FPNFormer: Rethink the Method of Processing the Rotation-Invariance and Rotation-Equivariance on Arbitrary-Oriented Object Detection

Yang Tian, Mengmeng Zhang, Jinyu Li, Yangfan Li, Hong Yang, and Wei Li, *Senior Member, IEEE*

***Abstract*— Feature pyramid network transformer decoder (FPNFormer) module, which can effectively deal with the strong rotation arbitrary of remote sensing images while improving the expressiveness and robustness of the model. It is a plug-and-play module that can be well transferred to various detection models and significantly improves performance. Specifically, we use the computational method of transformer decoder to deal with the problem that the image has any orientation, and its output weakly depends on the order of the input data. We apply it to the feature fusion stage and design two ways top-down and down-top to fuse features of different scales, which enables the model to have a more vital ability to perceive objects at different scales and angles. Experiments on commonly used benchmarks (DOTA1.0, DOTA1.5, SSDD, and RSDD) demonstrate that the proposed FPNFormer module significantly improves the performance of multiple arbitrary-oriented object detectors, such as 1.99% map improvement of rotated retinanet on DOTA's cross-validation set. On RSDD datasets, the baseline model using FPNFormer improves the map of large objects by 5.1%. Combined with more competitive models, the proposed method can achieve a 79.39% map on the DOTA1.0 dataset. The code is available at https://github.com/bityangtian/FPNFormer.**

***Index Terms*— Arbitrary-oriented object detection, deep learning, rotation-invariant, transformer.**

I. Introduction

OBJECT detection is a very important downstream task in computer vision and Earth vision, which has a wide range of applications in many fields. Among them, the development of arbitrary-oriented object detection is in progress [1], [2], [3]. Different from the common object detection task, arbitrary-oriented object detection aims to predict the class, height, width, and rotation angle of the object. In other words, the task of arbitrary-oriented object detection is to predict oriented bounding boxes (OBBs), not horizontal bounding boxes (HBBs) [4], [5], [6], [7].

Manuscript received 24 August 2023; revised 21 November 2023; accepted 1 January 2024. Date of publication 8 January 2024; date of current version 29 January 2024. This work was supported in part by the National Key Research and Development Plan of China under Grant 2021YFB3901300, in part by the National Natural Science Foundation of China under Grant 62001023, and in part by the Beijing Natural Science Foundation under Grant 4232013. *(Corresponding author: Mengmeng Zhang.)*

Yang Tian, Mengmeng Zhang, Jinyu Li, Yangfan Li, and Wei Li are with the School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China (e-mail: yangtian@bit.edu.cn; mengmengzhang@bit.edu.cn; 3120215389@bit.edu.cn; yangfanli2020@163.com; liwei089@ieee.org).

Hong Yang is with the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China (e-mail: yanghong@aircas.ac.cn).

Digital Object Identifier 10.1109/TGRS.2024.3351156

However, most of the models that perform well on mainstream datasets (e.g., DOTA) still use the framework of convolution neural networks (CNNs). They pay more attention to applying the method of common object detection to arbitrary-oriented object detection and then put forward targeted improvements to the problems existing in arbitrary-oriented object detection. For example, some works focus on how to represent the intersection over union (IOU) between OBBs. For example, GWD [8] and KLD [9] model the oriented boxes as a 2-D Gaussian distribution, and then compute the Gaussian Wasserstein distance and Kullback–Leibler divergence to replace the IOU loss between oriented boxes, respectively. Some work has focused on designing sophisticated modules to give models more powerful representations, such as S^2ANet [6] and R^3Det [10]. The former realizes feature alignment by designing a feature alignment module (FAM) and an oriented detection module (ODM), and the latter realizes feature reconstruction and alignment by a feature refinement module. There are also some works focusing on the research of high accuracy and lightweight detectors so that the arbitrary-oriented object detector can be effectively applied to various fields, such as PP-YOLO-R [11] and RTMDet [12]. Recent research focuses on large convolution kernel, large receptive field, and rotation convolution [12], [13], [14], [15], but the research is still based on CNN components.

It is undeniable that the above works have greatly promoted the development of rotating object detection, but most of them are limited to the structure of CNNs. With the proposal of transformer [16], this structure has a profound impact on various fields such as natural language processing and computer vision. A large number of excellent works have emerged in object detection [17], [18], [19], [20] image segmentation [21], [22], [23] and hyperspectral and multispectral image fusion [24], and provide new ideas for the following researchers. However, there are still few studies on the use of transformer in arbitrary-oriented object detection, and the characteristics of transformer over CNNs are ignored. We summarize the two advantages of the transformer for solving rotating object detection as follows.

1) *Transformer can better deal with the problem of arbitrary orientation of objects in remote sensing images:* The acquisition method of remote sensing images is special. It is taken from the perspective of overhead, which results in remote sensing images having a high degree of rotation, and objects in the images having

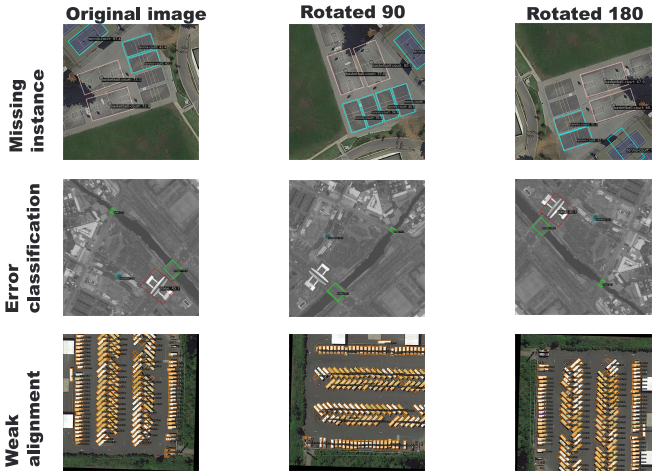

Fig. 1. Figure shows the limitations of the existing model. All results are output by the retinanet detector. With the change of image orientation, the model will lose the target, misclassify an instance, and predict the rotation angle inaccurately. The calculation method of CNN has strong orientation dependence, which makes it difficult to deal with complex object orientation.

