi-Gradient Path Network*

Many classic networks, such as DarkNet [18] and ResNet [30], have been designed with a strong emphasis on feed-forward propagation, which refers to the direction of data flow. The objective is to extract features with specific attributes through feature selection, thereby enhancing the expressiveness of the network as the sensitivity of convolutional layers varies at different stages. Shallow convolutional layers have relatively small receptive fields but high resolution and strong localization. They excel at capturing detailed information, such as specific directional textures, colors, and certain shape templates. In contrast, deep layers have larger receptive fields and lower resolution, allowing them to capture high-level semantic information but making it challenging to detect small objects.

Nonetheless, the conventional practice in training CNNs typically employs the backward propagation algorithm. This algorithm initiates the gradient flow from the loss function and propagates it backward along the computational graph. This process enables the guidance of gradient information for weight updates in a gradient descent manner, ultimately facilitating the convergence of the network. So, rather than solely emphasizing feed-forward propagation, our approach is more concerned with gradient flow combination analysis.

The proposed Multi-gradient Path Network (MgpNet) was inspired by gradient path design strategy [31]. Referring to efficient layer aggregation network's (Elan) block [21], we introduced the MgpNet with the aim of enhancing the network's capacity for more gradient flow combinations during the training process.

### 3.2.1. Elan Block

Firstly, we analyze the structure of the Elan block. Subsequently, we study the block from viewpoint of Gradient Source and Gradient Timestamp when the gradient flows through the Elan block. The former focuses on the static perspective, evaluating the quantity of gradient flows at a specific moment. The latter takes a dynamic standpoint, assessing ways of gradient flow combinations at each moment.

Elan block details are shown in Figure 4. Short branch in Elan block acts as cross-stage connection, utilizing a 1 × 1 convolution to reduce the number of channels while keeping the feature map size unchanged:

$$X_{short} = f_{CBS(ks=1\times1)}(X_{input}) \tag{10}$$

$f_{CBS}$ represents the non-linear transformation function, which combines the operations of Convolutional layer, Sigmoid-Weighted Linear Unit (SiLU) layer, and Batch Normalization (BN) layer. In the main branch, a 1 × 1 convolution is applied before cascading $k$ convolutional modules; each one consists of $m$ 3 × 3 convolutional layers:

$$\begin{aligned} X_{main} &= f_{CBS(ks=1\times1)}(X_{input}) \\ X_k &= F_{ConvMoudule}(X_{k-1}) \\ &= f^{1}_{CBS(ks=3\times3)}(\dots(f^{m}_{CBS(ks=3\times3)}(X_{k-1}))) \end{aligned} \tag{11}$$

The incorporation of multiple branches enriches the gradient flow. Eventually, there will be $k+2$ branches of feature maps concatenated together.

$$X_{out} = [X_{short}; X_{main}; X_{\mathrm{I}}; \dots; X_{\mathrm{k}}] \tag{12}$$

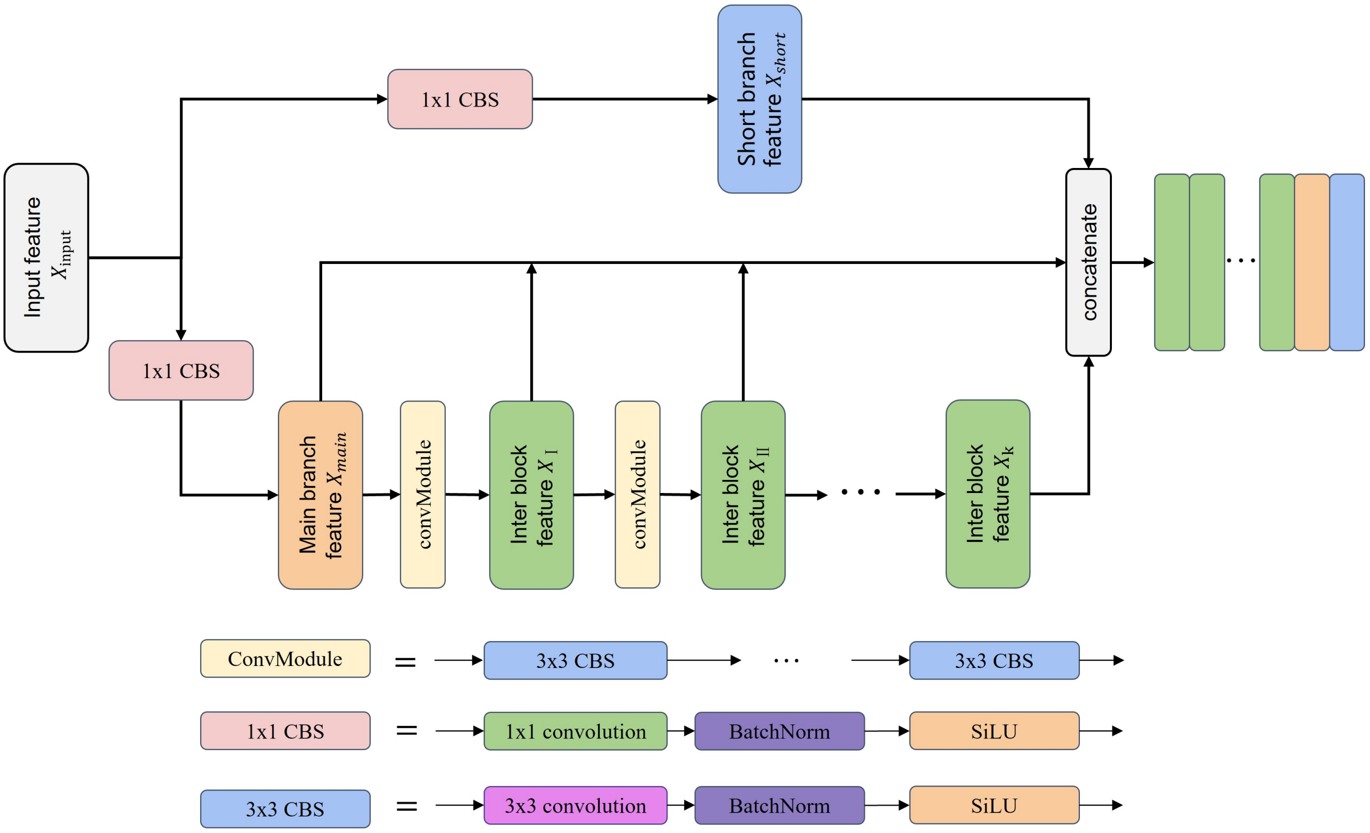
Input feature $X_{input}$
1x1 CBS
1x1 CBS
Short branch feature $X_{short}$
concatenate
Main branch feature $X_{main}$
convModule
Inter block feature $X_{\mathrm{I}}$
convModule
Inter block feature $X_{\mathrm{II}}$
Inter block feature $X_{\mathrm{k}}$
ConvModule
3x3 CBS
3x3 CBS
1x1 CBS
1x1 convolution
BatchNorm
SiLU
3x3 CBS
3x3 convolution
BatchNorm
SiLU

**Figure 4.** The efficient layer aggregation network's (Elan) block details.

When updating parameters in gradient descent manner, the gradient flows between layers in the block as shown in Figure 5. For the last convolutional layer in $ConvModule_i$, the Gradient Source consists of the shortcut branch and its predecessor $ConvModule_{i+1}$:

$$\begin{aligned} \dot{w}_i &= \begin{cases} f(w_i, g_{shortcut}, g_{i+1}), & i \in 1,2,3,\dots,k-1 \\ f(w_i, g_{shortcut}), & i = k \end{cases} \\ w_i &= w_i - lr \times \dot{w}_i \end{aligned} \tag{13}$$

$g_{short}$ represents the gradient information returned directly from concatenated feature maps through shortcut connection, and $g_{i+1}$ represents the ones of previous layer's connection. It is evident that the Elan block possesses a wealth of Gradient Source during network training.

