remote sensing

MDPI

Article

HAM-Transformer: A Hybrid Adaptive Multi-Scaled Transformer Net for Remote Sensing in Complex Scenes

Keying Ren [1], Xiaoyan Chen [1,*], Zichen Wang [2], Xiwen Liang [1], Zhihui Chen [1] and Xia Miao [1]

[1] College of Electronic Information and Automation, Tianjin University of Science and Technology, Tianjin 300222, China; renkeying@mail.tust.edu.cn (K.R.); 21807914@mail.tust.edu.cn (X.L.); tustczh@mail.tust.edu.cn (Z.C.); miaoxia@mail.tust.edu.cn (X.M.)

[2] School of Electronic & Information Engineering, Tiangong Univeristy, Tianjin 430070, China; wangzc@mail.tust.edu.cn

* Correspondence: cxywxr@tust.edu.cn

Abstract: The quality of remote sensing images has been greatly improved by the rapid improvement of unmanned aerial vehicles (UAVs), which has made it possible to detect small objects in the most complex scenes. Recently, learning-based object detection has been introduced and has gained popularity in remote sensing image processing. To improve the detection accuracy of small, weak objects in complex scenes, this work proposes a novel hybrid backbone composed of a convolutional neural network and an adaptive multi-scaled transformer, referred to as HAM-Transformer Net. HAM-Transformer Net firstly extracts the details of feature maps using convolutional local feature extraction blocks. Secondly, hierarchical information is extracted, using multi-scale location coding. Finally, an adaptive multi-scale transformer block is used to extract further features in different receptive fields and to fuse them adaptively. We implemented comparison experiments on a self-constructed dataset. The experiments proved that the method is a significant improvement over the state-of-the-art object detection algorithms. We also conducted a large number of comparative experiments in this work to demonstrate the effectiveness of this method.

Keywords: aerial images; object detection; hybrid structure; adaptive multi-scaled net; transformer

Citation: 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. https://doi.org/10.3390/rs15194817

Academic Editors: Syed Agha Hassnain Mohsan, Pascal Lorenz, Khaled Rabie, Muhammad Asghar Khan and Muhammad Shafiq

Received: 13 August 2023
Revised: 15 September 2023
Accepted: 22 September 2023
Published: 3 October 2023

1. Introduction

With advancements in science and technology, the quality of remote sensing images has significantly improved. Due to their low cost, small size, and flexibility, UAVs are increasingly utilized in various tasks, such as maritime searching and rescuing [1], parking vehicle searching, and specific person recognition [2,3]. However, UAV aerial images pose unique challenges, including shooting angles, numerous small and overlapping targets, and onerous manual recognition. To address these challenges, deep-learning-based object detection methods are widely used. While object detection algorithms have made significant progress in the fields of face and pedestrian detection, their application to UAV aerial photography is relatively limited, particularly for accurately detecting small targets. Hence, the aim of this study was to explore and develop a small object detection algorithm specifically tailored to UAV aerial images, highlighting its importance and serving as the motivation for this research [4,5].

Traditional object detection algorithms use sliding windows, of different sizes, to traverse the image [6,7]. This method is time-consuming and not robust; it has difficulty meeting the requirements of object detection in UAV images with complex scenes. In recent years, deep-learning-based methods, especially convolutional neural networks (CNNs), have achieved good results in several areas of computer vision research [8,9]. In object detection algorithms, there are two main categories: two-stage detection algorithms, such as R-CNN [10], Fast R-CNN [11], and Faster R-CNN [12], and single-stage detection algorithms, such as YOLO [13–21] and SSD [22]. The two-stage detection algorithm is

divided into two subtasks, which first determine the candidate regions of possible objects and then perform regression and classification for each candidate region. The two-stage detector has higher accuracy, but the detection speed is slower. The one-stage detection algorithm removes the step of identifying candidate frames, and it directly performs object classification and regression. Compared to the two-stage detection algorithm, the one-stage detection algorithm has a faster detection speed, but its detection accuracy is relatively low, especially in the UAV aerial image detection task.

In recent years, with the development of deep learning, convolutional neural networks have been dominant in the field of object detection. CSADet [23] uses deformable convolution to construct a context-aware block that can extract both local high-frequency information and global semantic features. The semantic and location information of feature maps at different scales is shared using a multi-scale feature optimization block. Parallel extended convolution, to learn the contextual information of different objects at multiple scales, is used by mSODANet [24]. It has been experimentally demonstrated that the introduced hierarchical extension network captures the semantic information in the images more effectively. Ref. [25] designed a backbone network based on cross-stage and residual segmentation attention (CSP-ResNeSt) and a multiscale bidirectional feature pyramid with a simple attention module (Bi-SimAM-FPN). Experiments showed that the network can improve the recognition accuracy of small targets in images. Ref. [26] constructed a feature enhancement module (RFA), which consists of a pooling layer of deformable regions of interest and location attention. The spatial information of small objects is enriched by fusing the region of interest features at different scales. Ref. [27] used dilated convolution to study the contextual information of small objects. A module that dilated ResNet (DRM) was proposed, which was highly adaptable to scale variations of small objects at low altitude. Ref. [28] introduced the DDMA module, which incorporates coordinate attention, channel attention, and space attention, into the neck of YOLOv5s. By integrating local and global features with the DDMA module, the problem of missing error detection for small targets is reduced. Ref. [29] enhanced modeling capability by integrating deformable convolution within the network. Ref. [30] designed a global context (GC) block that can efficiently model the global context. The network showed excellent performance in various recognition tasks.

Since the vision transformer (ViT) introduced the transformer to the field of computer vision, transformer architecture has been continuously optimized [31–34]. Ref. [35] proposed a general transformer structure, MaxViT. MaxViT is composed of local and global attention that can fuse local and global features at each stage of the network. The effectiveness of MaxViT has been demonstrated by a large number of ablation experiments. Ref. [36] proposed a novel contextual transformer (CoT) module. The CoT enhances the learning capability of the attention module by using contextual information between the key and value. Ref. [37] built a new backbone network, by using the multi-head self-attention module of pyramid pooling (pyramid pooling transformer, P2T), which can extract the context features of the network. Ref. [38] disintegrated the self-attention mechanism into horizontal and vertical, which can be computed in parallel in both directions. At the same time, local enhanced position coding (LePE) was introduced, and the experiment proved that the CSWin transformer has a good effect in the field of vision. Ref. [39] constructed a vision transformer that alternately stacks the scalable attention mechanism and the windowed self-attention mechanism. This structure allows the network to achieve a good balance between accuracy and speed. Ref. [40] used overlapping convolutions with different kernel sizes as patch embedding to obtain patches of the same length. These patches are passed to the transformer. Finally, the output features are aggregated to represent the features at different granularity.

Recent work has shown that combining a CNN and a transformer allows the network to take advantage of the strengths of both architectures. Ref. [41] proposed an efficient hybrid architecture, EdgeNeXt. In EdgeNeXt, a slice-depth-transposed attention module is introduced, which can split features into multiple groups and use depth convolution and channel self-attention to increase receptive field and fuse multi-scale information.

Ref. [42] used a bidirectional bridge to connect MobileNet to the transformer. This design can integrate the local features of a CNN and the global features of a transformer, which can achieve higher computational efficiency and stronger feature extraction ability. Ref. [43] have discovered the potential relationship between the CNN and transformer by analyzing their operation principles. A transformer and a CNN have been cleverly combined to design a hybrid architecture called ACmix. Ref. [44] designed a parallel structure using a transformer and deep convolution, which makes the channel dimension and spatial dimension complementary through the interaction structure. The combination of the two designs achieves a deep fusion of local and global features. Ref. [45] enhanced the global perception capability of the CNN by fusing the global information of the transformer. Experimental results show that the conformer outperforms the CNN and ViT architectures alone with the same number of parameters. Ref. [46] proposed a convolutional transformer block (CTB) and a convolutional multi-head self-attention block (CMHSA). This design can improve the algorithm's ability to recognize obscured objects by aggregating context information.