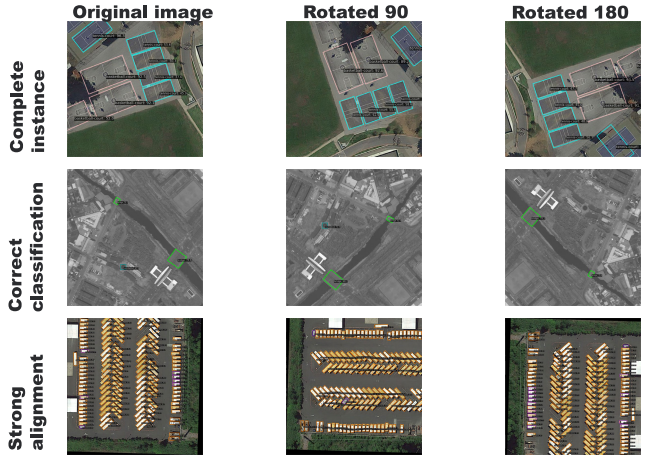

Fig. 2. All the results are from the retinanet with the FPNFormer module. We can see that after using FPNFormer, retinanet detects the previously lost instances, classifies the previously misclassified objects correctly, and captures more accurate angle information. It shows that FPNFormer makes the model more expressive and robust.

arbitrary orientation. For the same remote sensing image, when rotating it by any angle, the semantic information of the objects in the image will not change and only the position information has changed accordingly. This naturally brings corresponding challenges. How to ensure that the extraction of semantic information is not affected by image rotation while extracting the position information suitable for image rotation in the case of highly rotating images? As shown in Fig. 1, the way convolution neural computation works cannot deal with this problem. In the field of point cloud processing, we face a similar problem as the above, that is, the input of the point cloud is out of order, and we do not want the process of the point cloud after the network processing to be affected by the order of the point cloud input. The classical point cloud processing network PointNet [25] uses the max pooling and shared multilayer perception (MLP) to avoid the influence of the point cloud input order. In other words, we want to find a calculation method different from the CNN, so that the semantic information extraction of the image in the network is independent of the image rotation, and the transformer decoder is an excellent calculation method. With the use of positional encoding, the transformer has the ability to perceive the orientation of objects. In this article, we propose to add a transformer processing stage to the features extracted by the backbone network.

2) *Transformer is more flexible to handle multiscale object features:* In remote sensing images, in addition to the high degree of rotation of the image, the different heights of the image acquisition also bring a great challenge. Even if the same object has different scales in different remote sensing images, the traditional feature fusion mostly adopts the way of feature pyramid network (FPN) or improves based on the FPN [26], [27]. However, this feature fusion method still uses convolution to further map the fused features, which has two shortcomings. First, the same object may occupy different sizes of pixel areas in different depths of feature maps. FPN uses the way of upsampling feature maps and using convolution for feature fusion, which cannot make the object get a more consistent representation on different feature maps. Second, limited by the selection of the size of the convolution kernel, the mapped features only interact locally, and the overall perception ability is poor. In this article, we propose to use the transformer decoder to flexibly interact with multiscale features, and design a top-down and down-top way to effectively and fully fuse features.

In general, there are two major challenges in remote sensing images, namely the image is rotatable and the object scale changes greatly. Our proposed feature pyramid network transformer decoder (FPNFormer) can effectively obtain semantic information, position, and orientation information of objects while alleviating the influence of image rotation on object semantics with the help of transformer. It is used to solve the problem of large-scale change in cross-scale processing. Fig. 2 illustrates the performance improvement obtained by the model using the FPNFormer module. The main contributions of this article are as follows.

1) At present, most arbitrary-oriented object detection methods are based on CNN, which have natural limitations and a weak ability to model the rotation-invariance of object semantics and the rotation-equivariance of object orientation. To the best of our knowledge, we first employ a transformer to achieve rotation-invariance and rotation-equivariance and solve large-scale changes in remote sensing images. And we input and analyze the advantages of using transformer, which provides new ideas for later research.
2) The proposed FPNFormer uses the cross-scale transformer decoder to model the rotation-invariance of the object. Moreover, the stacked multilayer two-stream transformer decoder structure is used to fully perform feature fusion, which has a stronger ability to fuse multiscale features than the traditional FPN.

3) The proposed FPNFormer uses sinusoidal position embedding to make the model more focused on the relative position relationship of each part of the object and model the rotation-equivariance of object orientation, which has a stronger ability to perceive object orientation information than CNN.
4) The proposed FPNFormer is a plug-and-play module, which can be easily applied to other models, and makes the model obtain good improvement with adding a small number of parameters and computational cost. The proposed FPNFormer is verified on multiple models and datasets.

The rest of this article is arranged as follows. Section II reviews and analyzes previous related work. Section III introduces the design idea and implementation details of FPNFormer in detail. Section IV provides our experimental results on various datasets. Section V draws conclusions and finally we analyzes the limitations of our proposed method.

II. Related Works

A. Arbitrary-Oriented Object Detection

Benefiting from the emergence of large-scale open-source datasets, arbitrary-oriented object detection has a foundation for long-term development. At present, the mainstream arbitrary-oriented object detectors still use the framework of CNNs. According to the inference stage, the detectors can be divided into single-stage detectors [11], [12], refine-single-stage detectors [10], and two-stage detectors [28]. The relatively high accuracy of two-stage detectors has attracted much attention in the industry. At present, the most advanced single-stage detector can reach the same level as the two-stage detector, and the two-stage detector still dominates in performance. To the best of our knowledge, the current models [14] that perform extremely well on the DOTA1.0 dataset are still based on oriented R-CNN [28].

The arbitrary-oriented object detector based on CNN has been quite mature developed, but the pipeline of most methods is still relatively complex, such as S^2aNet [6] proposed to generate high-quality anchors by refining horizontal anchors into rotating anchors, and then use the proposed FAM module to achieve feature alignment with deformable convolution. Adaptive rotated convolution (ARC) [14] regards the convolution kernel as a parameter space, and uses interpolation technology to adaptively rotate the convolution kernel, to construct a backbone network to solve the problem of arbitrary orientation of the object. Different from the above methods based on CNNs, FPNFormer uses transformer-based components to propose a more elegant method to improve the model's ability to extract object semantic information and capture object orientation changes in the case of highly rotatable images, and solve the problem of large object scale changes in remote sensing images.

B. Transformer Network and Its Application

The core of the transformer is the self-attention mechanism in the transformer encoder and the cross-attention mechanism in the transformer decoder. The model has excellent long-distance modeling ability and the ability to exchange information across modes, which is a more flexible calculation method. And many studies have shown the advance of transformer. In object detection, end-to-end object detection with transformers (DETR) [17] has designed an end-to-end detector using transformer. A series of subsequent works are completed on the basis of DETR, some focus on solving the problems of expensive training of DETR [18], [29], [30], [31], some focus on improving the detection performance of DETR [32], [33], and some works devote to making DETR available for real-time detection [34]. Although there are still few transformer detectors represented by DETR in the field of remote sensing, the strong performance of transformers exhibited in these works is enough to attract our attention. Transformer is very suitable for operating on points and is often used in multimodal data fusion [35], [36], [37], [38]. This kind of data interaction inspires us to apply it to multiscale feature map fusion, replacing the traditional feature fusion method represented by FPN, and by virtue of its output is not affected by the input order we can effectively extract important semantic information without the effect of image rotation. In general, our proposed FPNFormer is a more flexible way of context modeling.