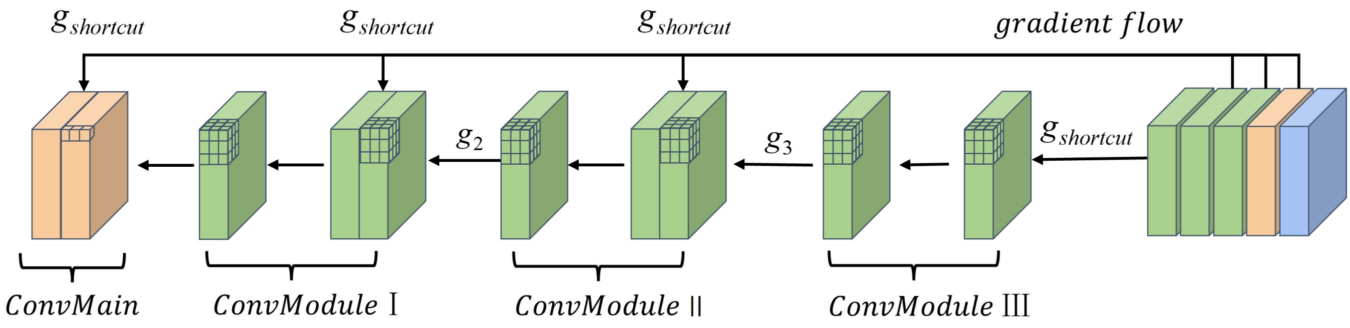

**Figure 5.** Gradient Source of main branch in Elan block.

Gradient flows through the block layer by layer in a breadth-first manner and reaches each layer at different time points. The Gradient Timestamp records the specific moment when it traverses across the entire network. The following Table 1, based on the arrival Gradient Timestamp, can be generated.

When passing through the block in Figure 5, gradient will first arrive at ConvModule's shortcut layer at the same time due to the shortcut connection. Subsequently, these shortcut layers act as successor nodes, and the gradient sequentially moves forward in a breadth-first propagation manner, enabling it to simultaneously access successor layers that shortcut connections have reached in the previous moment, only until they have completed flowing through the entire Elan block. Each column in the table represents one gradient combination, resulting in 10 different gradient combinations in total. This strategy enables the network to efficiently acquire and update gradient information, thereby enhancing its learning ability and training efficiency.

**Table 1.** Gradient Timestamp of main branch in Elan block.

| **Module** | *ConvMain* | | *ConvModule*I | | | *ConvModule*II | | | *ConvModule*III | |
|---|---|---|---|---|---|---|---|---|---|---|
| **ConvLayer** | **Layer1** | | **Layer2** | **Layer3** | | **Layer4** | **Layer5** | | **Layer6** | **Layer7** |
| **Shortcut** | ✓ | / | / | ✓ | / | / | ✓ | / | / | / |
| moment·1 | G1 | / | / | G1 | / | / | G1 | / | / | G1 |
| moment·2 | / | / | G2 | / | / | G2 | / | / | G2 | / |
| moment·3 | / | G3 | / | / | G3 | / | / | G3 | / | / |
| moment·4 | / | / | G4 | / | / | G4 | / | / | / | / |
| moment·5 | / | G5 | / | / | G5 | / | / | / | / | / |
| moment·6 | / | | G6 | / | / | / | / | / | / | / |
| moment·7 | / | G7 | / | / | / | / | / | / | / | / |

### 3.2.2. Structure

MgpNet is composed of four stages as shown in Figure 6a. Each stage consists of a downsampling module and an Elan block. The downsampling module combines max pooling with convolutional layers in a parallel pattern as shwon in Figure 6b. The Elan block incorporates only two sizes of convolution kernels, silu activation function and batch normalization layer, without complex designs such as automatic search [51] and complex scaling [52]. In each Elan block, there are $k$ cascading convolutional modules, and each module consists of $m$ $3 \times 3$ convolutional layers. We set $k$ and $m$ to 3 and 2, respectively.

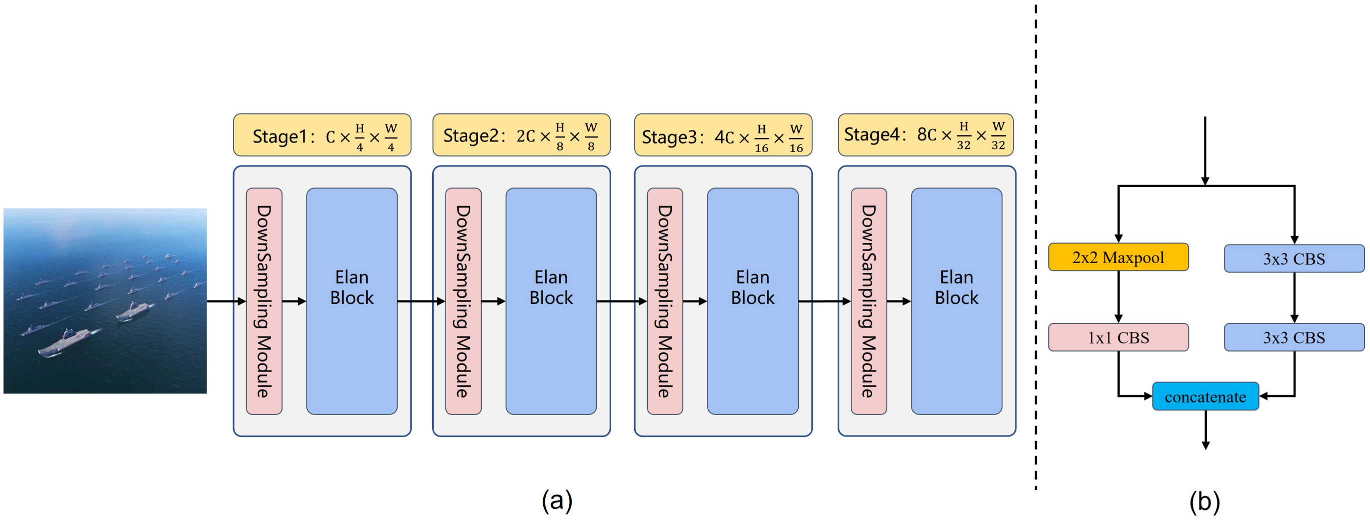
Stage1: $C \times \frac{H}{4} \times \frac{W}{4}$
Stage2: $2C \times \frac{H}{8} \times \frac{W}{8}$
Stage3: $4C \times \frac{H}{16} \times \frac{W}{16}$
Stage4: $8C \times \frac{H}{32} \times \frac{W}{32}$
DownSampling Module
Elan Block
2x2 Maxpool
3x3 CBS
1x1 CBS
3x3 CBS
concatenate
(a)
(b)

**Figure 6.** (**a**) The structure of the MgpNet; (**b**) downsampling module.

### 3.3. Channel-Attention-Based Feature Fusion Module

We propose a feature fusion strategy for fusing features extracted from two branches. The feature map downsampling ratios of two branches in four stages are both 4, 8, 16, and 32, corresponding to 64, 128, 256, and 512 channels. Feature fusion comprises three steps in turn: Cross-Concatenation Interaction, Global Context Embedding, and Cross-Channel Interaction, as shown in Figure 7.

