

Transformation-Invariant Network for Few-Shot Object Detection in Remote Sensing Images

Nanqing Liu, Xun Xu, Turgay Celik, Zongxin Gan, Heng-Chao Li

arXiv:2303.06817v2 [cs.CV] 20 May 2023

Abstract—Object detection in remote sensing images relies on a large amount of labeled data for training. However, the increasing number of new categories and class imbalance make exhaustive annotation impractical. Few-shot object detection (FSOD) addresses this issue by leveraging meta-learning on seen base classes and fine-tuning on novel classes with limited labeled samples. Nonetheless, the substantial scale and orientation variations of objects in remote sensing images pose significant challenges to existing few-shot object detection methods. To overcome these challenges, we propose integrating a feature pyramid network and utilizing prototype features to enhance query features, thereby improving existing FSOD methods. We refer to this modified FSOD approach as a Strong Baseline, which has demonstrated significant performance improvements compared to the original baselines. Furthermore, we tackle the issue of spatial misalignment caused by orientation variations between the query and support images by introducing a Transformation-Invariant Network (TINet). TINet ensures geometric invariance and explicitly aligns the features of the query and support branches, resulting in additional performance gains while maintaining the same inference speed as the Strong Baseline. Extensive experiments on three widely used remote sensing object detection datasets, i.e., NWPU VHR-10.v2, DIOR, and HRRSD demonstrated the effectiveness of the proposed method.

Index Terms—Remote sensing images, few-shot learning, meta-learning, object detection, transformation invariance.

I. INTRODUCTION

Optical remote sensing analyzes images captured by satellites and aerial vehicles. There are huge values for analyzing these remote sensing images (RSIs), e.g. environmental monitoring [1], resource survey [2], and building extraction [3]. Detecting natural and man-made objects from RSIs is the most critical capability that supports the above analytic tasks. The state-of-the-art approaches toward object detection in RSIs [4, 5, 6, 7, 8, 9] employ a deep learning-based paradigm that requires a substantial amount of labeled data for supervised training. Nevertheless, there are several key challenges that prohibit standard supervised object detection

This work was supported in part by the National Natural Science Foundation of China under Grant 62271418, and in part by the Natural Science Foundation of Sichuan Province under Grant 23NSFSC0058.

Nanqing Liu (lansing163@163.com), Turgay Celik (celikturgay@gmail.com), Zongxin Gan (gccxeon@my.swjtu.edu.cn), Heng-Chao Li (lihengchao_78@163.com) are with the School of Information Science and Technology, Southwest Jiaotong University, Chengdu, China. Turgay Celik is also with the School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg, South Africa.

Xun Xu (xux@i2r.a-star.edu.sg) is with I2R, A-STAR, Singapore 138632, and also with the School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China.

Manuscript revised May 23, 2023.

training from scaling up. First, the existing object detection approaches [10, 11, 12] often detect the objects from the seen semantic categories while the potential objects of interest are always non-exhaustive. When new categories of objects emerge, collecting enough labeled training data for novel categories is prohibitively expensive. Moreover, there are many classes with low quantities in RSIs, as evidenced by the frequency of objects in the DIOR dataset [13] in Fig. 1 (a). This observation suggests that even if exhaustive annotation might be possible, collecting enough training examples for the minority classes is non-trivial which again motivates us to explore learning from a few labeled examples. This can help reduce the demand for annotated data and better adapt to the detection tasks of unknown and low-frequency categories. Common techniques used for limited annotation and unknown class learning include few-shot learning [14, 15, 16], zero-shot learning [17, 18], and open vocabulary learning [19, 20] and so on.

In this paper, we adopt the Few-Shot Object Detection (FSOD) paradigm to address the aforementioned challenges. In the field of object detection in natural semantic images, meta-learning-based FSOD methods [21, 22, 23, 24] have been extensively studied, which mainly consist of two branches: the query branch and the support branch. The query branch learns the object detection task from query images, while the support branch provides auxiliary information and feature representation for base and novel categories, allowing the query branch to better adapt to variations in different object categories. The interaction learning mechanism between the query and support features enables the few-shot object detection methods to have stronger generalization adaptability. However, directly applying existing meta-learning-based FSOD methods to RSIs is not the optimal choice for two reasons. Firstly, objects in images exhibit significant scale variations, and fixed-resolution object detection cannot effectively generalize to objects with large-scale variations. Secondly, the orientation variations in RSIs are more diverse than objects in natural images, as the cameras are positioned at a vertical angle, allowing arbitrary rotations in the XOY plane. This will cause spatial misalignment between the query and support images. Since the query branch learns from query images, which may have different orientations, it is difficult for it to effectively aggregate the feature representation provided by the support branch.