C. Rotation-Invariance and Rotation-Equivariance in Arbitrary-Oriented Object Detection

Rotation-invariance and rotation-equivariance have always been challenging problems in arbitrary-oriented object detection. The former means that the semantic information of the foreground remains unchanged during the image rotation, while the latter means that the spatial position and orientation information of the former change during the image rotation. Most of the existing methods to solve this problem are still mostly based on designing rotation-sensitive CNN [6], [39], [40], [41]. For example, ReDet [42] incorporated rotation-equivariant networks into the detector to extract the rotation-equivariant feature, which can predict the orientation accurately. S^2Net [6] proposed an ODM, which is constructed to encode the orientation information and capture the rotation-invariant feature by adopting active rotating filters. What's more, some works solve this problem by weak supervision. Such as RINet [43] used flexible multibranch online detector refinement to be naturally more rotation-perceptive against oriented objects. In addition, we can improve the rotation sensitivity of CNN by using some data augmentation methods, such as random rotation.

In the face of the problem of rotation-invariance, the predecessors have made a lot of efforts on CNN, but it is undeniable that the calculation method of convolution itself is difficult to deal with the task in the case of high rotation, and we need to find a more suitable calculation method to deal with this problem instead of convolution.

III. Proposed Method

The overall framework diagram of the proposed FPNFormer is shown in Fig. 3, which consists of position encoding and window transformer encoder. FPNFormer contains a top-down cross-scale transformer decoder and a down-top cross-scale transformer decoder, where the choice of position encoding is a key point; good position embedding can greatly improve

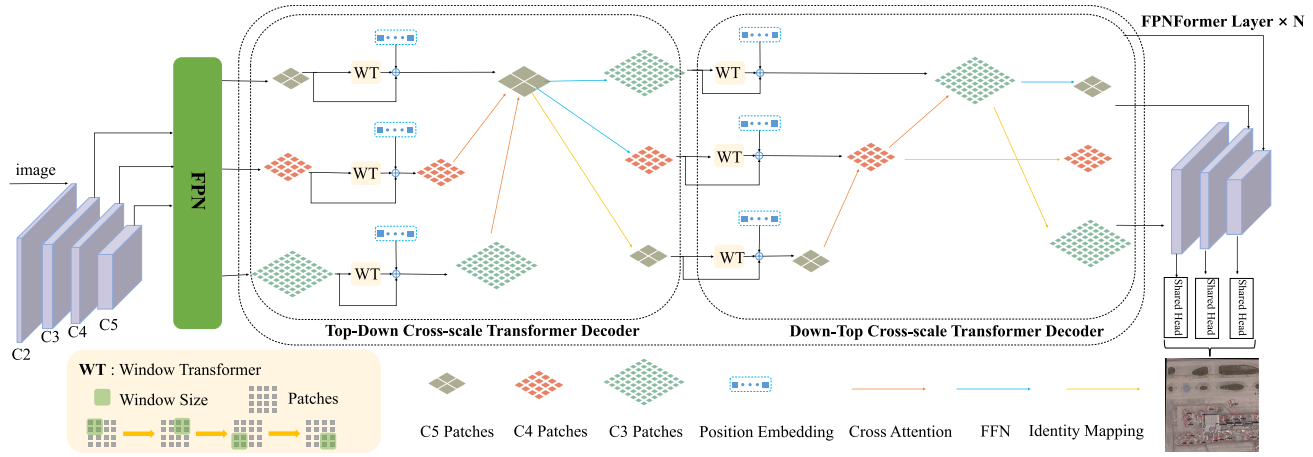

Fig. 3. Illustration of the proposed FPNFormer. After extracting the feature map from the image through the backbone network, FPN is first used to compress the feature map to the same number of channels. After that, the top-down cross-scale transformer decoder and the down-top cross-scale transformer decoder are used to model the rotation invariance. To better capture the orientation equivariant information of the object, position embedding is used in each attention calculation. The FPNFormer layer can be repeated many times to fully fuse the features.

the performance of FPNFormer. Window transformer encoder is used to deal with the problem that the large receptive field of cross-scale interaction is difficult to deal with local information. Top-down cross-scale transformer decoder and down-top cross-scale transformer decoder are designed for information flow between different layers.

A. Position Embedding

Position embedding is the key point in our proposed method. It is undeniable that transformer has a weak dependence on the order of input data. If it keeps its weak dependence completely, it is not conducive to processing information with strong context, such as RGB images. So, we strongly recommend incorporating positional embedding, and we adopt sinusoidal position embedding. The exact location embedding is as follows:

$$\begin{aligned} p_{i,2j} &= \sin\left(\frac{i}{10000^{2j/d}}\right) \\ p_{i,2j+1} &= \cos\left(\frac{i}{10000^{2j/d}}\right). \end{aligned} \tag{1}$$

Among them, i represents the position of the feature vector in the row or column of the feature map, j represents the jth position of the feature dimension of the feature vector, and d represents the feature dimension of the feature vector. Since both the row and column of the feature vector need to be encoded, we take d to be one-half of the feature dimension.

Not only can absolute position information be obtained, but also relative position information can be obtained using this position embedding, this is because for any determined position offset δ, the position embedding at position $\delta + i$ can be represented by linearly projecting the position embedding at position i. We can give a mathematically simple proof by letting

$$\omega_j = \frac{1}{10000^{2j/d}} \tag{2}$$

such that for any determined position offset δ, any pair of $(p_{i,2j}, p_{i,2j+1})$ in (1) has been linearly projected to obtain $(p_{i+\delta,2j}, p_{i+\delta,2j+1})$.

The proof is as follows:

$$\begin{aligned} &\begin{bmatrix} \cos(\delta\omega_j) & \sin(\delta\omega_j) \\ -\sin(\delta\omega_j) & \cos(\delta\omega_j) \end{bmatrix} \begin{bmatrix} p_{i,2j} \\ p_{i,2j+1} \end{bmatrix} \\ &\quad = \begin{bmatrix} \cos(\delta\omega_j)\sin(i\omega_j) + \sin(\delta\omega_j)\cos(i\omega_j) \\ -\sin(\delta\omega_j)\sin(i\omega_j) + \cos(\delta\omega_j)\cos(i\omega_j) \end{bmatrix} \\ &\quad = \begin{bmatrix} \sin\left((i+\delta)\omega_j\right) \\ \cos\left((i+\delta)\omega_j\right) \end{bmatrix} \\ &\quad = \begin{bmatrix} p_{i+\delta,2j} \\ \\ p_{i+\delta,2j+1} \end{bmatrix}. \end{aligned} \tag{3}$$

The mapping matrix formed by w_j and δ does not depend on any position index i. The linear transformation nature of this position encoding gives the model a better perception of relative position, which is what we need, rather than absolute position because it is more important to capture the consistent dependence of object parts under high rotation than where they are in the feature map.