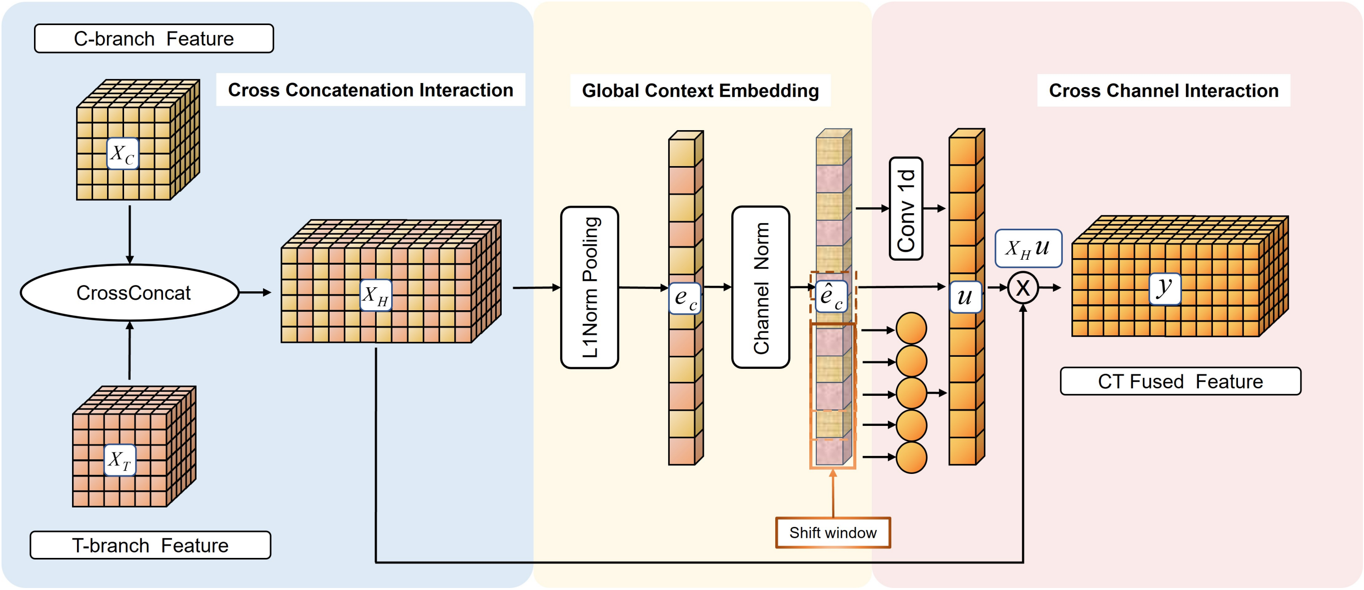

**Figure 7.** An overview structure of proposed feature fusion strategy.

#### 3.3.1. Cross-Concatenation Interaction

The abstract information of each pixel in the feature map is embedded in the channel dimension. As a result, deeper layers with high-level semantic information typically have a smaller scale and a larger number of channels. To effectively combine the channel information of feature maps $X_C$ and $X_T$ generated by C-branch and T-branch, respectively, we employ Cross-Concatenation Interaction operation to alternately stack $X_C$ and $X_T$ together:

$$\begin{aligned} X_H[0::2,:,:] &= X_C \\ X_H[0::1,:,:] &= X_T \end{aligned} \tag{14}$$

Here, $X_C$,$X_T \in R^{C\times H\times W}$, $X_H \in R^{2C\times H\times W}$.

### 3.3.2. Global Context Embedding

To extract global contextual information from each channel, we utilize $l_1$ norm average pooling as the channel information embedding module. The pooling operation aggregates each channel of the feature map to generate a fixed-size global feature vector $e \in R^{2C\times H\times W}$. The element of $e$ is calculated by formula as follows:

$$e_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_H[c,:,:](i,j) \tag{15}$$

Following work [53], we also apply channel normalization operation. $e = [e_1, e_2, \ldots, e_{2C}]$, which is defined by the following formula:

$$\hat{e} = \frac{\sqrt{2C}e_c}{\|e\|_2} = \frac{\sqrt{2C}e_c}{\left[\sum_{c=1}^{2c} e_c^2 + \varepsilon\right]^{\frac{1}{2}}} \tag{16}$$

where $\varepsilon$ is a small constant to avoid calculation error leading by zero value, $\hat{e} \in R^{2c\times 1\times 1}$.

### 3.3.3. Cross-Channel Interaction

We utilize $1D$ convolution to efficiently implement local Cross-Channel Interaction to the vector generated from Global Context Embedding operation. Specifically, we hope every element in the vector pays attention to itself and its $K$ neighbors as the formula:

$$u = Conv1d(\hat{e}) \tag{17}$$

Here, $u \in R^{C\times 1\times 1}$ and $K$ is a hyperparameter. For instance, when $K = 5$, the 3rd channel calculates the interaction attention with the 1st, 2nd, 4th, and 5th channels, as well as itself. We derive channel weights and compute the dot product with the original feature map in the specified order:

$$y = X_H u \tag{18}$$

where $y \in R^{2C\times H\times W}$. Proposed fusion module has the advantage of being plug-and-play, and we take the output of the last three stages of the branches as the input to the fusion module for feature fusion.

## 4. Datasets

Aerial remote sensing images are commonly captured from a top-down perspective at high altitudes [54–56]. However, for UAV reconnaissance patrol missions, collecting a number of images from such a viewpoint proves impractical. The top-down perspective makes it easier for the UAV to expose its position and be attacked. In contrast, a large oblique perspective allows for better concealment and reduces the risk of being detected. Additionally, the top-down perspective provides flattened planar information, while oblique perspectives offer more details, rendering targets more visibly distinct and facilitating localization and identification of potential threats.

Based on the above background, the existing datasets cannot meet our demands. We created a military targets dataset for air-to-ground and air-to-sea scenarios to simulate more realistic environments. The dataset includes common units on ground and at sea. Partial images in the dataset are shown in Figure 8.

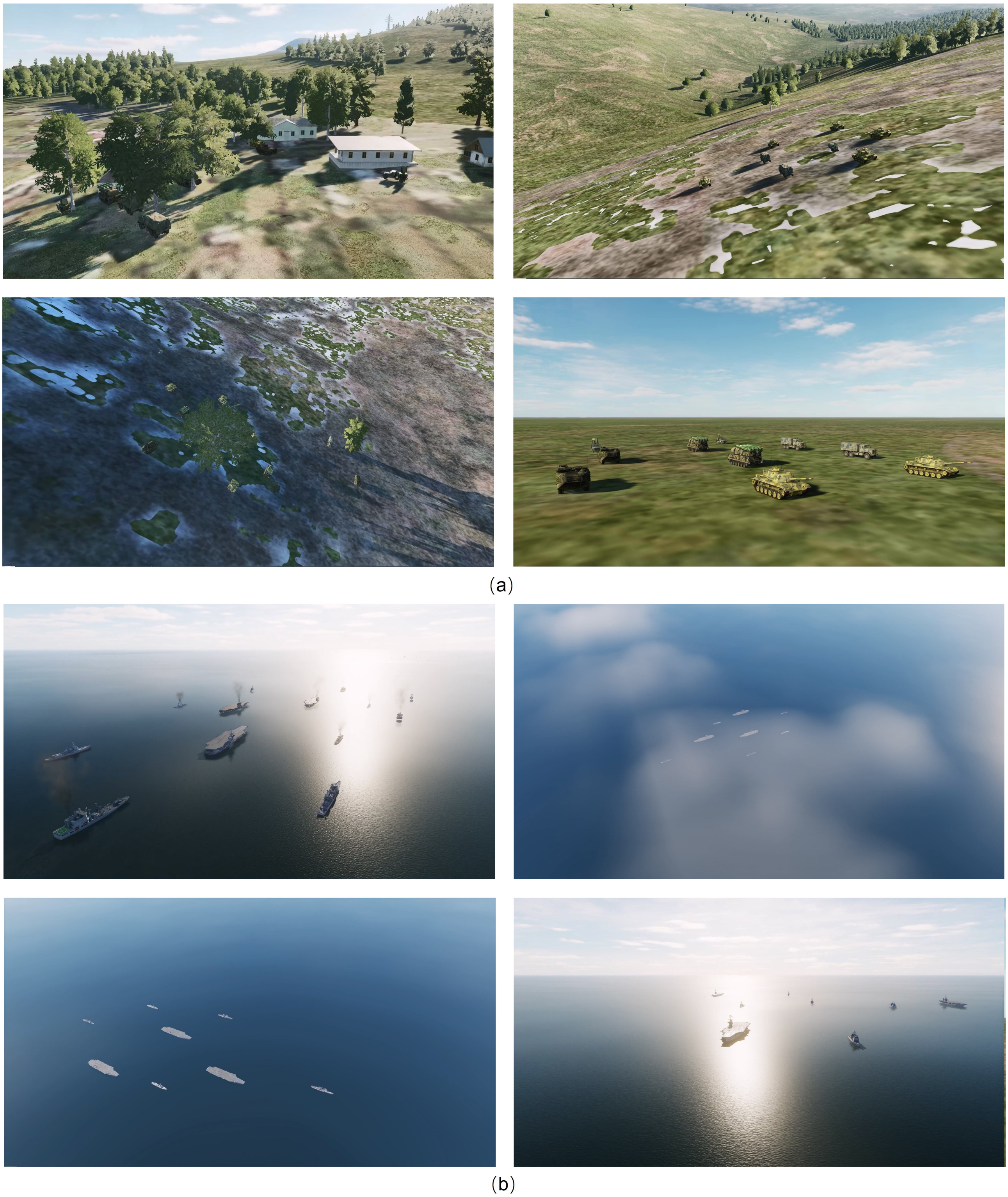

**Figure 8.** Examples of created dataset images. (**a**) Air-to-ground scenario; (**b**) air-to-sea scenario.

### *4.1. Simulation Environments*

Digital Combat Simulator (DCS) World is a sandbox game in which maps and combat units are designed to be very realistic and to accurately reproduce real-world situations. The proposed military dataset is based on the game and includes air-to-ground and air-to-sea scenarios. In the air-to-sea scenario, we collected combat units of six categories, namely aircraft carriers, cruisers, destroyers, patrol ships, speedboats, and landing ships.