To achieve high accuracy and fast detection methods, this paper proposes a hybrid adaptive multi-scaled transformer network (HAM-Transformer Network) for UAV aerial photography, which consists of three basic blocks: the convolutional local feature extraction block (CLB), multi-scale position embedding block (MPE), and adaptive multi-scale transformer block (AMT). Specifically, we use the CLB to extract local texture features in the initial three stages of HAM-Transformer Net. The CLB borrows the overall architectural idea of efficient layer aggregation network (ELAN) [47] but differs from the ELAN in that we redesigned the basic blocks. The MPE introduces the idea of multi-scale feature fusion into the overlapping embedding module by stacking max pooling layers. The AMT merges adjacent embedding blocks by using deep convolution with different kernel sizes and uses multi-branch adaptive fusion to balance features at different scales. Experiments demonstrate that the HAM-Transformer network outperforms state-of-the-art target detection methods. With the same number of parameters, HAM-Transformer improves 4.1% mAP over YOLOv8-S and 5.9% mAP over YOLOv6-S on the remote sensing-UAV aerial photography dataset.

The contributions of this study can be summarized as follows:

- We propose three efficient blocks, namely a convolutional local feature extraction block, multi-scale position embedding block and adaptive multi-scale transformer block, which can be easily inserted into any network without adjusting the overall architecture.
- We designed a novel efficient feature extraction backbone network, HAM-Transformer, which cleverly fuses the CNN and transformer. It can adaptively adjust the feature contribution for different receptive fields in the last stage of the network.
- We have combined existing UAV aerial photography and remote sensing datasets to enrich the diversity of our datasets, which include urban, maritime, and natural landscapes.
- We have carried out extensive experimental validation, and the experimental results show that HAM-Transformer Net balances speed and accuracy and outperforms the existing single-stage object detection feature extraction backbone network with similar parameter quantity.

The rest of this article is structured as follows. We introduce the overall structure of HAM-Transformer Net and the details of each block in Section 2. In Section 3, we describe the dataset used in the experiments and the implementation of the comparisons. We discuss the methodology proposed in this paper in Section 4. We summarize this work in Section 5.

2. Methodology

In this section, we outline the design of the overall structure of HAM-Transformer Net. Then, the details of each component in HAM-Transformer Net are introduced, including the convolutional local feature extraction block (CLB), multi-scale position embedding block (MPE), and adaptive multi-scale transformer block (AMT). In addition, in order to

support object detection in different hardware environments, we designed specific models with different sizes.

2.1. Overview

The overall structure of HAM-Transformer Net is shown in Figure 1. HAM-Transformer Net adopts a hierarchical pyramid structure. The input image X is fed into the stem, during which time its spatial dimension is 2× downsampled. Stages 1, 2, and 3 play the role of refining local features in the whole backbone. And the downsampling operation is performed before each Stage. Stage 4 is the last stage of the backbone. This stage is able to extract multi-scale global features. Each of these stages is built from multiple foundation blocks. In the figure, we use N_1, N_2, ... to represent the number of foundation blocks. H, W, and C_n denote feature map length, width, and number of channels. Split indicates that the input channel is evenly sliced into two parts. Conv3 × 3 and Conv1 × 1 represent 3 × 3 and 1 × 1 size convolution operations, respectively. MaxPool2d denotes the maximum pooling operation. Layer normalization represents a classical normalization operation. The shuffle operation mixes the input channels. DWConv3 × 3 denotes deep convolution with kernel size of 3 × 3.

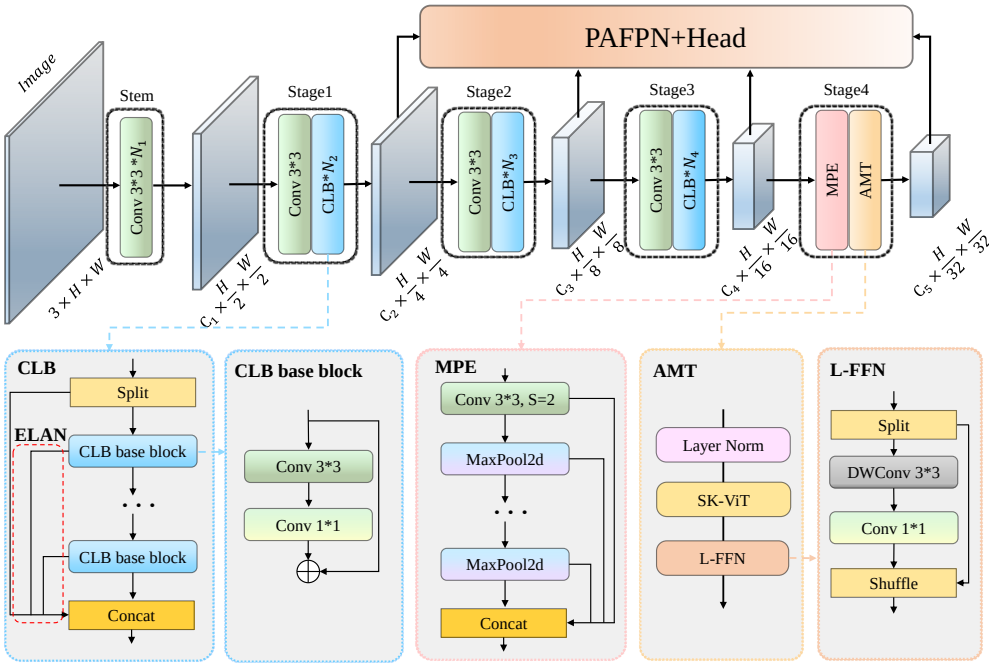


Figure 1. The overall architecture of HAM-Transformer Net. It consists of the convolutional local feature extraction block (CLB), multi-scale position embedding block (MPE) and adaptive multi-scale transformer block (AMT).

2.2. Convolutional Local Feature Extraction Block (CLB)

To compare with existing feature extraction methods, we first browsed some feature extraction blocks based on convolution, as shown in Figure 2. ResNet possesses a residual structure to solve the problem of difficult training [48]. MobileNetV2 is a lightweight structure due to an inverted residual structure [49]. Unlike ResNet, MobileNetV2 uses deep convolution to extract features and reduce the number of parameters. Although the above two structures have their own advantages, they still do not perform as well as vision transformers in the field of computer vision. ConvNeXt takes the advantages of transformer architecture and introduces the transformer idea into the ResNet structure [50]. At the macro level, ConvNeXt sets the convolutional kernel size to 7 × 7 and moves it up to the top of the residual block. At the micro level, the normalization layers and activation functions are replaced by layer normalization (LN) and GELU, which the authors call the

"modernization" of ResNet. The accuracy of ConvNeXt is higher than ResNet, but the large kernel convolution and LN are not suitable for mobile devices.

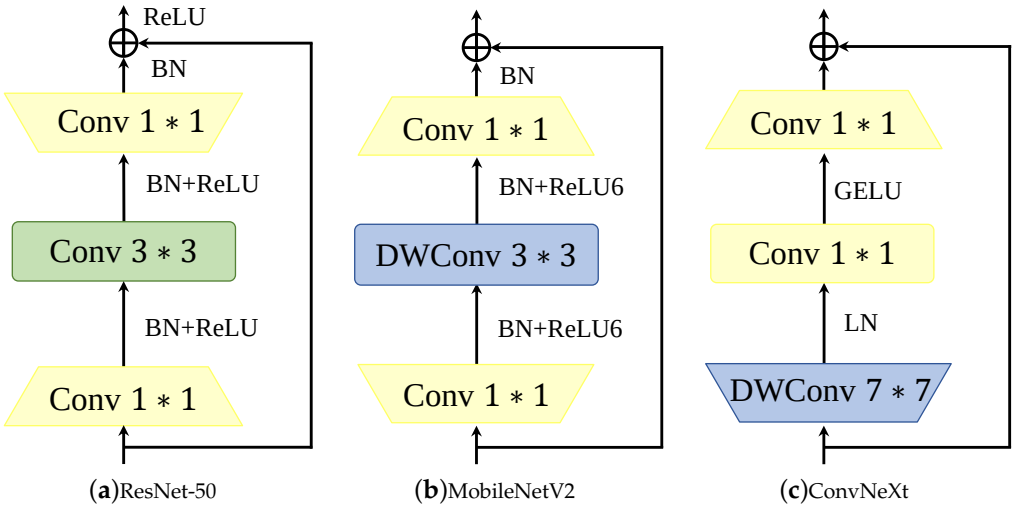


Figure 2. Comparison of different convolution-based blocks. BN denotes the batch normalization operation, and LN denotes the layer normalization operation. ReLU, ReLU6, and GELU denote several classical linear activation functions

The design process of the base CLB is similar to ConvNeXts, but we did not use the large kernel convolution, LN and GELU. We replaced the residual structure with the inverse residual structure, which is similar to MobileNetV2 but without deep convolution in the inverse residual structure. After the inverse residual structure, we proposed the 3×3 convolution and removed the last 1×1 convolution operation. The base CLB can be expressed as follows:

$$F(x) = X + W_{1\times1}(W_{3\times3}(X) \times e) \tag{1}$$

where $W_{1\times1}$ and $W_{3\times3}$ are the weights of 1×1 convolution as well as 3×3 convolution, e is the residual ratio, and X is the input feature.