B. Window Transformer Encoder

Considering that each query in the cross-scale transformer decoder we use interacts with the whole cross-scale feature map, which has a large receptive field but is difficult to take into account the local information in the feature map, inspired by the swin transformer [44], We use self-attention-based window attention to enhance the capture of local features, and different from swin transformer, we do not move the window to process local information effectively. Here is how the window transformer encoder works

$$\begin{aligned} \mathbf{z}^{l} &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1} \\ \mathbf{z}^{l} &= \text{FFN}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l} \\ \mathbf{z}^{l} &= \text{W-MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l} \\ \mathbf{z}^{l+1} &= \text{FFN}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l} \end{aligned} \tag{4}$$

where z^{l-1} and z^{l+1} denote the input features and output features, respectively. W-MSA denotes the window attention-based self-attention, while FFN denotes feed forward networks, and LN is short for layernorm.

TABLE I
ABLATION EXPERIMENT ON THE VALIDATION SET OF DOTAV1.0 DATASET

Model	Up-To-Down	Down-To-Up	Num-Layers	Sinusoidal-Position-Embedding	Learnable-Position-Embedding	mAP	FLOPs (G)	Parameters (MB)
Baseline						61.92	215.92	36.42
FPNFormer	✓		1	✓		62.20	237.43	39.58
FPNFormer	✓	✓	1	✓		62.94	215.92	36.42
FPNFormer	✓	✓	2	✓		**63.91**	263.25	49.06
FPNFormer	✓	✓	3	✓		63.58	286.92	55.39
FPNFormer	✓	✓	2		✓	62.77	263.25	72.13
FPNFormer	✓	✓	2			62.63	263.25	72.13

C. Up-to-Down Cross-Scale Transformer Decoder

The deep feature map has richer semantic information, and some work [45] has shown that deep semantic information is crucial for object detection. We fuse the deep semantic information to the shallow layer to further enrich the information of the shallow feature map. Different from FPN, we propose to use cross-attention to fuse features. The calculation process can be expressed by the following formula:

$$Q_i = f_i W_q + E_i^{\text{pos}}, \quad K_j = f_j W_k + E_j^{\text{pos}}, \quad V_j = f_j W_v \tag{5}$$

where W_q, W_k, W_v are the project matrix of Q_i, K_i, V_i, and $E_i^{\text{pos}}, E_j^{\text{pos}}$ are the position embedding matching different scale features, while f_i, f_j is the corresponding feature map

$$\text{SoftMax}(z_j) = \frac{\exp(z_j)}{\sum_j \exp(z_j)} \tag{6}$$

$$f_i^{\text{out}} = \sum_{m=1}^{M} W_m \left[\sum_{j \in \Omega_j} \text{SoftMax}\left(\frac{Q_i K_j}{\sqrt{d}} \right) \cdot V_j \right] \tag{7}$$

where Ω_j represents the domain in which the feature map which is decoded is located, W_m denotes the coefficient at the m^{TH} self-attention head, and f_i^{out} is the output of layer i feature map decoding layer j feature map.

Inspired by YOLOF [45], how to make good use of the semantic information of the deep feature map is very important, so we do not design a progressive decoder method, but use the c4 feature map to perform cross-scale transformer decoder operation on the c5 feature map when fusing the deep information to the shallow layer while c3 feature map performs cross-scale transformer decoder operation on c5 feature map. In other words, the information of the c5 feature map is transmitted to the c3 feature map without passing through the c4 feature map. On the one hand, it can reduce the amount of calculation and computational complexity; on the other hand, the semantic information of c5 is more abundant than c4, which is more beneficial to detection.

Some works [46] implement the transformer decoder to limit the computation to a window of a certain size (e.g., 7×7 window size), but in this article, we propose to decode the entire feature map, which avoids the choice of window size as a hyperparameter. At the same time, because objects of different sizes occupy different ranges on the feature map, directly decoding the whole feature map can flexibly deal with the change of object scale.

D. Down-to-up Cross-Scale Transformer Decoder

The up-to-down cross-scale transformer decoder receives the limitation of unidirectional information flow. To solve this problem, we design the bottom-up path aggregation network. In Table I, we do detailed experiments to illustrate the effectiveness of the bottom-up path aggregation network.

The down-to-up cross-scale transformer decoder works very similar to the up-to-down cross-scale transformer decoder. Specifically, we treat each point in the c5 feature map as a patch to obtain a single-scale patch. We first process the local features through the two-layer window transformer encoder and then treat each point in the c3 feature map and c5 feature map as a patch to obtain a multiscale patch. Then, single-scale patch is used to perform cross-attention operation on a multiscale patch, and a more visualized process is shown in Fig. 4.

This bottom-up information flow can not only fully capture any information, but also transfer the information of small objects to the deep feature map, which alleviates the problem of losing small objects due to downsampling. The detection effect of small objects such as small vehicle (SV) and ship (SH) has been significantly improved, and the specific results are shown in Table II.

Our proposed FPNFormer module is a kind of module that can be stacked, and an appropriate amount of increasing the number of modules can further improve the effect of the model. In the ablation experiment, we discuss the effect improvement brought by stacking module.

IV. EXPERIMENTS

In this section, we conduct experiments on publicly available rotating object detection datasets to verify the effectiveness of the proposed FPNFormer. We conduct ablation experiments on the DOTAv1.0 validation set to detail the interactions within the FPNFormer module. Then, we use the FPNFormer module on different models to verify its broad applicability. The highly competitive method RCNN [28] with our proposed FPNFormer is to compare with the mainstream detectors. Finally, we conduct comparative experiments on public datasets with different scenes.

A. Experimental Conditions

1) *Experimental equipment and experimental environment:* All experiments are performed on an NVIDIA