We classified the ships into three categories: large warships, medium warships, and small warships. Large warships, typically referring to aircraft carriers, have a displacement of over 20,000 tons; medium warships, including cruisers and destroyers, have a displacement between 2000 and 20,000 tons; and small warships, including patrol ships, speedboats, and landing ships, have a displacement below 2000 tons. In the air-to-ground scenario, we collected combat units in five categories: tanks, radar vehicles, transport vehicles, rocket launchers, and armored vehicles.

### *4.2. Details of Dataset*

We set an oblique perspective from 10 to 45 degrees to capture images. In the air-to-ground scenario, the UAVs' flight altitude ranges from 25 to 200 m, while, in the air-to-sea scenario, the flight altitude ranges between 200 and 1000 m. To make the data more reliable and generalized, there are four weather conditions: sunny, cloudy, overcast, and cloudy with rain. We carefully filtered and cleaned the data and removed any irrelevant or redundant information. We also classified and labeled each image with bounding boxes using both manual and semi-automatic methods. In the air-to-ground scenario, there are 5238 images, including 31,606 boxes. In the air-to-sea scenario, there are 4878 images, including 38,702 boxes. The image resolution is 1080 × 1920. The detailed contents of the dataset are available in Table 2. Considering the constraints of an actual reconnaissance mission, the UAV flies at an almost fixed altitude for 15 rounds in each scenario. The distribution details of the dataset are visualized in Figure 9.

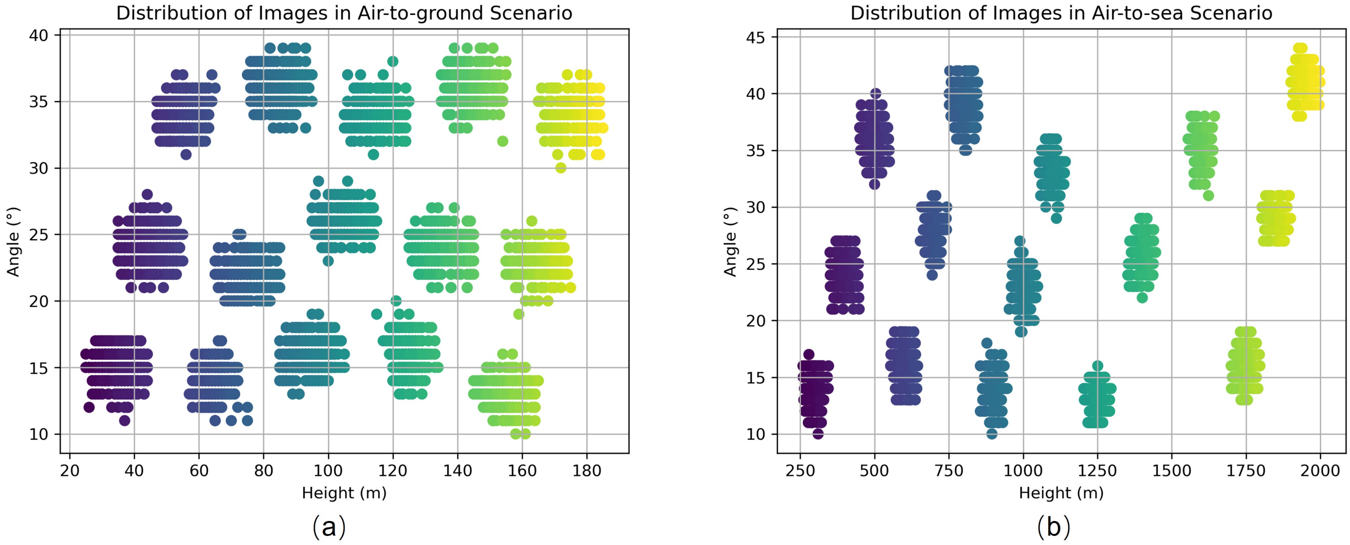

**Figure 9.** Distribution details of the dataset. (**a**) Air-to-ground scenario; (**b**) air-to-sea scenario.

**Table 2.** Details of created military dataset.

| Scenario | Target Category | Box Number | Height Range | Image Size | Angle Range |
|---|---|---|---|---|---|
| Air-to-ground | tank<br>radar vehicle<br>transport vehicle<br>rocket launcher<br>armored vehicle | 6598<br>4186<br>6850<br>7216<br>6756 | 25–200 m | 1080 × 1920 pixels | 10–45 degrees |
| Air-to-sea | large warship<br>medium warship<br>small warship | 11,128<br>8440<br>19,134 | 200–2000 m | | |

## 5. Experiments and Discussion

### *5.1. Experimental Setup*

We carried out all the experiments on a personal computer equipped with an Intel CPU E5-2678 and NVIDIA GeForce RTX 3090 (24 G). The operating system was Ubuntu 20.04 LTS. Python was 3.8. We completed codes in PyTorch framework 1.13.1 with the help of toolbox MMdetection. In the optimization step, we used the SGD optimizer with Nesterov momentum. The momentum was 0.937, the initial learning rate was 0.01, and the weight decay was 0.0005. We did not use pretrained weights to initialize the network since there is a significant difference in data distribution between generic datasets like ImageNet and the specific UAVs scenario mentioned above. We applied the Kaiming initialization [57] method to initialize the weights. The number of iterations was set to 100 epochs and batchsize was set to 16. We employed mean Average Precision (mAP), Average Precision (AP) at 0.5 and 0.75 IoU thresholds, APs for small object sizes (less than 32 × 32 pixels), APm for medium object sizes (between 32 × 32 pixels and 96 × 96 pixels), and APl for large object sizes (larger than 96 × 96 pixels) as our evaluation metrics.

We divided the collected data into two subdatasets according to the air-to-ground and air-to-sea scenarios. For each dataset, we split it into a training set and testing set, with a ratio of 80 percent and 20 percent, respectively. We resized the input data to 640 × 640.

### *5.2. Comparison with State of the Art*

To provide a comprehensive comparison of the proposed model, we conducted comparative experiments. We employed RetinaNet [32] as a baseline and replaced several advanced backbone networks, including CNN structures such as EfficientNet [52], ResNest [58], CSPDarkNet [18], and ResNet [30] , as well as Transformer structures such as Swin Transformer [26], Pyramid Vision Transformer [28], and CSWin Transformer [27]. The results are presented in Table 3.

**Table 3.** Comparative experiments of different methods in air-to-sea and air-to-ground scenarios.

| Scenario | Method | Flops (G) | Params (M) | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|---|---|
| Air-to-sea | EfficientNet-b0 | 29.8 | 5.3 | 50.0 | 87.3 | 49.5 | 6.6 | 41.5 | 66.1 |
| | ResNet-50 | 47.6 | 25.6 | 52.0 | 89.0 | 51.0 | 7.6 | 43.1 | 66.6 |
| | ResNest-50 | 50.9 | 27.5 | 53.6 | 91.1 | 55.0 | 11.0 | 46.1 | 68.2 |
| | CSPDarkNet-53 | 66.9 | 43.2 | 54.5 | 91.5 | 56.1 | 13.7 | 47.2 | 69.2 |
| | PVT-s | 51.3 | 24.5 | 54.8 | 92.1 | 55.0 | 13.8 | 47.1 | 69.0 |
| | Swin-s | 49.4 | 49.6 | 55.7 | 92.8 | 57.8 | 14.2 | 48.6 | 69.4 |
| | CSWin-s | 42.7 | 35.3 | 56.0 | 93.1 | 58.4 | 14.6 | 49.1 | 69.6 |
| | Ours | 61.7 | 37.5 | 57.9 | 95.1 | 60.2 | 17.1 | 50.3 | 70.6 |
| Air-to-ground | EfficientNet-b0 | 29.8 | 5.3 | 58.7 | 83.1 | 65.9 | 6.8 | 57.2 | 83.9 |
| | ResNet-50 | 47.6 | 25.6 | 61.1 | 85.4 | 68.3 | 8.4 | 60.1 | 84.6 |
| | ResNest-50 | 50.9 | 27.5 | 62.9 | 87.3 | 70.2 | 9.9 | 61.2 | 84.9 |
| | CSPDarkNet-53 | 66.9 | 43.2 | 63.7 | 88.2 | 71.1 | 10.7 | 61.9 | 85.7 |
| | PVT-s | 51.3 | 24.5 | 64.6 | 89.1 | 72.7 | 12.2 | 63.4 | 85.8 |
| | Swin-s | 49.4 | 49.6 | 65.2 | 89.6 | 73.3 | 14.4 | 64.7 | 86.1 |
| | CSWin-s | 42.7 | 35.3 | 66.5 | 90.4 | 74.5 | 14.9 | 65.2 | 86.4 |
| | Ours | 61.7 | 37.5 | 68.9 | 92.5 | 76.9 | 18.1 | 67.6 | 87.4 |

In air-to-sea and air-to-ground scenarios, the proposed hybrid method achieved mAP values of 57.9 and 68.9, demonstrating better performance compared to the single-branch network in various metrics, particularly in APs, where it improved by 2.5 and 3.2 over CSWin-s. This proved the effectiveness of the hybrid network. The inference results in two scenarios are shown in Figures 10 and 11. It can be seen that the proposed method achieved better results when targeting certain distant targets.