The convolutional neural network includes feature aggregation path and gradient path design. By adjusting the gradient propagation path, different blocks can learn different information from a variety of points; at the same time, the information is passed to the next block and the final stage simultaneously. This gradient propagation method can make further use of the information of each module, thus greatly improving the utilization rate of network parameters. The features of different scales can be aggregated to improve the capability of feature expression at each stage. Between each CLB and concatenation, there the ELAN carries out skip connection, which makes the network more efficient in terms of parameter utilization. The ELAN is able to learn the important features efficiently by controlling the gradient path. In this study, we adopted this efficient architecture and combined it with our base CLB to form a local feature extraction module. The CLB can be expressed as follows:

$$x_{out} = cat(x_0, x_1, \cdots, x_n) \tag{2}$$

where x_{out} is the output feature $x \in \mathbb{R}^{c_{out}\times H\times W}$ of the CLB and $x_0, x_1, \cdots, x_n$ is the output feature of each base CLB.

2.3. Multi-Scale Position Embedding Block (MPE)

The original vision transformer encodes the position information of the embedded block by absolute or relative position. However, these two positional encodings have limitations. For example, absolute positional encoding adds a unique positional encoding to each token, which breaks the translational invariance of the transformer. Relative position encoding can increase the computational complexity and is unsuitable for self-attention

mechanisms. To avoid the above problems, we used convolutional position embedding (CPE) [51] for our study.

Multi-scale patch embedding [40] builds an embedding block that can operate in both fine-grained and coarse-grained systems. It achieves the same receptive fields as 3×3, 5×5, and 7×7 by stacking three 3×3 convolutions, and finally the embedded blocks with different scale sizes are fed into the self-attention module. We constructed a multi-scale position-encoding block. Similar to multi-scale patch embedding, in this stage we used overlapping convolutional positional embedding to encode the positional information of the input features. Unlike non-overlapping convolutional positional embedding, overlapping convolution maintains the image information continuity and avoids truncating the critical information in the image. We used zero padding to maintain the spatial size of the input feature maps. Specifically, given an input $x \in \mathbb{R}^{C \times H \times W}$, we performed zero padding of size $\frac{k-1}{2}$, where k is the kernel size. The final output size is $x \in \mathbb{R}^{2C \times \frac{H}{2} \times \frac{W}{2}}$.

We first performed a positional embedding operation on the input features using the overlapping convolution described above. After that, we extracted multi-scale feature information for the position embedding block. It is worth emphasizing that we adopted a different approach to the extraction of multi-granularity features than multi-scale patch embedding. We used a max pooling layer, which is more efficient than convolution, to aggregate features at different scales. To be specific, by stacking multiple max pooling layers with kernel size of 5×5, we can concatenate features from different scaled receptive fields and finally merge them with the original features to form a multi-path aggregation structure. Finally, the aggregated features are fed into the next block.

2.4. *Selective Kernel Transformer Block (SK-ViT)*

Inspired by the visual nerve in neuroscience, the receptive field of the visual nerve is not static but is modulated by the size of objects and the surrounding environment [52]. A dynamic selection mechanism is proposed in the visual transformer structure, which is rarely considered in most visual transformers. Specifically, we divided the self-attention heads into several groups and used different compression coefficients to compress the number of input embedding blocks, as shown in the bottom half of Figure 3. The picture on the right shows the number of embedded blocks (original embedded blocks) with a compression factor of $1\times$. And the left shows the number of embedded blocks with a compression factor of two which are merged with two neighboring embedded blocks. Finally, we used adaptive weights to adjust the contribution of different branches. In order to enable the network to adaptively adjust the size of the receptive field, we propose an automatic selection operation among different groups of self-attention heads. The operation includes grouping, selection, and fusion, as shown in Figure 3, where W_1 and W_2 are the weight coefficients of each branch. For simplicity, we only describe the case of two groups, but this structure can easily be extended to multiple groups.

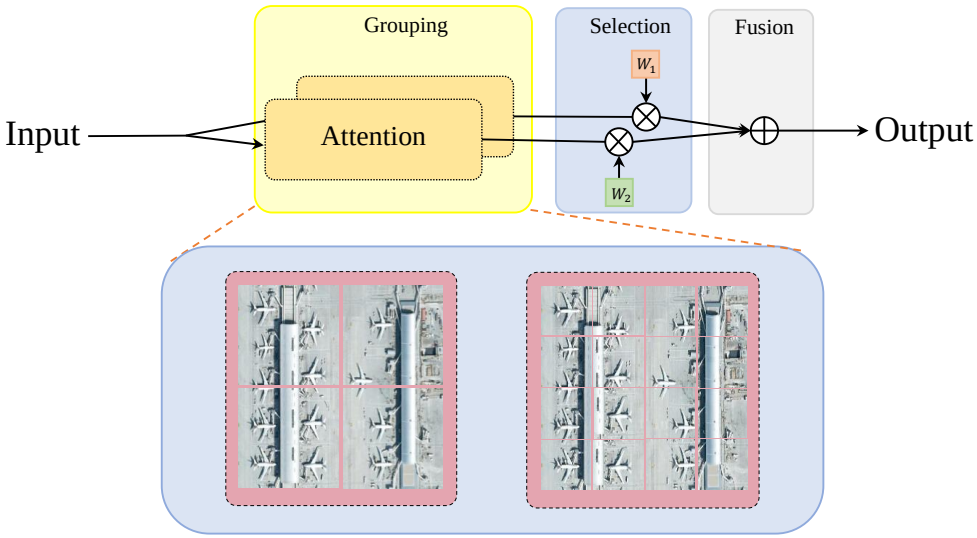


Figure 3. The overall architecture of the SK-ViT. The operation includes grouping, selection, and fusion. It divides the self-attention heads into several groups and uses different compression coefficients to compress the number of input embedding blocks.

2.4.1. Grouping

Figure 4 shows the attention of one of the branches in the grouping. where linear denotes the linear layer, matmul denotes matrix multiplication, and softmax denotes the softmax activation function. Q, K, and V stand for query, key, and value, respectively. For input $X \in \mathbb{R}^{N \times C}$, where $N = \frac{H \times W}{P}$ is the number of embedding blocks, we use a linear mapping to obtain the query Q. H and W are the spatial dimensions of the input features, and $P = h \times w$ is the spatial size of the embedded block. h and w are the height and width of the embedded block. To obtain different sizes of embedding blocks, we reshape the two-dimensional input features $X \in \mathbb{R}^{N \times C}$ into three-dimensional features $X_1 \in \mathbb{R}^{C \times H \times W}$ in the spatial dimension. Then, the spatial dimension is compressed using deep convolution with the compression coefficients $S_0, S_1, \cdots, S_n$. Specifically, we use kernel sizes and strides of $S_0, S_1, \cdots, S_n$ to achieve this operation, as shown in Figure 4. DWConv1 with kernel sizes 1×1 or 2×2 is illustrated as an example. Finally, the compressed feature x_1 is reshaped into a 2D feature map to obtain key K and value V. We perform self-attention calculation on the three mappings of Q, K, and V. To refine the final feature output, we introduced DWConv2 to refine the value of K to enhance the modeling capability of this block. This process can be expressed as follows:

$$\begin{aligned} Q &= Linear(X) \\ K, V &= Conv(X \rightarrow X_1) \\ Attention(Q, K, V) &= Softmax((\frac{QK^T}{\sqrt{d_k}})(V + Depthwise_{3\times3}(V)) \end{aligned} \tag{3}$$

where linear is a fully connected operation, $d_C = \frac{C}{k}$ denotes the channel dimension of a single head in the group, C stands for the dimension of the input channel, and k is the number of attention heads in the group.