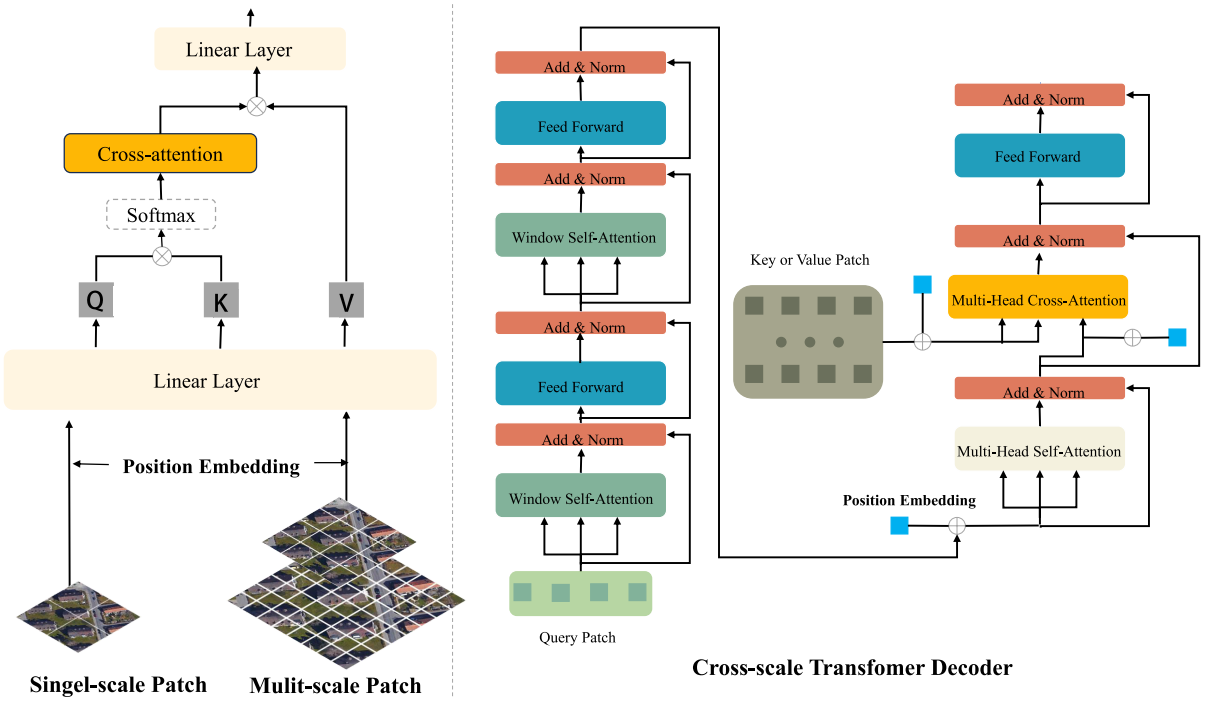

Fig. 4. Execution details of the cross-scale transformer decoder. The left figure shows how the cross-attention is calculated in the down-top cross-scale transformer decoder. That is, the feature map is first mapped into fine-grained patches, and then the position encoding is added, respectively, and the query mapped by the high-level patches is used to calculate the attention with the low-level patches. The figure on the right shows how the cross-scale transformer decoder works. The window transformer is used to strengthen the local features of the feature map, and the standard transformer decoder is used to fuse the cross-scale features.

GeForce RTX 3090 GPU, we use torch1.13.1 and torchvision0.14.1; in addition, we use the mmrotate [49] framework with mmengine0.7.3, mmdet3.0.0, and mmrotate1.0.0rc1, more detailed experimental environment to see our open source code.

2) *Datasets:* In order to fully evaluate the performance of our proposed FPNFormer, we conduct experiments on multiple public datasets.

   a) *DOTAv1.0:* The DOTAv1.0 [1] dataset is the most widely used public dataset for rotating object detection at present. It has 2806 aerial images with a total of 188 282 instances from 800 × 800 pixels to 4000 × 4000 pixels, containing 15 common categories. The categories of the objects in DOTA are: plane (PL), baseball diamond (BD), bridge (BR), ground field track (GFT), SV, large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). The proportions of the training set, validation set, and testing set in DOTA-v1.0 are 1/2, 1/6, and 1/3, respectively. Due to the sharp change of image size in the DOTA dataset, we crop each image to a 1024 × 1024 size image and use three scales for data augmentation 0.5, 1.0, 1.5. Random flip and random rotation were used in all training experiments. The random flip flipped horizontally, vertically, and diagonally with 75% of the time, and the random rotation rotated within 180° with 50% of the time.

   b) *DOTAv1.5:* DOTAv1.5 uses the same images as DOTAv1.0, but extremely small instances (less than 10 pixels) are also annotated. In addition, a new category, "container crane," was added. It contains a total of 403 318 instances. The number of images and dataset division is the same as in DOTA-v1.0.

3) *Rotated ship detection dataset (RSDD):* RSDD [58] is a SAR dataset, which consists of 84 scenes GF-3 data slices, 41 scenes TerraSAR-X data slices, and two scenes uncropped large images, includes 7000 slices and 10263 ship instances of multiobserving modes, multipolarization modes, and multiresolutions, and has the characteristics of arbitrary rotation direction, large aspect ratio, high proportion of small targets, and rich in scenarios. This dataset provides a very rich set of ships with various orientations and complex backgrounds, which is a good dataset to test the performance of our proposed method.

4) *SAR ship detection dataset (SSDD):* The SSDD [59] dataset is the most widely used SAR dataset. It contains 1160 images with 2456 ships, and the average number of ships per image is 2.12. SAR image samples in SSDD have different resolutions, different sensors, different polarizations, different sea states, different ship scenes, both offshore and inshore, and different ship sizes. We chose this dataset in the first place because we wanted to see how the proposed method performs on small dataset.

5) *Implementation details:* For experiments on DOTAv1.0 dataset, retinanet uses stochastic gradient

TABLE II
EXPERIMENT RESULTS ON THE TEST SET OF DOTAV1.0 DATASET AMONG VARIOUS DETECTORS

Method	Neck	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
One-Stage Methods																	
retinanet[47]	FPN	89.18	82.36	46.89	72.63	76.90	72.71	83.85	90.66	82.56	86.04	64.67	64.94	63.07	70.02	61.65	73.89
	FPNFormer	89.69	81.73	45.21	76.74	78.88	74.17	85.80	90.16	82.89	85.96	63.41	70.16	62.68	77.20	55.50	**74.68**
S^2aNet[6]	FPN	89.53	81.51	56.01	79.68	79.81	83.53	88.21	90.85	85.04	87.80	71.14	65.69	77.06	72.67	61.10	77.98
	FPNFormer	88.02	81.62	56.64	80.63	81.64	85.57	88.37	90.87	84.29	87.21	69.43	66.67	76.81	75.80	58.24	**78.12**
R^3Det[10]	FPN	89.06	80.37	52.84	78.46	79.88	82.04	87.97	90.83	82.39	87.44	66.13	70.08	67.55	69.72	47.53	75.49
	FPNFormer	89.32	82.05	52.83	79.62	51.55	83.47	88.16	90.79	82.84	86.90	65.13	68.71	67.74	72.12	50.11	**76.10**
Two-Stage Methods																	
Oriented R-CNN[28]	FPN	88.05	82.95	58.20	77.15	79.71	85.49	88.41	90.69	83.59	86.42	67.38	66.28	78.15	77.01	61.15	78.05
	FPNFormer	87.87	81.57	59.71	78.08	79.49	85.55	88.40	90.48	83.23	86.29	67.38	69.39	77.86	76.71	62.43	**78.30**
Rotated Faster R-CNN[48]	FPN	86.97	81.19	55.71	76.01	79.21	79.67	87.58	90.18	83.68	87.14	65.98	66.20	76.07	75.06	55.54	76.41
	FPNFormer	87.31	80.89	55.78	75.56	79.33	80.45	87.53	90.22	84.43	86.64	65.13	67.45	76.53	77.91	60.82	**77.07**