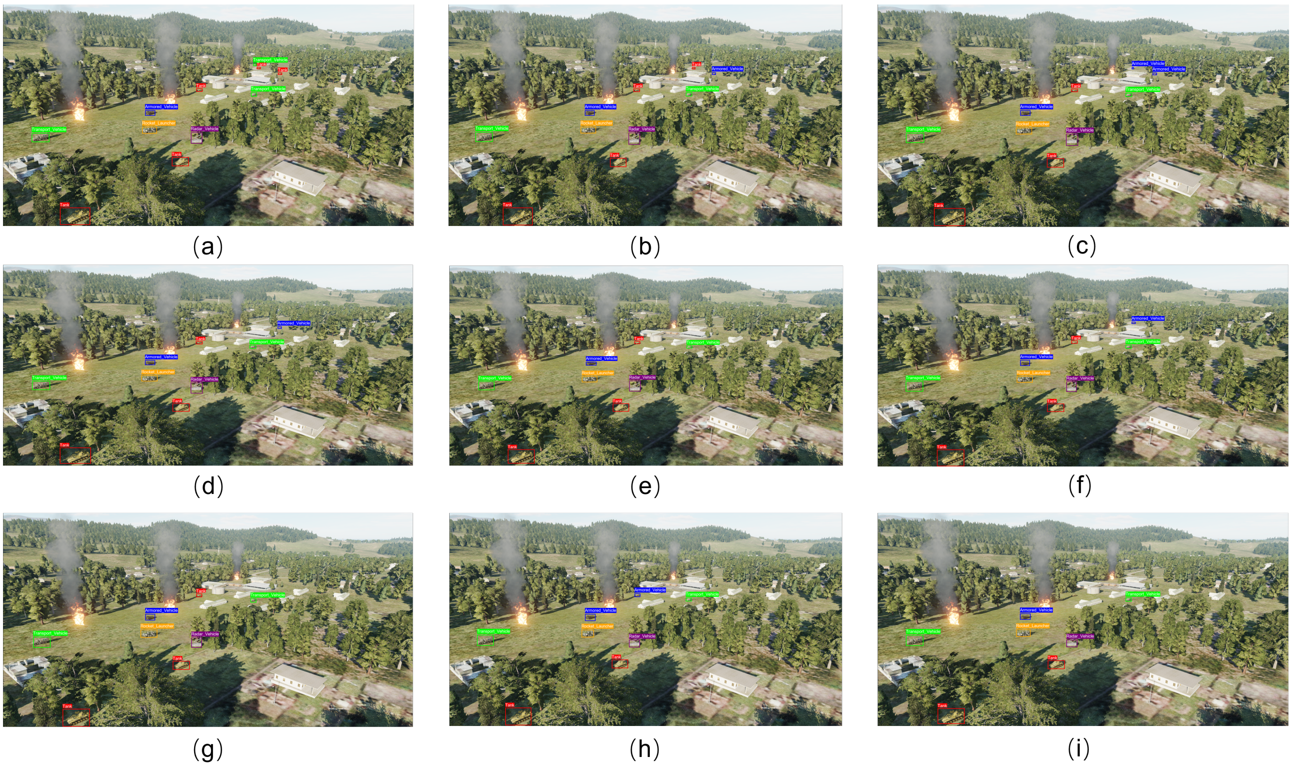

**Figure 10.** The comparative inference results of different methods in the air-to-ground scenario. (**a**) Ground truth; (**b**) proposed hybrid method; (**c**) CSWin Transformer-small; (**d**) Swin Transformer-small; (**e**) PVT-small; (**f**) RestNest-50; (**g**) CspDarkNet-53; (**h**) ResNet-50; (**i**) EfficientNet-b0.

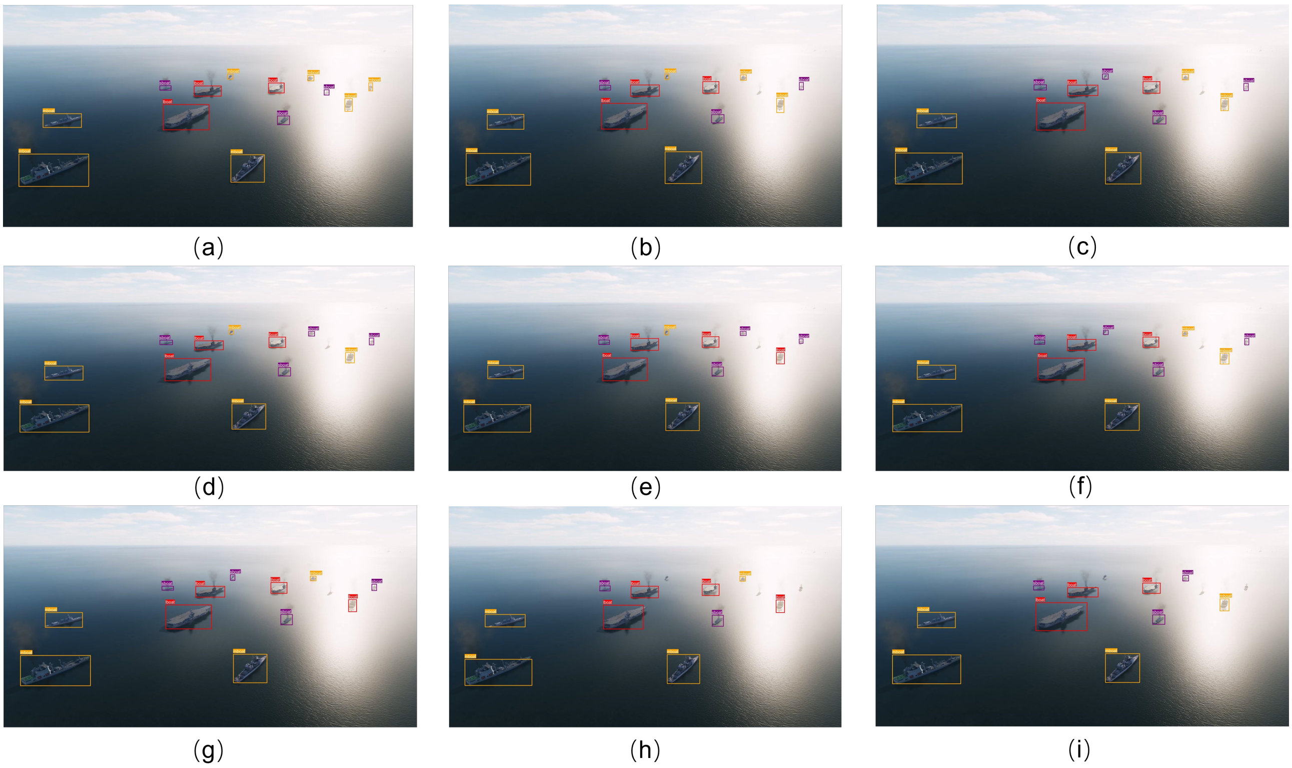

**Figure 11.** The comparative inference results of different methods in the air-to-sea scenario. (**a**) Ground truth; (**b**) proposed hybrid method; (**c**) CSWin Transformer-small; (**d**) Swin Transformer-small; (**e**) PVT-small; (**f**) RestNest-50; (**g**) CspDarkNet-53; (**h**) ResNet-50; (**i**) EfficientNet-b0.

To further analyze the feature extraction process of these models, we used the Grad-CAM [59] method to visualize the feature maps at different depths. We representatively selected the feature maps of CSWin-s, CSPDarkNet-53, and our method in three stages, and the heatmaps are shown in Figures 12 and 13. It can be seen that the shallow layers emphasized local texture, and, as the network deepened, it was adept at capturing abstract semantic information. The proposed method exhibited larger coverage in the target scope compared to the other models at the same depth, which could obtain more comprehensive features. At the same location, the pixels around the military targets in our method were darker, indicating a higher level of response.