The AMT can be easily expanded into multiple branches by expanding the number of groupings of attention heads. In order to be able to clearly express the overall idea of the method in this paper as well as to facilitate carrying out the validation of the method, the AMTs mentioned in the following are all two-branch examples.

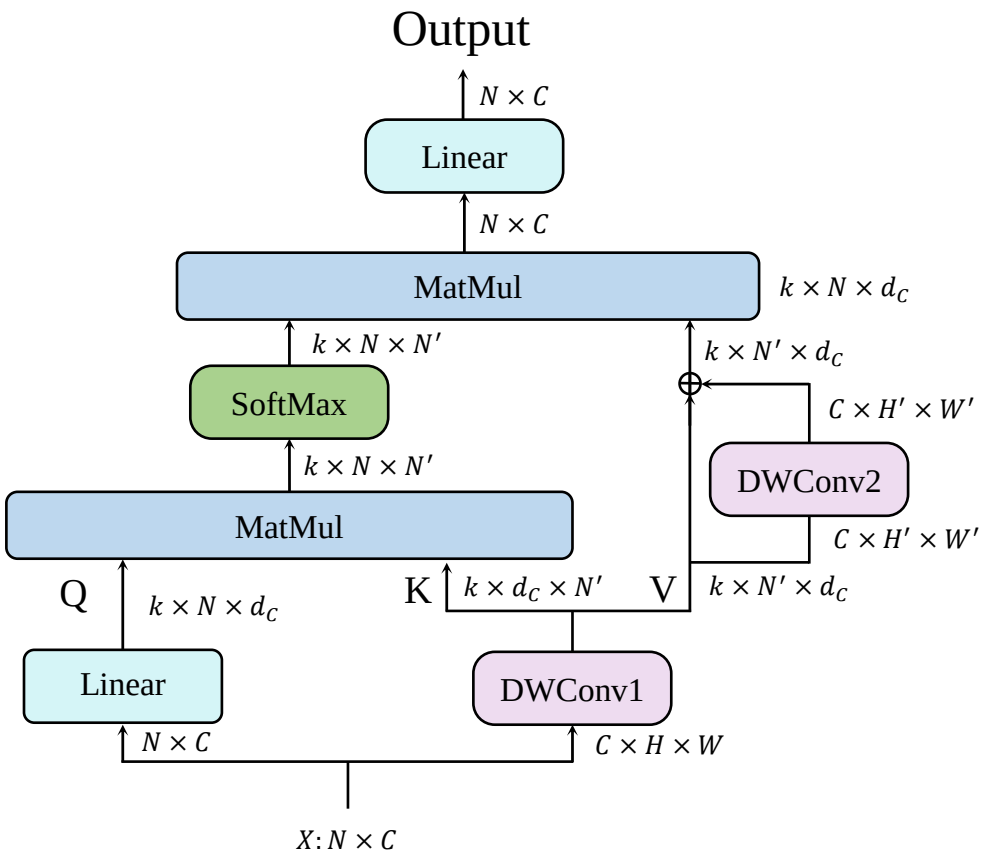


Figure 4. The overall architecture of attention computation in a single branch. DWConv1 can merge neighboring embedding blocks, and DWConv2 can refine the final feature output of K.

2.4.2. Selection

Weight coefficients were used to implement the selection of multiple receptive fields while discarding the complex attention mechanism. As the weight coefficients are adaptively adjusted during network training, there may be negative weights. To avoid this situation, we used RELU as the activation function with non-negativity. We propose a novel form-efficient normalized weight using adaptive weights for multi-scale selection.

The initial attention mechanism can be expressed as $Out = \sum_i w_i \cdot x_i$, where Out is the feature output after fusion; w_i is a learnable weight coefficient, which expresses the contribution of different perceptual fields; and x_i is the output of a certain scale. In order to reduce the computational cost, we used scalars as weights. However, since scalars are unbounded, weights with large differences can make network training difficult. To avoid this problem, we performed a normalization operation on the weights.

Softmax is a common normalization function that is widely used in deep learning. The contribution of each scale can be expressed by adjusting the weights to between 0 and 1, as shown in Equation (4).

$$Out = \sum_i \frac{e^{w_i}}{\sum_i e^{w_i}} \cdot x_i \tag{4}$$

However, softmax can affect the inference speed of the network, so we used a novel form-efficient normalized weight, as shown in Equation (5). Specifically, each branch weight is divided by the sum of the values of all weights to obtain the normalized value. To avoid the case of a zero denominator, we added correction terms to the denominator. Finally, the normalized weights were multiplied with each branch output feature to obtain the final output.

$$Out = \sum_i \frac{w_i \cdot x_i}{\sum_j w_j + \epsilon} \tag{5}$$

After comprehensive consideration, the output of selection can be can be expressed as follows:

$$Out = \sum_i \frac{RELU(w_i) \cdot x_i}{\sum_j RELU(w_j) + \epsilon} \tag{6}$$

where x_i is the output features of different branches, w_i is the weight coefficient, and ϵ is the correction term. *ReLU* was adopted to realize the normalization.

2.4.3. Fusion

The fusion of multi-scale information is achieved through two hybrid operations.

Concat: This fusion method is similar to the multi-level feature fusion in DenseNet. It can avoid information lossing of input, but compared to the addition operation, its computational cost is relatively high.

Add: Add operation superimposes the information of each channel without increasing the number of channels. This operation increases the proportion of beneficial information with less computational cost compared to concatenation.

2.5. Lightweight Feed-Forward Network (L-FFN)

In the standard vision transformer, the feed-forward network (FFN) consists of two fully connected layers with LN before the FFN [31]. To improve the network inference speed, we removed the layernorm before the FFN and replaced the fully connected layer with efficient deep convolution and point convolution.

We show three common feed-forward neural networks in Figure 5. The multi-head self-attention module extracts information in the spatial dimension, while the FFN extracts information between channels through a fully connected layer. Specifically, the standard FFN interacts within the channel dimension and does not operate in the spatial dimension. PVTv2 [51] and SSA [53] add a 3×3 deep convolution between two fully connected layers to refine spatial features. But this does not completely solve the problem of large parameter counts. The lightweight FFN(L-FFN) proposed in this work remedies this problem. We used deep convolution with a kernel size of 3×3 to extract information in the spatial dimensions and then used point convolution to enable the interaction of information between channels. This structure not only reduces the number of network parameters but also compensates for the disadvantage of the traditional FFN, which is insensitive to spatial dimensions. The specific computational process of the lightweight feed-forward neural network can be expressed as follows:

$$\begin{aligned} FFN_{out1} &= GeLU(BN(Depthwise_{3\times3}(X)) \\ FFN_{out2} &= GeLU(BN(Conv_{1\times1}(L - FFN_{out1})) \\ FFN_{out} &= X + LFFN_{out2} \end{aligned} \tag{7}$$

where $Depthwise_{3\times3}$ is the depth convolution with kernel size of 3×3, X is the input feature of FNN, FFN_{out1} is the spatial information output of *L-FFN*, and FFN_{out2} is the channel interaction output.

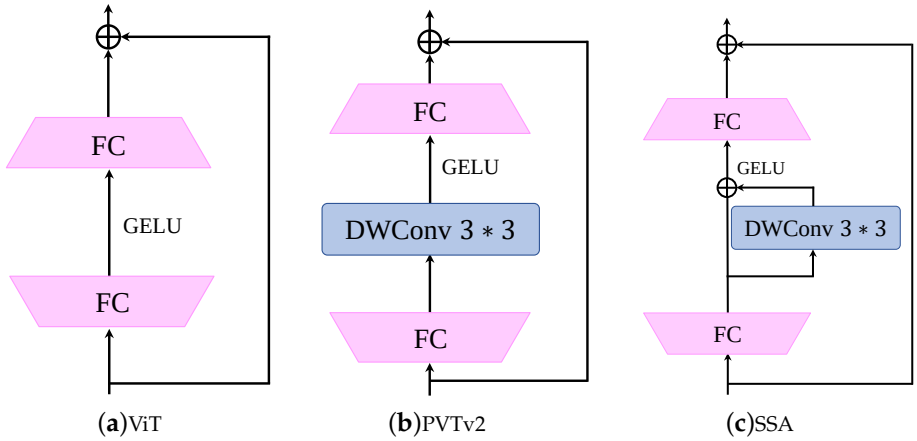


Figure 5. Comparison of different feed-forward networks. FC denotes fully connected layer.