To address these challenges, we have made improvements to existing meta-learning-based FSOD methods. Previous meta-learning methods [21, 22] only utilize C4-layer to generate proposals from the backbone network. Therefore, to adapt to scale variations in RSIs, we introduce the Feature Pyramid

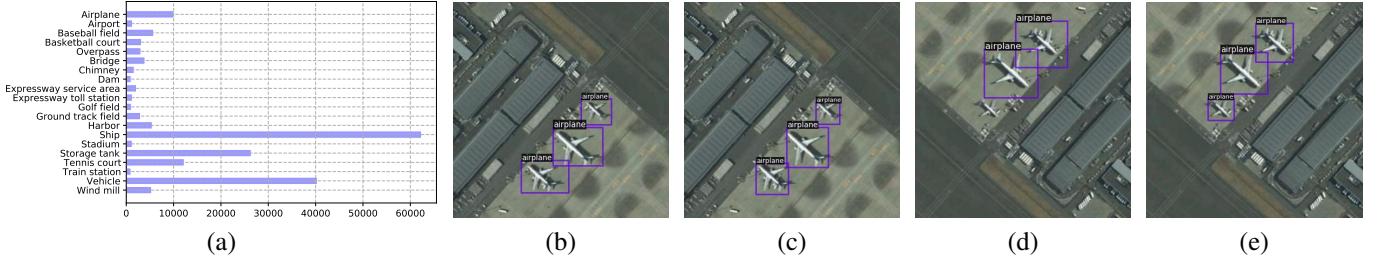


Fig. 1. (a) Number of object instances per class in the DIOR dataset. (b-e) Comparison of the detection results of the Strong Baseline and TINet. (b) Detection results of the Strong Baseline on the original image. (c) Detection results of TINet on the original image. (d) Detection results of the Strong Baseline on the diagonally flipped image. (e) Detection results of TINet on the diagonally flipped image.

Network (FPN) [11] into the existing meta-learning methods. We further propose to highlight query feature maps with support prototype features through depth-wise convolution. These methods are simple but effective, and significantly improve the performance of FSOD on RSIs. We then refer to the modified FSOD method as the Strong Baseline.

In addition to the Strong Baseline, we further address the challenges posed by large orientation variations by introducing a Transformation-Invariant Network (TINet). Specifically, we observe that the Strong Baseline cannot adapt well to object orientation. Therefore, we utilize both the query image and its transformed version as inputs to the network. Then a one-to-one consistency constraint are used to supervise the predicted bounding boxes of the original and transformed images. With these operations, the TINet is forced to produce consistent predictions on input images agnostic to camera poses. This explicitly aligns the spatial features of the query branch and the support branch. As a result, TINet can better identify objects with more pose variations. For example, in Fig. 1 (b)(c)(d)(e), we demonstrate the prediction results of the Strong Baseline and TINet on different inputs. It can be observed that TINet can adapt well to perturbations caused by transformed images. However, the Strong Baseline fails to locate all airplanes accurately.

To evaluate our proposed model, we reproduced multiple generic FSOD methods for FSOD on RSIs and we conduct extensive experiments on the DIOR [13], HRRSD [25] and NWPU VHR-10.v2 [26] datasets. The proposed TINet achieved the state-of-the-art few-shot object detection performance on all the above datasets. The main contributions of this paper are summarized as follows:

- Motivated by the large-scale variation, we proposed a Strong Baseline few-shot object detection method, which incorporates an FPN and uses 1×1 depth-wise convolution to aggregate query and support features. With these operations, the Strong Baseline improves significantly from previous meta-learning FSOD approaches.
- We propose a transformation-invariant network (TINet) based on the Strong Baseline to account for the large orientation variation. TINet only requires adding additional consistency losses between the classification and regression outputs of original and transformed images.