TABLE III
EXPERIMENTAL RESULTS ON THE TEST SET OF DOTAV1.0 DATASET COMPARED WITH OTHER METHODS

Method	Backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
One-Stage Methods																	
DAL[50]	resnet101	88.68	76.55	45.08	66.80	67.00	76.76	79.74	90.84	79.54	78.45	57.71	62.27	69.05	73.14	60.11	71.44
R^3Det[10]	resnet50	89.80	83.77	48.11	66.77	78.76	83.27	87.84	90.82	85.38	85.51	65.57	62.68	67.53	78.56	**72.62**	76.47
S^2aNet[6]	resnet50	89.11	82.84	48.37	71.11	78.11	78.39	87.25	90.83	84.90	85.64	60.36	62.60	65.26	69.13	57.94	74.12
AO^2-DETR[7]	resnet50	89.27	84.97	**56.67**	74.89	78.87	82.73	87.35	90.50	84.68	85.41	61.97	69.96	74.68	72.39	71.62	77.73
Two-Stage Methods																	
RoI-Transformer[4]	resnet101	88.64	78.52	43.44	75.92	68.81	73.68	83.59	90.74	77.27	81.46	58.39	53.54	62.83	58.93	47.67	69.56
Gliding Vertex[51]	resnet101	89.64	**85.00**	52.26	77.34	73.01	73.14	86.82	90.74	79.02	86.81	59.55	**70.91**	72.94	70.86	57.32	75.02
SCRDet[5]	resnet101	89.98	80.65	52.09	68.36	68.36	60.32	72.41	90.85	**87.94**	86.86	65.02	66.68	66.25	68.24	65.21	72.61
CSL[52]	resnet152	**90.25**	85.53	54.64	75.31	70.44	73.51	77.62	90.84	86.15	86.69	**69.60**	68.04	73.83	71.10	68.93	76.17
Anchor-Free																	
DRN[53]	hourglass104	89.45	83.16	48.98	62.24	70.63	74.25	83.99	90.73	84.60	85.35	55.76	60.79	71.56	68.82	63.92	72.95
Oriented RepPoints[54]	resnet50	87.02	83.17	54.13	71.16	80.18	78.40	87.28	**90.90**	85.97	86.25	59.90	70.49	73.53	7.27	58.97	75.97
CFA[55]	resnet101	89.26	81.72	51.81	67.17	79.99	78.25	84.46	90.77	83.40	85.54	54.86	67.75	73.04	70.24	64.96	75.05
CFA[55]	resnet152	89.08	83.20	54.37	66.87	81.23	80.96	87.17	90.21	84.32	86.09	52.34	69.94	75.52	**80.76**	67.96	76.67
Ours																	
S^2aNet_FPNFormer	resnet50	89.10	82.41	56.15	**81.24**	**82.05**	**85.16**	**88.74**	90.89	85.78	**87.07**	68.93	68.47	**77.93**	78.65	68.35	**79.39**

descent (SGD) optimizer with a learning rate set to 0.025, momentum set to 0.9, and weight decay set to 0.0001. The first 500 iterations use a linear learning rate warmup strategy. The learning rate of oriented R-CNN is 0.005, and the rest of the policies are the same as the former. We fine-tune the information strategy when using FPNFormer, that is, all models learn the same strategy before and after using FPNFormer.

TABLE IV
COMPARISON WITH OTHER METHODS THAT USE FEATURE FUSION ON THE VALIDATION SET OF DOTAV1.0

Method	mAP	FLOPs (G)	Parameters (MB)
Baseline	61.92	215.92	36.42
PAFPN[60]	62.30	221.96	38.78
NASFPN[61]	59.86	326.23	58.41
BIFPN[26]	60.23	318.95	69.97
FPNFormer	63.91	263.25	49.06

B. Ablation Experiment

In the ablation experiment, we use a train set to train the model and verify the results on the validation set, and retinanet [47] is selected as our baseline model. The detailed experimental results are shown in Table I. In this article, we take 0.5 as the threshold of IOU, which is used as the basis of whether the object is detected or not. When we use only the top-to-bottom part of FPNFormer, we get a small improvement in the map. When we add the bottom-to-top path aggregation network, we get a 1.02% improvement in the map compared to the baseline. We treat the top-to-bottom information passing, and bottom-to-top path aggregation as a layer of FPNFormer, and the map is further improved (+1.99%) after stacking two layers. However, when we further add the FPNFormer layer, the map decreases slightly, indicating that the two-layer FPNFormer performance is saturated for the retinanet model.

Position encoding is also an important part of the FPNFormer. We ablate the use of position encoding based on using two FPNFormer layers. When using learnable position embedding, it is only 0.14% map higher than that of sinusoidal position embedding, but 0.81% map lower than that of cosine position encoding. It indicates that the model is difficult to learn useful position information, and our cosine position encoding gives the model a strong ability to perceive relative position.

TABLE V
EXPERIMENTAL RESULTS ON THE TEST SET OF DOTAV1.5 DATASET COMPARED WITH OTHER METHODS. MS MEANS MULTISCALE TRAINING AND TESTING

Method	MS	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP
RetinaNet[47]		71.43	77.64	42.12	64.65	44.53	56.79	73.31	90.84	76.02	59.96	46.95	69.24	59.65	64.52	48.06	0.83	59.16
Oriented Faster R-CNN[48]		71.89	74.47	44.45	59.87	51.28	69.98	79.37	90.78	77.38	67.50	47.75	69.72	61.22	65.28	60.47	1.54	62.00
HTC[56]		77.80	73.67	51.40	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.40
DAFNet[57]	✓	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	71.99
Ours																		
RetinaNet[47]	✓	80.11	84.49	46.75	67.74	55.99	69.71	78.96	89.37	79.51	75.90	60.74	73.94	63.33	71.39	62.97	6.25	66.70
RetinaNet_FPNFormer	✓	80.97	84.46	49.25	72.44	50.12	73.37	86.10	90.42	80.54	73.49	60.41	74.09	63.11	72.90	65.64	14.30	**68.23**
S^2aNet[6]	✓	80.92	84.66	56.74	80.05	72.62	81.36	89.63	90.87	83.29	85.36	67.13	75.72	76.16	74.75	65.98	20.52	74.11
S^2aNet_FPNFormer	✓	80.98	83.18	57.94	78.69	73.53	82.52	89.59	90.89	83.72	85.56	65.01	74.98	76.24	73.77	67.81	24.89	**74.33**

We use retinanet as our baseline to compare the differences between different methods by replacing its feature fusion network. Specifically, for bi-directional feature pyramid network (BIFPN), we used three layers, and for FPNFormer we followed the same configuration as for ablation experiments and used two layers. It can be seen that for path aggregation feature pyramid network (PAFPN) [60], although FLOPs and parameters are not significantly improved, the mAP improvement is also negligible. For learning scalable feature pyramid network (NASFPN) [61] and BIFPN [26], increasing FLOPs and parameters did not improve the performance of the model. The above methods all have two shortcomings: first, they all use convolution to perform feature fusion, the receptive field is limited by the size of the convolution kernel, and the model lacks the ability to model long-range features in cross-scale feature fusion. Second, they cannot capture consistent features when the object has any orientation. Our proposed method achieves a considerable improvement in model performance with a small increase in FLOPs and parameters, showing strong competitiveness compared with the above methods.