2.6. HAM-Transformer Net Architecture

As with the existing backbone network, we constructed three variants of HAM-Transformer Net, namely HAM-Transformer Net-S/M/L. The architectural specification is shown in Table 1, where c represents the output channel, b represents the number of branches of the adaptive multi-scale transformer block, n represents the number of heads, and s is the compression factor of each branch. S, M, and L refer to three variants with small, medium, and large size, respectively.

Table 1. Configurations of HAM-Transformer Net-S/M/L.

Stage	Output Size	Layer Name	HAM-Transformer Net-S	HAM-Transformer Net-M	HAM-Transformer Net-L
Stem	$\frac{H}{2} \times \frac{W}{2}$	Convolution Layer	Conv3 × 3; c = 32; s = 2		Conv3 × 3; c = 48; s = 2
Stage 1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embedding	Conv3 × 3; c = 64; s = 2		Conv3 × 3; c = 96; s = 2
		CLB Block	[CLB base block × 1, 64] × 1	[CLB base block × 2, 64] × 2	[CLB base block × 3, 96] × 3
Stage 2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embedding	Conv3 × 3; c = 128; s = 2		Conv3 × 3; c = 192; s = 2
		CLB Block	[CLB base block × 2, 128] × 2	[CLB base block × 4, 128] × 4	[CLB base block × 6, 192] × 6
Stage 3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embedding	Conv3 × 3; c = 256; s = 2		Conv3 × 3; c = 384; s = 2
		CLB Block	[CLB base block × 2, 256] × 2	[CLB base block × 4, 256] × 4	[CLB base block × 6, 384] × 6
Stage 4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embedding	MPE, c = 512		MPE, c = 768
		AMT Block	$[AMT \times 1, b = 2, n = 1, s = (1,2)] \times 1$	$[AMT \times 2, b = 2, n = 2, s = (1,2)] \times 1$	$[AMT \times 4, b = 2, n = 8, s = (1,2)] \times 1$
Params			9.9 M	14.0 M	45.3 M

2.7. Dataset

We merged various scenarios from five publicly available datasets to expand the original VisDrone DET [54], namely Roundabout Aerial Images for Vehicle Detection [55], LEVIR [56], NWPU VHR-10 [57], UCAS-AOD [58], and UAVDT [59]. Our dataset has a total of 17,912 images and 12 categories, including pedestrians, people, bicycles, cars, minivans, trucks, tricycles, tricycles with canopies, buses, motorcycles, airplanes, tanks, and ships. The merged dataset has more categories and more complex scenes.

3. Experimental Results

3.1. Implementation

HAM-Transformer Net was trained using stochastic gradient descent (SGD) with the following parameter settings: momentum = 0.9 and weight decay = 5 $\times 10^{-4}$. The learning rate was adjusted using cosine annealing, and the initial learning rate was set to 0. During training, an exponential moving average (EMA) strategy with a decay of 0.9998 was used. The traditional data enhancement methods were adopted such as random cropping, random level flipping, color distortion, and multi-scale training. The rest of the models were kept with default parameter settings. To fairly compare the performance of

the models, no pre-training was carried out with the models. The images with resolution of 640 × 640 were for training and testing.

3.2. Comparisons with State-of-the-Art Models

The comparison with the state-of-the-art model is shown in Table 2, where param denotes the number of parameters of the method used. FLOPs denotes the number of floating point operations, which can be used to measure the complexity of the model. mAP denotes the mean average precision, which reflects the average precision of all classes. AP_{50} and AP_{75} represent the average precision of a model when the intersection over union (IoU) between the predicted bounding box and the ground truth bounding box is 50% and 75%, respectively. Latency represents the time to be consumed for detecting an image. As shown in Table 2, HAM-Transformer Net achieves the best balance between accuracy and FLOPs. Specifically, with a similar number of parameters, we compared HAM-Transformer Net with CNN- and transformer-based algorithms such as YOLOv8, YOLOv7, ConvNeXt, and the Swin transformer. Compared with the state-of-the-art CNN-based method, HAM-Transformer Net-S improves AP by 4.1% over YOLOv8-S with similar number of parameters. The AP improvement over YOLOv6-S is 5.9%, but the number of parameters is only half that of YOLOv6-S. Compared with transformer-based methods, HAM-Transformer Net-S has fewer parameters but higher accuracy. Specifically, HAM-Transformer Net-S has 3× fewer parameters but 2× improved accuracy compared to Swin-T. Compared with the recently proposed ConvNet, the accuracy is improved by nearly 15%. The accuracy rate of HAM-Transformer Net-M reached 37.6%, which improved by 0.4% compared with that of HAM-Transformer Net-S. Compared with HAM-Transformer Net-S, the accuracy of HAM-Transformer Net-L improved by 2.9%. These results show that the proposed HAM-Transformer Net is an effective network structure.

Table 2. Comparison with state-of-the-art models.

Methods	Param (M)	FLOPs (G)	mAP (%)	AP_{50}	AP_{75}	Latency (ms)
YOLOv5-S [16]	7.1	16.0	28.2	46.8	28.5	4.9
YOLOX-S [17]	8.9	26.8	30.3	50.3	31.7	6.3
YOLOR [18]	9.1	15.6	20.3	30.7	-	10.3
PP-YOLOE-S [19]	7.7	16.6	30.3	48.9	32.	6.7
YOLOv7-tiny [15]	6.1	13.3	30.0	51.5	-	4.8
YOLOv8-S	9.5	24.6	33.1	53.2	36.0	3.8
HAM-Transformer Net-S	**9.9**	**32.0**	**37.2**	**57.8**	40.3	4.5
YOLOv6-S [20]	18.5	45.3	31.3	50.9	33.2	3.6
YOLOF [21]	44	86	25.1	42.2	27.1	15.6
PVTv2 [32]	33.71	75.45	22.4	36.1	24.7	49.1
Deformable DETR [33]	40.51	79.19	27.1	46.9	27.9	50.7
DCNv2 [29]	42.06	80.34	22.1	35.9	24.5	37.1
Swin-T [34]	37.07	85.53	19.0	31.4	20.6	35.3
GCNet [30]	51.19	90.92	20.9	35.0	22.8	39.3
ConvNeXt [50]	66.74	126.41	23.0	36.7	24.8	46.7
HAM-Transformer Net-M	**14.0**	**40.0**	**37.6**	**58.6**	41.0	7.1
HAM-Transformer Net-L	**45.3**	**103.8**	**40.1**	**62.0**	43.9	15.1

3.3. Ablation Study and Visualization

In order to prove the effectiveness of our proposed blocks, ablation experiments were conducted. For the convenience of the experiments, HAM-transformer Net-S was adopted as the baseline.

3.3.1. Impact of Convolutional Local Feature Extraction Block

In order to prove the validity of the CLB proposed in this work, we replaced the CLB in HAM-Transformer Net with a classical bottleneck block structure, such as the bottleneck in ResNet-50, the inverted residual in MobileNetV2, the ConvNeXt block in ConvNeXt, and DarkNet53. As shown in Table 3, the CLB improves accuracy by 2.7% over ConvNeXt,

0.3% over inverted residual bottleneck blocks, and 0.6% over DarkNet53, with similar parameters and FLOPs. The experiment verified that the CLB is effective.

Table 3. Comparison of different convolution blocks.

Block	AP^{val}	Param (M)	FLOPs (G)
Residual block [48]	36.2	9.1	30.8
ConvNeXt [50]	34.4	9.9	33.8
Inverted residual block [49]	36.5	9.3	31.4
DarkNet53 block [60]	36.6	9.4	31.8
CLB (ours)	37.2	9.9	32.0

3.3.2. Impact of Efficient Layer Aggregation Network

To demonstrate the effectiveness of the ELAN in the CLB, we compared it with the classical path aggregation method, CSP, as shown in Table 4. From the experimental results, we can see that the ELAN is 0.4% higher than CSP. And the param FLOPs of both are similar. This also proves the effectiveness of the ELAN in this paper.

Table 4. Comparison of different layer aggregation networks.

Method	AP^{val}	Param (M)	FLOPs (G)
CSP [61]	36.8	9.8	31.2
ELAN (ours)	37.2	9.9	31.4

3.3.3. Impact of Multi-Scale Position Embedding Block

We further investigated the effect of multi-scale position embedding blocks on HAM-Transformer Net, as shown in Table 5, where we compared the CPE, overlapping convolutional position-encoding block (OPE), and the multi-scale position-encoding block (MPE) proposed in this work.