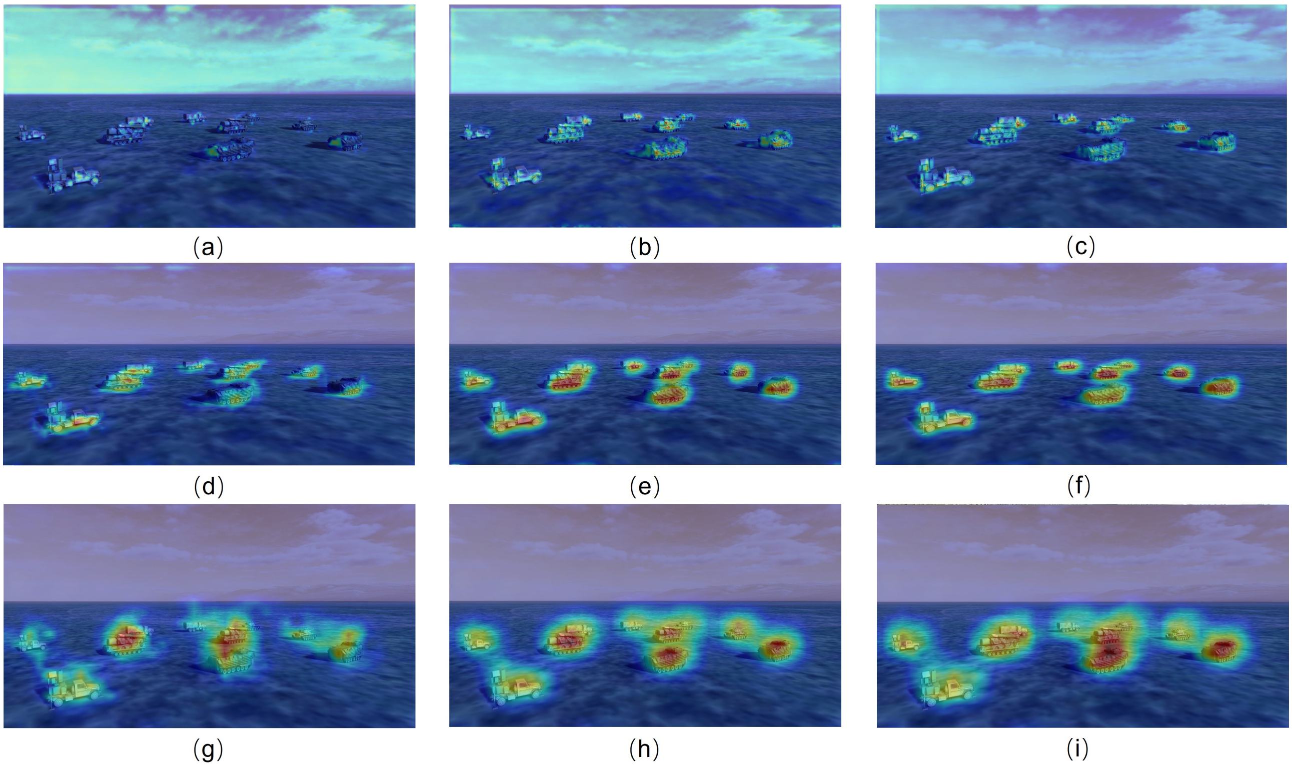

**Figure 12.** The comparative feature extraction results of different methods in the air-to-ground scenario. (**a**) CspDarkNet-53 stage1; (**b**) CSWin Transformer-small stage1; (**c**) proposed hybrid method stage1; (**d**) CspDarkNet-53 stage2; (**e**) CSWin Transformer-small; (**f**) proposed hybrid method stage1; (**g**) CspDarkNet-53 stage3; (**h**) CSWin Transformer-small; (**i**) proposed hybrid method stage1.

Moreover, in order to test the difficulty of recognizing different military targets, we conducted comparative experiments across eight target categories in air-to-ground and air-to-sea scenarios. The corresponding experimental results are presented in Table 4. In the air-to-ground scenario, the detectors consistently exhibited superior performance for Radar Vehicle and Rocket Launcher targets, with the proposed method proving to be the most effective, achieving accuracy of 73.7 and 73.3, respectively. We hypothesized that both target types were more distinctive, where the Radar Vehicle featured a radar unit at the rear, while the Rocket Launcher presented a tubular launcher at the top. This distinctiveness likely contributed to the enhanced recognition performance. The detectors exhibited a notable challenge in recognizing Armored Vehicles, particularly evident in the case of EfficientNet-b0, which achieves 50.4 AP. We thought this difficulty resulted from the relatively fewer external contour features, contributing to stealthiness. In the air-to-sea scenario, the detectors excelled in recognizing large warships, and we achieved optimal performance with mAP 74.6. This was attributed to the aircraft landing decks and bridges on aircraft carriers. However, the model demonstrated relatively lower effectiveness in recognizing small warships, which occupied a limited pixel area and exhibited fewer features.

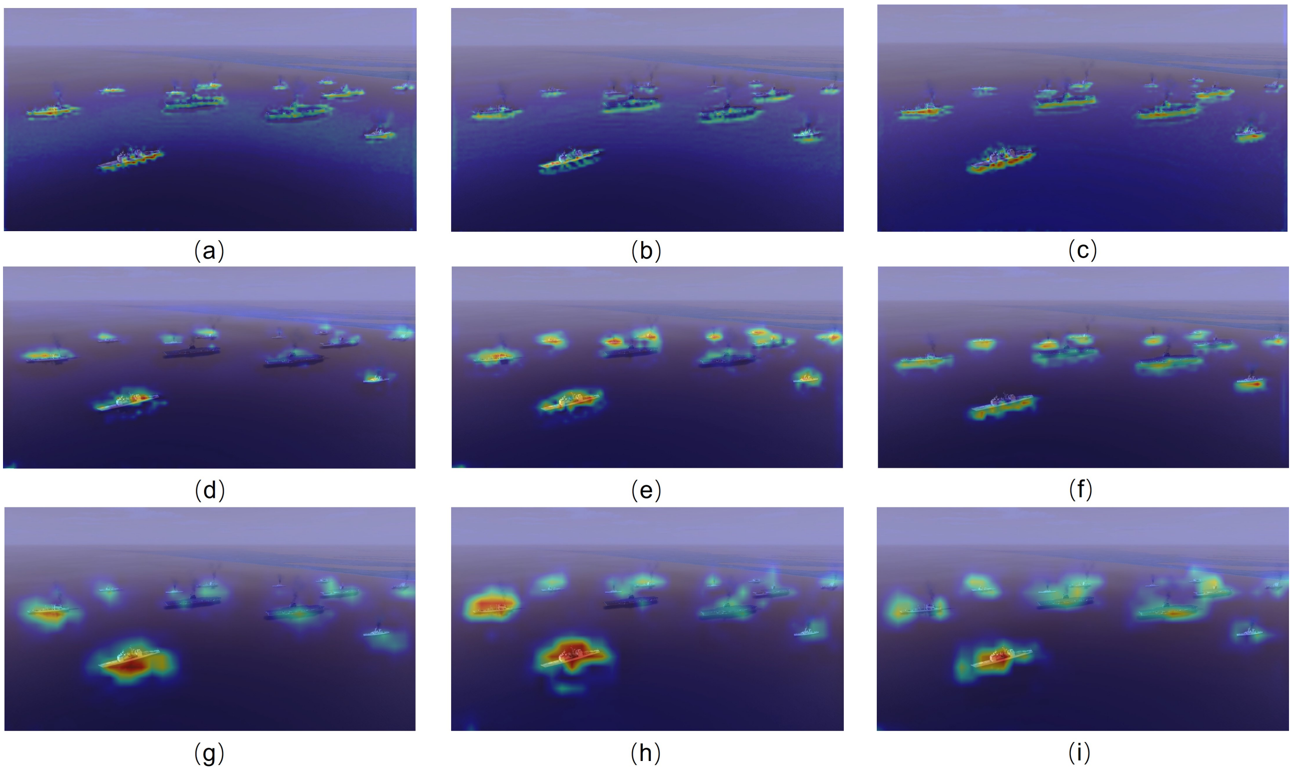

**Figure 13.** The comparative feature extraction results of different methods in the air-to-sea scenario. (**a**) CspDarkNet-53 stage1; (**b**) CSWin Transformer-small stage1; (**c**) proposed hybrid method stage1; (**d**) CspDarkNet-53 stage2; (**e**) CSWin Transformer-small; (**f**) proposed hybrid method stage1; (**g**) CspDarkNet-53 stage3; (**h**) CSWin Transformer-small; (**i**) proposed hybrid method stage1.