C. Effectiveness on Various Architectures

To further investigate whether the proposed method has broad applicability, we conduct a large number of experiments using different models on the DOTAv1.0 dataset, and the detailed experimental results are shown in Table II. In this experiment, we do not use the method of using the training set and cross-validation set together to make the model have better performance while training. The widely used training method, instead, only uses the training set to train, and when using the proposed method on the baseline model, the training policy is consistent with the baseline model's training policy without any fine-tuning, to demonstrate the broad effectiveness of the proposed method.

Retinanet, S^2aNet, and R^3Det are used in the single-stage segment model, and R-CNN and faster R-CNN are used in the two-stage model. It can be seen that the map is significantly improved after using our proposed method on various baseline models. An interesting phenomenon we notice is that after using our proposed method, the model has greatly improved the detection of large objects, such as retinanet's map for the SP is increased by 7.18%, the map for an LV is increased by 1.46%, RCNN's map for SP is increased by 2.85%, and the map for an LV is increased by 0.78%. The detection ability of each model for large objects has been improved. We attribute this to the excellent long-range modeling capabilities of transformer.

In Table III, we choose S^2aNet using our proposed method and use multiscale training, train the training set, and cross-validation set together, and obtain a map of 79.39% on the test set, which shows strong competitiveness compared with other models. To further demonstrate the superiority of our proposed method over other feature fusion methods, we compare different methods in Table IV.

D. Effectiveness on Various Datasets

To further investigate the performance of our proposed method on different datasets, we use different datasets and different models for further research.

First of all, in Table V, we use DOTAv1.5 for experiments. It can be seen that after using our proposed method, there is a considerable improvement on the map of extremely small objects such as CC. For example, retinanet improves the map of the container crane by 8.05% after using the proposed method. S^2aNet improves the map of the container crane by 4.37% after using the proposed method. For small targets such as container cranes, with the expansion of the receptive field and the deepening of the number of network layers, its information is difficult to effectively preserve. Although the traditional FPN operation fuses multiscale information, its information fusion is still incomplete. Our proposed FPNFormer adopts point-to-point transformer decoder operation, which can fully perform multiscale feature fusion. The orientation information and semantic information of small objects can be retained in the deep layer, alleviating the loss of small object information caused by too large a receptive field, which is an important reason for the huge improvement of the performance of small objects such as container cranes.

Then, we conduct experiments on the RSDD dataset. In order to better represent the effectiveness of our proposed method, we use coco-type evaluation metrics. The definition of mAP is consistent with the above, and AP_{small} refers to the IOU threshold with a value of 0.5 to 0.95 every 0.05. The average of all AP is taken as the AP of small objects. AP_{large} refers to the IOU threshold of 0.5 to 0.95 every 0.05. Finally, the average of all AP is taken as the AP of large

TABLE VI
EXPERIMENTS ON THE RSDD DATASET. AP IS SHORT FOR "AVERAGE PRECISION" AND AR IS SHORT FOR "AVERAGE RECALL"

Method	Neck	mAP	AP_{small}	AP_{large}	AR_{small}	AR_{large}
R^3Det[10]	FPN	80.87	-	-	-	-
Faster R-CNN[48]	FPN	83.29	-	-	-	-
Ours						
Retinanet[47]	FPN	83.6	39.2	35.3	47.5	35.0
	FPNFormer	83.9	40.4	40.4	48.4	40.0

TABLE VII
EXPERIMENTS ON THE SSDD DATASET

Method	Neck	mAP	AP_{small}	AP_{large}	AR_{small}	AR_{large}
PSC[62]	FPN	83.7	33.3	58.6	41.3	60.0
Ours						
Retinanet[47]	FPN	83.9	33.7	53.8	41.5	55.0
	FPNFormer	83.8	33.7	53.6	41.3	57.5
GWD[8]	FPN	85.6	35.7	49.8	43.4	52.5
	FPNFormer	85.7	35.1	51.1	42.8	55.0

objects. AR_{small} refers to the IOU threshold with a value of 0.5 to 0.95 every 0.05. The average of all AR is taken as the AR of small objects. AR_{large} refers to the IOU threshold with a value of 0.5 to 0.95 every 0.05. The average of all AR is taken as the AR of large objects. It can be seen that the detection performance of retinanet is improved after using the proposed method for both small and large objects, and the detection performance for large objects is improved by up to 5.1% map in Table VI. This phenomenon is consistent with the experimental observation on the DOTAv1.0 dataset, which again indicates that the model obtains better long-range modeling ability after using our proposed method. Meanwhile, the AR index has also been greatly improved, indicating that the ability of our proposed method to advance the semantic information of the object has been improved when the image is highly rotatable.

Finally, we conducted experiments on the SSDD dataset. The detailed results are shown in Table VII. We observe that the semantic extraction ability of the model is improved after using the proposed method, that is, the AR_{large} is significantly increased. Regarding the reduction of other metrics, we will analyze in Section V.

V. CONCLUSION

In this article, we explore how to effectively handle highly rotatable remote sensing images in the arbitrary-oriented object detection task and propose the use of FPNFormer, which exhibits strong performance and excellent adaptability to different models. It uses a window transformer to enforce local features and uses a two-stream cross-scale transformer decoder and position embedding for rotation invariant and rotation equivariant modeling to cope with objects with high rotation and large object scale variation in remote sensing images. The performance of the model can be improved without any hyperparameter adjustment. We provide novel ideas for further exploring the potential of transformer in arbitrary-oriented object detection.

It is undeniable that our proposed method has shown excellent performance on different datasets and can be applied to different models, but its performance is not ideal on very small datasets such as SSDD. An important reason is that the transformer needs a relatively large amount of data support. We estimate that when the number of instances in the data is in the order of ten thousand, the performance of the model can be improved significantly by using our proposed method. Nowadays, arbitrary-oriented object detection is developing rapidly, and the acquisition of large-scale datasets is no longer a problem, and this drawback can be overcome.

REFERENCES

[1] G.-S. Xia et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3974–3983.

[2] W. Qian, X. Yang, S. Peng, J. Yan, and Y. Guo, "Learning modulated loss for rotated object detection," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 3, pp. 2458–2466.

[3] T. Zhang et al., "FFN: Fountain fusion net for arbitrary-oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–13, 2023, Art. no. 5609913.

[4] J. Ding, N. Xue, Y. Long, G.-S. Xia, and Q. Lu, "Learning RoI transformer for oriented object detection in aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2844–2853.

[5] X. Yang et al., "SCRDet: Towards more robust detection for small, cluttered and rotated objects," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8231–8240.