Table 5. Comparison of different position-encoding blocks.

Method	AP^{val}	Param (M)	FLOPs (G)
CPE [51]	36.3	8.6	30.9
OPE [32]	36.8	9.3	31.4
MPE (ours)	37.2	9.9	32.0

As the experimental results show, the CPE is less accurate than the other two position-coding methods, although the number of parameters is less. This also proves that the application of convolutional position coding in aerial remote sensing images is lacking. The multi-scale position-coding block achieves the optimal accuracy of these three position-coding blocks with almost the same FLOPs, which also proves the effectiveness of the MPE.

3.3.4. Impact of the Number of Attention Heads in the Branch

The number of attention heads is the key element of the AMT. To study its effect on the AMT, we conducted comparison experiments with two branches.

As shown in Table 6, we kept the number of attention heads the same for both branches. The best results are obtained when the number of attention heads is 1. As the number of attention heads increases, the accuracy decreases. Experiments show that the number of attention heads has an effect on the accuracy of the AMT. In the following ablation experiments, we set the number of heads to 1.

Table 6. Comparison of results of the number of attention heads.

Head	AP^{val}	Param (M)	FLOPs (G)
1	37.2	9.9	32.0
2	37.1	9.9	32.0
4	36.4	9.9	32.0
8	36.7	9.9	32.0

3.3.5. Impact of the Compression Factor of Each Branch

We also studied the effect of different combinations of compression coefficients on the AMT. In order to limit the search space, we set the compression coefficients to 1×, 2×, and 4×, and two branches were considered in the following experiments. As shown in Table 5, 1× represents that the compression factor is 1, which is the original feature dimension, and 2× represents that the number of embedded blocks is compressed to half size.

As can be seen from Table 7, the combination with compression coefficients of 1× and 2× is optimal in the search space, and the number of parameters and FLOPs are similar to other combinations.

Table 7. Comparison of results of the compression factor of each branch.

1×	2×	4×	AP^{val}	Param (M)	FLOPs (G)
	✓	✓	36.7	13.8	31.8
✓		✓	37.1	13.1	31.9
✓	✓		37.2	9.9	32.0

3.3.6. Impact of the Weight Generation Method

As described in Section 2.4.2, we analyzed different methods of weight generation and propose a efficient normalized weight generation method.

Table 8 shows the effects of different weight generation methods on the detection accuracy and speed. As the results show, the efficient normalized weight has higher accuracy and faster speed than softmax normalized weight.

Table 8. Comparison of efficient weight generation methods.

Method	AP^{val}	FLOPs (G)	Latency (ms)
Non-weight	35.8	32.0	4.5
Non-normalized weight	36.8	32.0	4.5
Softmax normalized weight	36.8	32.0	4.6
Efficient normalized weight	37.2	32.0	4.5

3.3.7. Impact of the Branch Fusion Method

As described in Section 2.4.3, two different fusion methods are proposed in this paper. These two methods are compared in this experiment.

It can be seen from Table 9 that the number of parameters and FLOPs of the fusion method using Cat are increased, but its accuracy is decreased by 0.7% compared to Add. This experiment verifies that the fusion method of Add is effective in the proposed network.

Table 9. Comparison of results of the branch fusion method.

Method	AP^{val}	Param (M)	FLOPs (G)
Cat	36.8	10.3	34.8
Add	37.2	9.9	32.0

3.3.8. Impact of the Lightweight Feed-Forward Network

To verify the impact of the L-FFN, we compared several FFN structures, such as the ViT, pyramid vision transformer (PVT), and shunted self-attention (SSA).

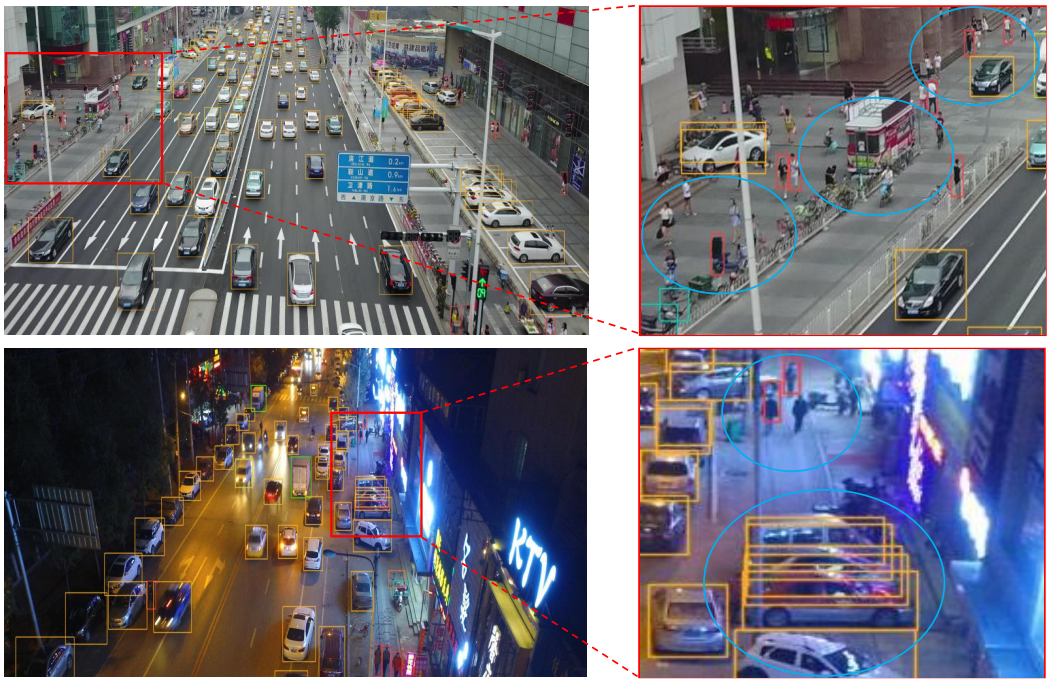
As can be seen from Table 10, the accuracy of the FFN in the ViT is nearly 1% lower than that of L-FFN, the accuracy of PVT and SSA is the same, the L-FFN accuracy reaches 37.2%, and the numbers of parameters and FLOPs are much lower than those of the remaining three FFNs. This further proves the effectiveness of our proposed L-FFN.

Table 10. Comparison of results of the lightweight feed-forward network.

Method	AP^{val}	Param (M)	FLOPs (G)
ViT [62]	36.3	11.9	33.6
PVTv2	36.5	12.0	33.6
SSA	36.5	12.0	33.6
L-FFN (ours)	37.2	9.9	32.0

3.3.9. Visualization

To verify the superiority of the HAM-Transformer network, we visualize the detection results and heat maps of HAM-Transformer-S and YOLOv8-S in Figure 6. The detection results show that YOLOv8-S has a good detection effect for objects at close distances but is less effective for overlapping objects and distant objects. HAM-Transformer-S has a better detection effect for overlapping objects and distant objects. This also intuitively demonstrates that our proposed HAM-Transformer-S can adaptively select different receptive fields as the size of the object changes.