II. RELATED WORKS

A. Few-shot Object Detection

Few-shot Object Detection (FSOD) can be classified into two main approaches: meta-learning-based and transfer-learning-based methods. Meta-learning-based approaches, such as FSRW [23], aim to extract generalized knowledge across different tasks by learning to learn. These approaches have been extended to a two-stage network, specifically Faster-RCNN, by subsequent works [21, 22, 27], resulting in significant accuracy improvements. On the other hand, transfer-learning-based approaches follow a two-phase strategy. They are initially trained on instances of base categories and then fine-tuned on a limited number of base and novel samples. TFA [28] improves the fine-tuning process by employing a cosine similarity-based classifier to fine-tune the last layer of Faster-RCNN. FSCE [29] addresses misclassification issues by introducing contrastive learning based on TFA. DeFRCN [30] and CFA [31] enhance network performance by focusing on loss gradients. In this paper, we focus on the meta-learning-based method. However, we have observed that in the context of RSIs, conventional meta-learning-based approaches fail to achieve comparable performance to transfer-learning-based methods. This discrepancy can be attributed to the fact that meta-learning-based approaches only utilize the C4 layer for RoI pooling, while transfer-learning-based methods employ Feature Pyramid Network (FPN) to enhance multi-scale feature extraction. To address this gap, we naturally incorporate FPN into the support branch of meta-learning-based methods. Additionally, we introduce depth-wise convolution to emphasize the aggregation between support features and query features. These operations enable us to establish a Strong Baseline that achieves results comparable to transfer-learning-based methods in remote sensing images.

B. Few-shot Object Detection in Remote Sensing Images

Compared to natural semantic images, RSIs exhibit a greater diversity in the size and orientation of objects. To address these challenges, previous works in the field have introduced more advanced feature extraction modules for adapting FSOD to RSIs [32, 33, 34, 35, 36]. Additionally, researchers have approached this problem from different perspectives. For instance, Cheng et al. [37] proposed a prototype-guided Region Proposal Network (RPN) that incorporates support feature information into candidate box scores, enabling better region

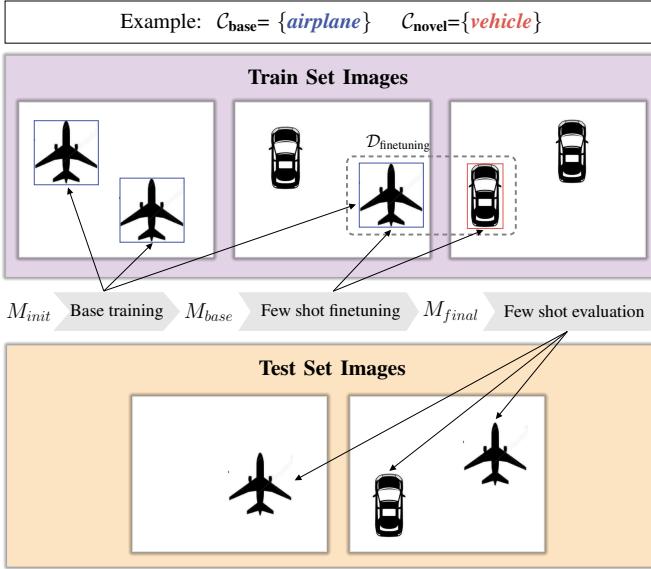


Fig. 2. Example of 2-way 1-shot few-shot object detection. The training processing can be divided into two phases (base training and few-shot finetuning). In base training, the init model M_{init} is trained on base classes (airplane), resulting in M_{base} . Then M_{base} can be transferred to M_{final} through fine-tuning on $\mathcal{D}_{finetuning}$, which includes one sample from each of the novel (vehicle) and base (airplane) classes. In the phase of evaluation, the M_{final} is applied to all the objects in the test set, which includes both novel and base classes.

proposal generation. Zhang et al. [38] employed oriented augmentation of support features to alleviate the diversity in object orientation. In contrast to these existing approaches, our method aims to improve network accuracy without compromising speed by avoiding the introduction of excessive feature extraction modules. Instead, we propose a simple modification to the network architecture by incorporating an FPN and depth-wise convolution. This modification enhances the network's capability to detect objects of diverse scales. Furthermore, to handle the diverse orientation of objects, we propose a transformation-invariant network that encourages the model to be invariant to transformations applied to input images.