**Table 4.** Comparative experiments for different targets in air-to-sea and air-to-ground scenarios.

| Method | Scenario | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Air-to-Sea | | | Air-to-Ground | | | | |
| | Large Warship | Medium Warship | Small Warship | Transport Vehicle | Tank | Rocket Launcher | Armored Vehicle | Radar Vehicle |
| EfficientNet-b0 | 69.0 | 50.9 | 31.1 | 55.3 | 59.8 | 62.6 | 50.4 | 65.4 |
| ResNest-50 | 69.3 | 53.9 | 32.9 | 58.3 | 61.5 | 64.4 | 53.5 | 67.7 |
| CSPDarkNet-53 | 71.1 | 55.8 | 33.8 | 61.9 | 62.1 | 66.8 | 55.6 | 68.1 |
| ResNet-50 | 71.7 | 57.5 | 34.2 | 62.8 | 63.2 | 67.9 | 55.9 | 68.7 |
| PVT-s | 72.2 | 57.9 | 34.2 | 64.8 | 62.6 | 68.2 | 57.0 | 70.4 |
| Swin-s | 73.4 | 59.3 | 34.5 | 66.1 | 63.0 | 68.8 | 57.7 | 70.6 |
| CSWin-s | 73.3 | 60.1 | 34.6 | 67.2 | 65.8 | 69.3 | 58.6 | 71.5 |
| Ours | 74.6 | 62.9 | 36.1 | 68.7 | 67.8 | 73.7 | 60.8 | 73.3 |

### *5.3. Ablation Experiments*

We conducted ablation experiments to better identify the key components of the proposed detector. We first conducted branches ablation experiments to evaluate the contributions of the C-branch and T-branch in the model based on a RetinaNet baseline. Region–Pixel two-stage Transformer and MgpNet were employed as the backbone. The results are shown in Table 5.

We separately evaluated the model performance of the two branches. In the air-to-sea scenario, the AP of the C-branch and T-branch reached 55.4 and 56.5, respectively, while, in the air-to-ground scenario, the AP of the C-branch and T-branch reached 64.3 and 67.4, respectively. The results of the T-branch in both scenarios are superior to those of the C-branch, indicating that the Transformer architecture has an advantage in such military target detection tasks. The proposed method achieved AP scores of 57.9 and 68.9, respectively, which were higher than single-branch ones. It proved the effectiveness and necessity of the proposed fusion method, especially for detecting small objects.

**Table 5.** Branches ablation experiments in air-to-sea and air-to-ground scenarios.

| Scenario | Branch | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|
| Air-to-sea | C | 55.4 | 92.2 | 56.5 | 14.1 | 47.9 | 69.5 |
| | T | 56.5 | 93.6 | 58.6 | 15.1 | 49.2 | 69.9 |
| | ours | 57.9 | 95.1 | 60.2 | 17.1 | 50.3 | 70.6 |
| Air-to-ground | C | 64.3 | 88.7 | 72.2 | 11.4 | 62.7 | 85.8 |
| | T | 67.4 | 90.8 | 75.3 | 15.8 | 65.9 | 86.3 |
| | ours | 68.9 | 92.5 | 76.9 | 18.1 | 67.6 | 87.4 |

Furthermore, we conducted feature fusion methods' ablation experiments. We evaluated strategies for feature map fusion, including add, concatenation, and cross-concatenation fusing operation. The experimental results are detailed in Table 6. It revealed that fusion methods such as addition and concatenation yielded improvements compared to a single-branch network. The proposed fusion method demonstrated notable performance with mAP 57.9 and 68.9 in air-to-sea and air-to-ground scenarios, respectively. Notably, for small-object detection, our method exhibited improvements of 1.5 and 1.1 in the air-to-sea scenario compared to the addition and concatenation methods. In the air-to-ground scenario, enhancements of 1.4 and 1.1 were observed.

Additionally, the single Cross-Concatenation Interaction method performed poorly on most metrics because the cross operation, without further feature fusion, disrupted the channel-wise connectivity of adjacent feature maps. To verify this hypothesis, we added an extra convolutional layer to the proposed fusion strategy, reducing the doubled channel count by half. As a result, the detection performance noticeably declined. This validated the viewpoint [60] of avoiding dimensionality reduction and ensuring appropriate Cross-Channel Interaction to learn effective channel attention.

**Table 6.** Feature fusion methods' ablation experiments in air-to-sea and air-to-ground scenarios.

| Scenario | Method | | | | | | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Add | Concat | CCI [1] | GCE [1] | CCHI [1] | Conv2d | | | | | | |
| Air-to-sea | ✓ | / | / | / | / | / | 56.8 | 94.0 | 58.9 | 15.6 | 49.8 | 70.6 |
| | / | ✓ | / | / | / | / | 57.0 | 94.3 | 59.3 | 16.0 | 49.4 | 70.7 |
| | / | / | ✓ | / | / | / | 55.5 | 92.7 | 57.3 | 14.3 | 48.3 | 69.3 |
| | / | / | ✓ | ✓ | / | / | 58.9 | 96.1 | 60.6 | 17.9 | 49.9 | 71.2 |
| | / | / | ✓ | ✓ | ✓ | ✓ | 57.2 | 94.5 | 59.5 | 16.4 | 49.9 | 70.5 |
| | / | / | ✓ | ✓ | ✓ | / | 57.9 | 95.1 | 60.2 | 17.1 | 50.3 | 70.6 |
| Air-to-ground | ✓ | / | / | / | / | / | 67.8 | 91.1 | 75.6 | 16.7 | 66.5 | 87.7 |
| | / | ✓ | / | / | / | / | 68.1 | 91.5 | 76.0 | 17.0 | 66.8 | 87.7 |
| | / | / | ✓ | / | / | / | 67.1 | 89.9 | 75.2 | 14.4 | 67.1 | 87.4 |
| | / | / | ✓ | ✓ | / | / | 68.3 | 91.9 | 76.3 | 17.7 | 67.0 | 87.2 |
| | / | / | ✓ | ✓ | ✓ | ✓ | 51.3 | 79.6 | 57.7 | 10.6 | 43.1 | 75.2 |
| | / | / | ✓ | ✓ | ✓ | / | 68.9 | 92.5 | 76.9 | 18.1 | 67.6 | 87.4 |

[1] CCI, GCE, and CCHI mean Cross-Concatenation Interaction, Global Context Embedding, and Cross-Channel Interaction, respectively.

## 6. Conclusions

This paper presented a hybrid detection model that combined CNN and Transformer architecture to detect military targets from UAVs' oblique perspective. The proposed detector combined the C-branch Multi-gradient Path Network and the T-branch RPFormer in a parallel decoupled manner. A feature fusion strategy was proposed to integrate the feature maps of the two branches in three steps: Cross-Concatenation Interaction, Global Context Embedding, and Cross-Channel Interaction. Because the existing remote sensing datasets were mostly collected in a top-down view, which did not correspond to the actual UAV reconnaissance mission scenes, we constructed a dataset consisting of air-to-ground and air-to-sea scenarios from a large oblique perspective to evaluate the effectiveness of the proposed approaches. We validated different fusion methods, including add and concatenation operations, and demonstrated the effectiveness of the proposed fusion strategy in ablation experiments. In comparison experiments, the proposed method achieved mAP values of 57.9 and 68.9, improvements of 1.9 and 2.4 in air-to-ground and air-to-sea scenarios, respectively, which surpassed most current detection methods.

This research contributes to UAV detection for military targets in challenging scenarios. However, our research still has shortcomings. Although we have carefully designed the dataset for military targets, there is still a gap regarding actual perception reconnaissance scenarios, such as low light, smoke, and flame environments. The detection for small targets has achieved improvements, but it is still easy to miss and wrongly detect the targets compared with medium and large targets. In future work, we will continue to focus on these problems.

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

**Funding:** This research was funded by the Young Scientists Fund of the National Natural Science Foundation of China OF FUNDER grant number No. 52302506.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** We have chosen to disclose a portion of the test dataset for security reasons and it is available from the first author on reasonable request. While we recognize the importance of data sharing, we hope readers understand our decision and appreciate their understanding and support of our research.