(**a**)YOLOv8-S

Figure 6. *Cont.*

(**b**)HAM-Transformer Net-S

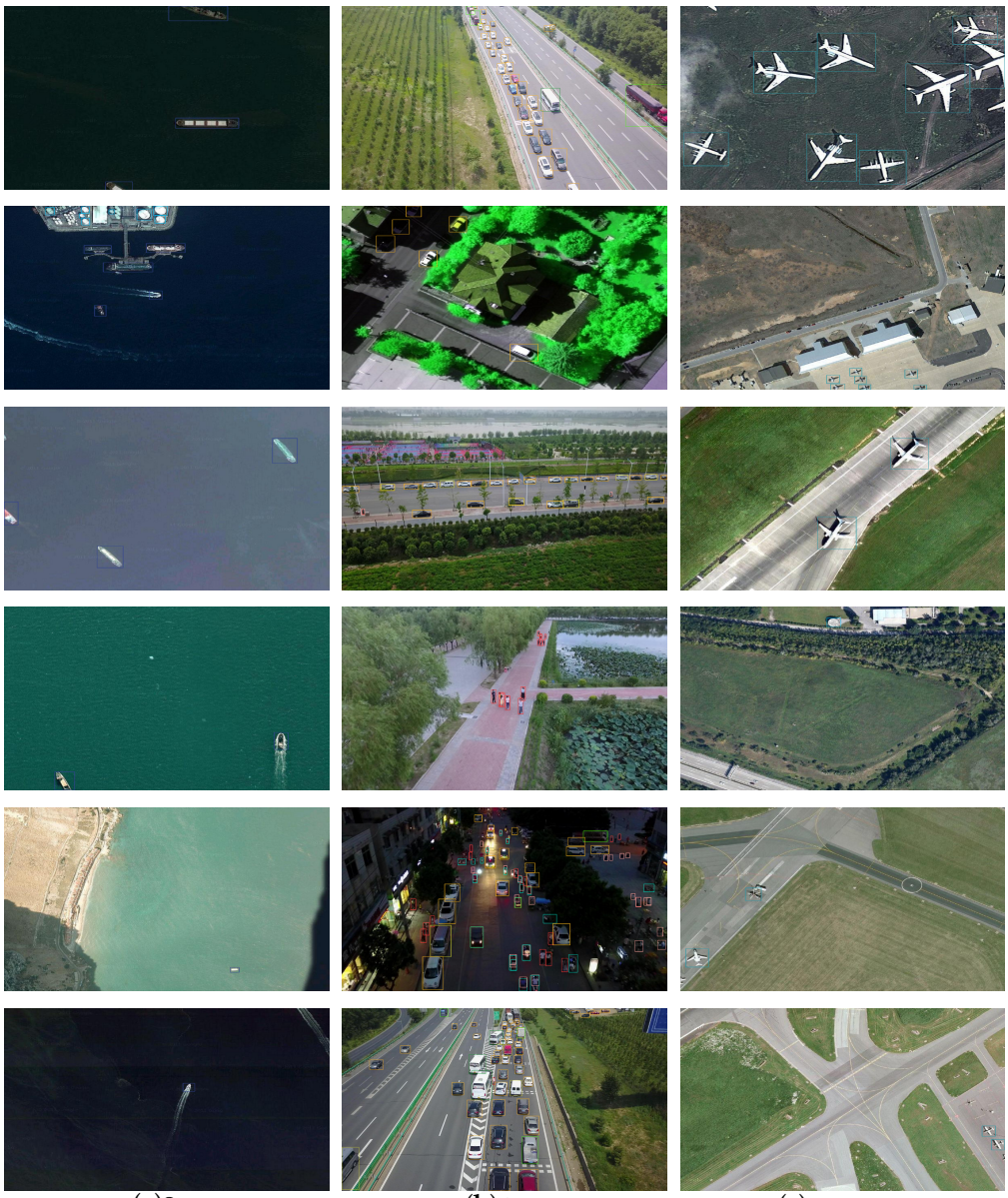
Figure 6. Object detection visualization with different algorithms.

Moreover, as shown in Figure 7, HAM-Transformer-S can focus on objects of different sizes compared to YOLOv8-S. And YOLOv8-S is insensitive to smaller targets. This proves that the SK-ViT has stronger feature aggregation ability.

To demonstrate the generalization ability of HAM-Transformer Net, we visualize different application scenarios in Figure 8, including oceans, cities, and mountains. In the figure, we can see that HAM-Transformer Net shows competitive detection ability in these complex and changing scenarios.

(**a**)Input (**b**)YOLOv8-S (**c**)HAM-Transformer-S

Figure 7. Attention visualization of different structures. To more intuitively verify the effectiveness of HAM-Transformer-S, we used GradCAM to visualize heat maps of network output features.



(**a**)Sea (**b**)City (**c**)Mountain

Figure 8. Visualization of object detection. We extracted some representative images from our dataset to demonstrate the performance of HAM-Transformer-S.

4. Discussion

UAV aerial images are affected by the shooting angle, resulting in a large number of small and overlapping targets. This brings great challenges to the object detection algorithm of UAV aerial images. To overcome the above challenges, we propose a novel object detection method called HAM-Transformer, which combines both a CNN and transformer. The method uses convolutional local feature extraction blocks to refine the information of feature maps and adopts adaptive multi-scale transformer blocks to adaptively fuse features for different receptive fields. Experimental results have shown that this method represents a great improvement compared to current advanced methods.

Traditional object detection methods require extensive manual feature design, which not only consumes time but also cannot guarantee robustness, which makes it difficult to meet the requirements of UAV image target detection in complex scenes. In the past two years, many scholars have combined transformer methods originating from natural language processing with CNNs and achieved a new level of performance. Different from

previous work, we propose a novel CNN–transformer feature extraction backbone network. As shown in Table 2, the proposed method in this paper exhibits a 4.1% improvement compared to YOLOv8-s and has similar speed. In addition, in order to prove the effectiveness of our proposed blocks, we conducted a large number of ablation experiments in experiments. As can be seen from Table 3, the CLB proposed by us has higher accuracy than other classical convolution modules. As can be seen from Table 7, the L-FFN proposed by us not only has fewer parameters but also higher precision than other methods.

Due to the limitations of the hardware environment, we limited the input format of the model to 640 × 640 pixels. This is unfavorable for large-sized aerial images. Our approach is based on images for object recognition, but for UAV remote sensing object recognition other forms of data are also crucial. Therefore, in the future, we will further investigate how to use the image form in conjunction with other forms of object detection data to compensate for the deficiencies in images.

In summary, in this study, we propose a novel hybrid feature extraction backbone network with the CNN–transformer method. After a large number of experiments, it was proved that the method proposed in this paper has better performance compared to other methods. HAM-Transformer can also be easily applied to other fields such as remote sensing object tracking and object segmentation.

5. Conclusions

In this study, we constructed a remote sensing image dataset with multiple devices and scenes and propose a CNN–transformer feature extraction backbone network, HAM-Transformer. HAM-Transformer first refines the texture information of feature maps using convolutional local feature extraction blocks. After that, multi-scale information is extracted using multi-scale location coding, and finally the adaptive multi-scale transformer block is used to extract features for different receptive fields and fuse them adaptively. The experiments prove that the method is a significant improvement over the state-of-the-art object detection algorithms. Although the experiment proves that our method is effective, there is still a long way to go for practical applications. Adapting to the hardware at the edge is one of the directions of our future research, along with balancing the algorithm performance and the number of parameters.

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

Funding: This work was supported by the Tianjin Research Innovation Project for Postgraduate Students under Grant No. KYS202108.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available as the final dataset used is private.

Acknowledgments: We thank AI Edge Computing Joint Laboratory for hardware support for this work.

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

References

1. Wang, Y.; Liu, W.; Liu, J.; Sun, C. Cooperative USV–UAV marine search and rescue with visual navigation and reinforcement learning-based control. *ISA Trans.* **2023**, *137*, 222–235. [CrossRef] [PubMed]
2. Li, R.; Yu, J.; Li, F.; Yang, R.; Wang, Y.; Peng, Z. Automatic bridge crack detection using Unmanned aerial vehicle and Faster R-CNN. *Constr. Build. Mater.* **2023**, *362*, 129659. [CrossRef]
3. Bouguettaya, A.; Zarzour, H.; Kechida, A.; Taberkit, A.M. A survey on deep learning-based identification of plant and crop diseases from UAV-based aerial images. *Clust. Comput.* **2023**, *26*, 1297–1317. [CrossRef]