[6] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5602511.

[7] L. Dai, H. Liu, H. Tang, Z. Wu, and P. Song, "AO2-DETR: Arbitrary-oriented object detection transformer," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2342–2356, May 2023.

[8] X. Yang, J. Yan, Q. Ming, W. Wang, X. Zhang, and Q. Tian, "Rethinking rotated object detection with Gaussian Wasserstein distance loss," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11830–11841.

[9] X. Yang et al., "Learning high-precision bounding box for rotated object detection via Kullback–Leibler divergence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 18381–18394.

[10] X. Yang, J. Yan, Z. Feng, and T. He, "R^3Det: Refined single-stage detector with feature refinement for rotating object," in *Proc. AAAI Conf. Artif. Intell.*, May 2021, vol. 35, no. 4, pp. 3163–3171.

[11] X. Wang, G. Wang, Q. Dang, Y. Liu, X. Hu, and D. Yu, "PP-YOLOE-R: An efficient anchor-free rotated object detector," 2022, *arXiv:2211.02386*.

[12] C. Lyu et al., "RTMDet: An empirical study of designing real-time object detectors," 2022, *arXiv:2212.07784*.

[13] Y. Li, Q. Hou, Z. Zheng, M.-M. Cheng, J. Yang, and X. Li, "Large selective kernel network for remote sensing object detection," 2023, *arXiv:2303.09030*.

[14] Y. Pu et al., "Adaptive rotated convolution for rotated object detection," 2023, *arXiv:2303.07820*.

[15] X. Ding, X. Zhang, J. Han, and G. Ding, "Scaling up your kernels to 31×31: Revisiting large kernel design in CNNs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 11963–11975.

[16] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[17] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Computer Vision—ECCV*. Glasgow, U.K.: Springer, 2020, pp. 213–229.

[18] Z. Zong, G. Song, and Y. Liu, "DETRs with collaborative hybrid assignments training," 2022, *arXiv:2211.12860*.

[19] L. Ouyang, G. Guo, L. Fang, P. Ghamisi, and J. Yue, "PCLDet: Prototypical contrastive learning for fine-grained object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5613911.

[20] Y. Zhou, H. Liu, F. Ma, Z. Pan, and F. Zhang, "A sidelobe-aware small ship detection network for synthetic aperture radar imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5205516.

[21] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1280–1289.

[22] H. Zhang et al., "MP-former: Mask-piloted transformer for image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 18074–18083.

[23] L. Wu, L. Fang, X. He, M. He, J. Ma, and Z. Zhong, "Querying labeled for unlabeled: Cross-image semantic consistency guided semi-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 8827–8844, Jul. 2023.

[24] Z. Wang, M. K. Ng, J. Michalski, and L. Zhuang, "A self-supervised deep denoiser for hyperspectral and multispectral image fusion," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5520414.

[25] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[26] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10778–10787.

[27] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10208–10219.

[28] X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han, "Oriented R-CNN for object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3500–3509.

[29] Q. Chen et al., "Group DETR: Fast DETR training with group-wise one-to-many assignment," 2022, *arXiv:2207.13085*.

[30] H. Zhang et al., "DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection," 2022, *arXiv:2203.03605*.

[31] Y. Wang, X. Zhang, T. Yang, and J. Sun, "Anchor DETR: Query design for transformer-based detector," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 3, pp. 2567–2575.

[32] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: Deformable transformers for end-to-end object detection," 2020, *arXiv:2010.04159*.

[33] Q. Chen et al., "Group DETR v2: Strong object detector with encoder–decoder pretraining," 2022, *arXiv:2211.03594*.

[34] W. Lv et al., "DETRs beat YOLOs on real-time object detection," 2023, *arXiv:2304.08069*.

[35] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 8748–8763.

[36] H. Luo et al., "Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning," *Neurocomputing*, vol. 508, pp. 293–304, Feb. 2022.

[37] J. Xu et al., "GroupViT: Semantic segmentation emerges from text supervision," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18113–18123.

[38] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," 2022, *arXiv:2201.03546*.

[39] Q. Yang, C. Li, W. Dai, J. Zou, G.-J. Qi, and H. Xiong, "Rotation equivariant graph convolutional network for spherical image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4302–4311.

[40] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 4, pp. 2337–2348, Apr. 2018.

[41] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Learning rotation-invariant local binary descriptor," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3636–3651, Aug. 2017.

[42] J. Han, J. Ding, N. Xue, and G.-S. Xia, "ReDet: A rotation-equivariant detector for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2785–2794.

[43] X. Feng, X. Yao, G. Cheng, and J. Han, "Weakly supervised rotation-invariant aerial object detection network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14126–14135.

[44] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[45] Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, "You only look one-level feature," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13034–13043.

[46] Z. Zhou, X. Zhao, Y. Wang, P. Wang, and H. Foroosh, "Centerformer: Center-based transformer for 3D object detection," in *Computer Vision—ECCV*. Tel Aviv, Israel: Springer, 2022, pp. 496–513.

[47] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[49] Y. Zhou et al., "Mmrotate: A rotated object detection benchmark using PyTorch," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 7331–7334.

[50] Q. Ming, Z. Zhou, L. Miao, H. Zhang, and L. Li, "Dynamic anchor learning for arbitrary-oriented object detection," 2020, *arXiv:2012.04150*.

[51] Y. Xu et al., "Gliding vertex on the horizontal bounding box for multi-oriented object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 4, pp. 1452–1459, Apr. 2021.

[52] X. Yang and J. Yan, "Arbitrary-oriented object detection with circular smooth label," in *Computer Vision—ECCV*. Glasgow, U.K.: Springer, 2020, pp. 677–694.

[53] X. Pan et al., "Dynamic refinement network for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11204–11213.

[54] W. Li, Y. Chen, K. Hu, and J. Zhu, "Oriented RepPoints for aerial object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 1819–1828.

[55] Z. Guo, C. Liu, X. Zhang, J. Jiao, X. Ji, and Q. Ye, "Beyond bounding-box: Convex-hull feature adaptation for oriented and densely packed object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8788–8797.

[56] K. Chen et al., "Hybrid task cascade for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4969–4978.

[57] S. Lang, F. Ventola, and K. Kersting, "DAFNe: A one-stage anchor-free approach for oriented object detection," 2021, *arXiv:2109.06148*.

[58] C. Xu et al., "RSDD-SAR: Rotated ship detection dataset in SAR images," *J. Radars*, vol. 11, no. 4, pp. 581–599, 2022.

[59] T. Zhang et al., "SAR ship detection dataset (SSDD): Official release and comprehensive data analysis," *Remote Sens.*, vol. 13, no. 18, p. 3690, Sep. 2021.

[60] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[61] G. Ghiasi, T.-Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7029–7038.

[62] Y. Yu and F. Da, "Phase-shifting coder: Predicting accurate orientation in oriented object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 13354–13363.