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

## References

1. Peng, H.; Zhang, Y.; Yang, S.; Song, B. Battlefield image situational awareness application based on deep learning. *IEEE Intell. Syst.* **2019**, *35*, 36–43. [CrossRef]
2. Yang, K.; Pan, A.; Yang, Y.; Zhang, S.; Ong, S.H.; Tang, H. Remote sensing image registration using multiple image features. *Remote Sens.* **2017**, *9*, 581. [CrossRef]
3. Zhou, L.; Leng, S.; Liu, Q.; Wang, Q. Intelligent UAV swarm cooperation for multiple targets tracking. *IEEE Internet Things J.* **2021**, *9*, 743–754. [CrossRef]
4. Mei, C.; Fan, Z.; Zhu, Q.; Yang, P.; Hou, Z.; Jin, H. A Novel scene matching navigation system for UAVs based on vision/inertial fusion. *IEEE Sens. J.* **2023**, *23*, 6192–6203. [CrossRef]
5. Fang, W.; Love, P.E.; Luo, H.; Ding, L. Computer vision for behaviour-based safety in construction: A review and future directions. *Adv. Eng. Inform.* **2020**, *43*, 100980. [CrossRef]
6. Stodola, P.; Kozůbek, J.; Drozd, J. Using unmanned aerial systems in military operations for autonomous reconnaissance. In Proceedings of the Modelling and Simulation for Autonomous Systems: 5th International Conference, MESAS 2018, Prague, Czech Republic, 17–19 October 2018; Revised Selected Papers 5; Springer: Cham, Switzerland , 2019; pp. 514–529.
7. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
8. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef]

9. Sumari, A.D.W.; Pranata, A.S.; Mashudi, I.A.; Syamsiana, I.N.; Sereati, C.O. Automatic target recognition and identification for military ground-to-air observation tasks using support vector machine and information fusion. In Proceedings of the 2022 International Conference on ICT for Smart Society (ICISS), Bandung, Indonesia, 10–11 August 2022; pp. 1–8.
10. Du, X.; Song, L.; Lv, Y.; Qiu, S. A lightweight military target detection algorithm based on improved YOLOv5. *Electronics* **2022**, *11*, 3263. [CrossRef]
11. Jafarzadeh, P.; Zelioli, L.; Farahnakian, F.; Nevalainen, P.; Heikkonen, J.; Hemminki, P.; Andersson, C. Real-Time Military Tank Detection Using YOLOv5 Implemented on Raspberry Pi. In Proceedings of the 2023 4th International Conference on Artificial Intelligence, Robotics and Control (AIRC), Cairo, Egypt, 9–11 May 2023; pp. 20–26.
12. Jacob, S.; Wall, J.; Sharif, M.S. Analysis of Deep Neural Networks for Military Target Classification using Synthetic Aperture Radar Images. In Proceedings of the 2023 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sakheer, Bahrain, 20–21 November 2023; pp. 227–233.
13. Yu, B.; Lv, M. Improved YOLOv3 algorithm and its application in military target detection. *Acta Armamentarii* **2022**, *43*, 345.
14. Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. Cvt: Introducing convolutions to vision transformers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 22–31.
15. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
16. 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, 26 June–1 July 2016; pp. 779–788.
17. Redmon, J.; Farhadi, A. Yolov2:yolo9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
18. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
20. Li, C.; Li, L.; Jiang, H.; Weng, K.; Geng, Y.; Li, L.; Ke, Z.; Li, Q.; Cheng, M.; Nie, W.; et al. YOLOv6: A single-stage object detection framework for industrial applications. *arXiv* **2022**, arXiv:2209.02976.
21. 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. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 7464–7475.
22. 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: Cham, Switzerland , 2016; pp. 21–37.
23. 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.
24. 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: Cham, Switzerland, 2020; pp. 213–229.
25. Zhu, X.; Su, W.; Lu, L.; Li, B.; Wang, X.; Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv* **2020**, arXiv:2010.04159.
26. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 10012–10022.
27. Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; Guo, B. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12124–12134.
28. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 568–578.
29. Srinivas, A.; Lin, T.Y.; Parmar, N.; Shlens, J.; Abbeel, P.; Vaswani, A. Bottleneck transformers for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 16519–16529.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
31. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing network design strategies through gradient path analysis. *arXiv* **2022**, arXiv:2211.04800.
32. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
33. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
34. Wang, Y.; Ning, X.; Leng, B.; Fu, H. Ship detection based on deep learning. In Proceedings of the 2019 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 4–7 August 2019; pp. 275–279.
35. Xiong, Z.; Wang, L.; Zhao, Y.; Lan, Y. Precision Detection of Dense Litchi Fruit in UAV Images Based on Improved YOLOv5 Model. *Remote Sens.* **2023**, *15*, 4017. [CrossRef]

36. Hou, H.; Chen, M.; Tie, Y.; Li, W. A universal landslide detection method in optical remote sensing images based on improved YOLOX. *Remote Sens.* **2022**, *14*, 4939. [CrossRef]
37. Dai, Z.; Cai, B.; Lin, Y.; Chen, J. Up-detr: Unsupervised pre-training for object detection with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 1601–1610.
38. Liu, S.; Li, F.; Zhang, H.; Yang, X.; Qi, X.; Su, H.; Zhu, J.; Zhang, L. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv* **2022**, arXiv:2201.12329.
39. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
40. Wang, W.; Chen, W.; Qiu, Q.; Chen, L.; Wu, B.; Lin, B.; He, X.; Liu, W. Crossformer++: A versatile vision transformer hinging on cross-scale attention. *arXiv* **2023**, arXiv:2303.06908.
41. Zhao, T.; Cao, J.; Hao, Q.; Bao, C.; Shi, M. Res-SwinTransformer with Local Contrast Attention for Infrared Small Target Detection. *Remote Sens.* **2023**, *15*, 4387. [CrossRef]
42. Xu, Y.; Wu, Z.; Wei, Z. Spectral-Spatial Classification of Hyperspectral Image Based on Low-Rank Decomposition. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 2370–2380. [CrossRef]
43. Lu, W.; Lan, C.; Niu, C.; Liu, W.; Lyu, L.; Shi, Q.; Wang, S. A CNN-Transformer Hybrid Model Based on CSWin Transformer for UAV Image Object Detection. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2023**, *16*, 1211–1231. [CrossRef]
44. Zhao, X.; Xia, Y.; Zhang, W.; Zheng, C.; Zhang, Z. YOLO-ViT-Based Method for Unmanned Aerial Vehicle Infrared Vehicle Target Detection. *Remote Sens.* **2023**, *15*, 3778. [CrossRef]
45. Ren, K.; Chen, X.; Wang, Z.; Liang, X.; Chen, Z.; Miao, X. HAM-Transformer: A Hybrid Adaptive Multi-Scaled Transformer Net for Remote Sensing in Complex Scenes. *Remote Sens.* **2023**, *15*, 4817. [CrossRef]
46. Ye, T.; Zhang, J.; Li, Y.; Zhang, X.; Zhao, Z.; Li, Z. CT-Net: An efficient network for low-altitude object detection based on convolution and transformer. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
47. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30* .
48. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
49. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
50. Ba, J.L.; Kiros, J.R.; Hinton, G.E. Layer normalization. *arXiv* **2016**, arXiv:1607.06450.
51. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8697–8710.
52. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 6105–6114.
53. Yang, Z.; Zhu, L.; Wu, Y.; Yang, Y. Gated channel transformation for visual recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11794–11803.
54. Li, K.; Wan, G.; Cheng, G.; Meng, L.; Han, J. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS J. Photogramm. Remote Sens.* **2020**, *159*, 296–307. [CrossRef]
55. Su, H.; Wei, S.; Yan, M.; Wang, C.; Shi, J.; Zhang, X. Object detection and instance segmentation in remote sensing imagery based on precise mask R-CNN. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 1454–1457.
56. 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 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3974–3983.
57. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1026–1034.
58. Zhang, H.; Wu, C.; Zhang, Z.; Zhu, Y.; Lin, H.; Zhang, Z.; Sun, Y.; He, T.; Mueller, J.; Manmatha, R.; et al. Resnest: Split-attention networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 2736–2746.
59. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 618–626.
60. Wang, Q.; Wu, B.; Zhu, P.; Li, P.; Zuo, W.; Hu, Q. ECA-Net: Efficient channel attention for deep convolutional neural networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11534–11542.

**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.