4. Zou, Z.; Chen, K.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *Proc. IEEE* **2023**, *111*, 257–276. [CrossRef]
5. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikainen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
6. 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*; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 886–893. [CrossRef]
7. Felzenszwalb, P.; McAllester, D.; Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008*; IEEE: Piscataway, NJ, USA, 2008; pp. 1–8. [CrossRef]
8. Deng, S.; Xiong, Y.; Wang, M.; Xia, W.; Soatto, S. Harnessing unrecognizable faces for improving face recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 3424–3433. [CrossRef]
9. Liu, W.; Hasan, I.; Liao, S. Center and Scale Prediction: Anchor-free Approach for Pedestrian and Face Detection. *Pattern Recognit.* **2023**, *135*, 109071. [CrossRef]
10. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA , 23–28 June 2014; pp. 580–587. [CrossRef]
11. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [CrossRef]
12. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef] [PubMed]
13. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA , 27–30 June 2016; pp. 779–788. [CrossRef]
14. Redmon, J.; Farhadi, A. 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. [CrossRef]
15. 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, 17–24 June 2023; pp. 7464–7475.
16. Jocher, G. ultralytics/yolov5: v6.2. 2022. Available online: https://doi.org/10.5281/zenodo.7002879 (accessed on 17 August 2022). [CrossRef]
17. Ge, Z.; Liu, S.; Wang, F.; Li, Z.; Sun, J. Yolox: Exceeding yolo series in 2021. *arXiv* **2021**, arXiv:2107.08430. https://doi.org/10.48550/arXiv.2107.08430.
18. Wang, C.Y.; Yeh, I.H.; Liao, H.Y.M. You only learn one representation: Unified network for multiple tasks. *arXiv* **2021**, arXiv:2105.04206. https://doi.org/10.48550/arXiv.2105.04206.
19. 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. https://doi.org/10.48550/arXiv.2203.16250.
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. https://doi.org/10.48550/arXiv.2209.02976.
21. Chen, Q.; Wang, Y.; Yang, T.; Zhang, X.; Cheng, J.; Sun, J. You only look one-level feature. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13039–13048. [CrossRef]
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*; Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37. [CrossRef]
23. Han, W.; Li, J.; Wang, S.; Wang, Y.; Yan, J.; Fan, R.; Zhang, X.; Wang, L. A context-scale-aware detector and a new benchmark for remote sensing small weak object detection in unmanned aerial vehicle images. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102966. [CrossRef]
24. Chalavadi, V.; Jeripothula, P.; Datla, R.; Ch, S.B. mSODANet: A network for multi-scale object detection in aerial images using hierarchical dilated convolutions. *Pattern Recognit.* **2022**, *126*,108548. [CrossRef]
25. Hao, K.; Chen, G.; Zhao, L.; Li, Z.; Liu, Y.; Wang, C. An insulator defect detection model in aerial images based on Multiscale Feature Pyramid Network. *IEEE Trans. Instrum. Meas.* **2022**, *71*,3522412. [CrossRef]
26. Bai, Y.; Li, R.; Gou, S.; Zhang, C.; Chen, Y.; Zheng, Z. Cross-connected bidirectional pyramid network for infrared small-dim target detection. *IEEE Geosci. Remote. Sens. Lett.* **2022**, *19*, 7506405. [CrossRef]
27. Mittal, P.; Sharma, A.; Singh, R.; Dhull, V. Dilated convolution based RCNN using feature fusion for Low-Altitude aerial objects. *Expert Syst. Appl.* **2022**, *199*, 117106. [CrossRef]
28. Bao, W.; Zhu, Z.; Hu, G.; Zhou, X.; Zhang, D.; Yang, X. UAV remote sensing detection of tea leaf blight based on DDMA-YOLO. *Comput. Electron. Agric.* **2023**, *205*, 107637. [CrossRef]
29. Zhu, X.; Hu, H.; Lin, S.; Dai, J. Deformable convnets v2: More deformable, better results. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9308–9316. [CrossRef]

30. Cao, Y.; Xu, J.; Lin, S.; Wei, F.; Hu, H. Gcnet: Non-local networks meet squeeze-excitation networks and beyond. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Republic of Korea, 27–28 October 2019. [CrossRef]
31. 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*, 6000–6010.
32. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pvt v2: Improved baselines with pyramid vision transformer. *Comput. Vis. Media* **2022**, *8*, 415–424. [CrossRef]
33. 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 International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
34. 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, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022. [CrossRef]
35. Tu, Z.; Talebi, H.; Zhang, H.; Yang, F.; Milanfar, P.; Bovik, A.; Li, Y. Maxvit: Multi-axis vision transformer. In *Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Part XXIV; Springer: Berlin/Heidelberg, Germany, 2022; pp. 459–479. [CrossRef]
36. Li, Y.; Yao, T.; Pan, Y.; Mei, T. Contextual transformer networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 1489–1500. [CrossRef]
37. Wu, Y.H.; Liu, Y.; Zhan, X.; Cheng, M.M. P2T: Pyramid pooling transformer for scene understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef] [PubMed]
38. 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. [CrossRef]
39. Yang, R.; Ma, H.; Wu, J.; Tang, Y.; Xiao, X.; Zheng, M.; Li, X. Scalablevit: Rethinking the context-oriented generalization of vision transformer. In *Proceedings of the Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, 23–27 October 2022*; Part XXIV; Springer: Berlin/Heidelberg, Germany, 2022; pp. 480–496. [CrossRef]
40. Lee, Y.; Kim, J.; Willette, J.; Hwang, S.J. Mpvit: Multi-path vision transformer for dense prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2022; pp. 7287–7296. [CrossRef]
41. Maaz, M.; Shaker, A.; Cholakkal, H.; Khan, S.; Zamir, S.W.; Anwer, R.M.; Shahbaz Khan, F. Edgenext: efficiently amalgamated cnn-transformer architecture for mobile vision applications. In *Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October* 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 3–20. [CrossRef]
42. Chen, Y.; Dai, X.; Chen, D.; Liu, M.; Dong, X.; Yuan, L.; Liu, Z. Mobile-former: Bridging mobilenet and transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2022; pp. 5270–5279. [CrossRef]
43. Pan, X.; Ge, C.; Lu, R.; Song, S.; Chen, G.; Huang, Z.; Huang, G. On the integration of self-attention and convolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2022; pp. 815–825. [CrossRef]
44. Chen, Q.; Wu, Q.; Wang, J.; Hu, Q.; Hu, T.; Ding, E.; Cheng, J.; Wang, J. Mixformer: Mixing features across windows and dimensions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2022; pp. 5249–5259. [CrossRef]
45. Peng, Z.; Guo, Z.; Huang, W.; Wang, Y.; Xie, L.; Jiao, J.; Tian, Q.; Ye, Q. Conformer: Local features coupling global representations for recognition and detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 9454–9468. [CrossRef] [PubMed]
46. Ye, T.; Qin, W.; Zhao, Z.; Gao, X.; Deng, X.; Ouyang, Y. Real-Time Object Detection Network in UAV-Vision Based on CNN and Transformer. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2505713. [CrossRef]
47. Wang, C.Y.; Liao, H.Y.M.; Yeh, I.H. Designing Network Design Strategies Through Gradient Path Analysis. *arXiv* **2022**, arXiv:2211.04800. https://doi.org/10.48550/arXiv.2211.04800.
48. 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, 27–30 June 2016; pp. 770–778. [CrossRef]
49. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
50. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986. [CrossRef]
51. Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Shen, C. Conditional Positional Encodings for Vision Transformers. In Proceedings of the ICLR 2023, Kigali, Rwanda, 1–5 May 2023.
52. Hubel, D.H.; Wiesel, T.N. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.* **1962**, *160*, 106. [CrossRef] [PubMed]

53. Ren, S.; Zhou, D.; He, S.; Feng, J.; Wang, X. Shunted self-attention via multi-scale token aggregation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10853–10862. [CrossRef]
54. Zhu, P.; Wen, L.; Du, D.; Bian, X.; Fan, H.; Hu, Q.; Ling, H. Detection and Tracking Meet Drones Challenge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 7380–7399. [CrossRef] [PubMed]
55. Puertas, E.; De-Las-Heras, G.; Fernández-Andrés, J.; Sánchez-Soriano, J. Dataset: Roundabout Aerial Images for Vehicle Detection. *Data* **2022**, *7*, 47. [CrossRef]
56. Zou, Z.; Shi, Z. Random access memories: A new paradigm for target detection in high resolution aerial remote sensing images. *IEEE Trans. Image Process.* **2017**, *27*, 1100–1111. [CrossRef] [PubMed]
57. Cheng, G.; Zhou, P.; Han, J. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Trans. Geosci. Remote. Sens.* **2016**, *54*, 7405–7415. [CrossRef]
58. Zhu, H.; Chen, X.; Dai, W.; Fu, K.; Ye, Q.; Jiao, J. Orientation robust object detection in aerial images using deep convolutional neural network. In *Proceedings of the 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 27–30 September 2015*; IEEE: Piscataway, NJ, USA, 2015; pp. 3735–3739.
59. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 370–386. [CrossRef]
60. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767. https://doi.org/10.48550/arXiv.1804.02767.
61. 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 390–391.
62. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.a. An Image is Worth 16 $\times$ 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, 3–7 May 2021.