C. Transformation Invariant Learning

Transformation invariant learning has been widely adopted in various domains, including natural images [39, 40, 41], remote sensing images [42, 43, 44, 45], and other scenes [46, 47, 48], which aims to enforce invariance within neural networks. There are two kinds of methods employed to achieve this objective: making the convolutional layer invariant [44, 45], or enforcing invariance through the loss function [40, 42, 46]. In this paper, we focus on the latter approach. This choice is driven by the fact that the former approach necessitates a substantial amount of data to enable the feature learning process to capture invariant features, which can be unrealistic in the few-shot setting. In the area of FSOD, TIP [40] introduces consistency regularization on predictions from various transformed images. However, its consideration is limited to classification consistency between two augmentations, restricting the use of non-geometric transformation methods

TABLE I

DIFFERENT ANALYSIS OF STRONG BASELINE ON DIOR TEST SET IN SPLIT1 (NAP). 20 AND 30 MEANS IN THE SETTING OF 20-SHOT AND 30-SHOT, RESPECTIVELY. THE FIRST ROW WITHOUT ANY STRATEGY IS EQUIVALENT TO META-RCNN [21]. WITH THE ADDITION OF DIFFERENT STRATEGIES, THE RESULTS ARE CONTINUOUSLY IMPROVED.

FPN	Depth-wise convolution	IoU threshold increasing	20	30
			27.9	30.0
✓			30.5	34.3
✓	✓		31.5	36.0
✓	✓	✓	32.1	36.8

(e.g., Gaussian noise and cutout) on input images. For remote sensing object detection tasks, regression consistency enforces the spatial locations to be consistent and should also be considered. Different from the above methods, our method incorporates this idea into FSOD in RSIs and verifies the influence of the different transformations and regularizations on the results. It is obvious that our method is the first attempt to address the obstacle of transformed variations in RSIs under few-shot settings.

III. PROPOSED METHODS

A. Problem setting

As in the previous works [21, 23, 22], we follow the standard problem settings of meta-learning-based FSOD in our paper. Specifically, the required data can be divided into two sets of categories, \mathcal{C}_{base} and \mathcal{C}_{novel} , where $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \emptyset$. The few-shot object detector aims at detecting objects of $\mathcal{C}_{base} \cup \mathcal{C}_{novel}$ by learning from a base dataset \mathcal{D}_{base} with abundant annotated objects of \mathcal{C}_{base} and a novel dataset \mathcal{D}_{novel} with very few annotated objects of \mathcal{C}_{novel} . In the task of K -shot object detection, there are exactly K annotated objects for each novel class in \mathcal{D}_{novel} . For the meta-learning approaches, the detector is trained in two phases, i.e., base training and few-shot fine-tuning. In the first phase, the init model \mathcal{M}_{init} is trained to base model \mathcal{M}_{base} only using base dataset \mathcal{D}_{base} . An episodic training scheme is applied, where each of the E episodes mimics the N -way K -shot setting. In each episode e , the model is trained on K training examples of N categories on a random subset $\mathcal{D}_{meta}^e \subset \mathcal{D}_{base}$, $|\mathcal{D}_{meta}^e| = K \cdot N$. Then, in the few-shot fine-tuning phase, the base model \mathcal{M}_{base} also adopts the same episodic training scheme as in the first phase, resulting in the final model \mathcal{M}_{final} . Different from the dataset \mathcal{D}_{base} , the fine-tuning dataset $\mathcal{D}_{finetune}$ contains K training examples from each of the categories in both the novel and base datasets. Hence, the entire training process can be simply expressed as follows:

$$\mathcal{M}_{init} \xrightarrow[e=1 \dots E]{\mathcal{D}_{meta}^e \subset \mathcal{D}_{base}} \mathcal{M}_{base} \xrightarrow{\mathcal{D}_{finetune}} \mathcal{M}_{final}. \quad (1)$$

In the phase of few-shot evaluation, the final model \mathcal{M}_{final} is applied to test datasets that contain objects from both the novel and base categories. Fig. 2 provides a visual representation of this process.

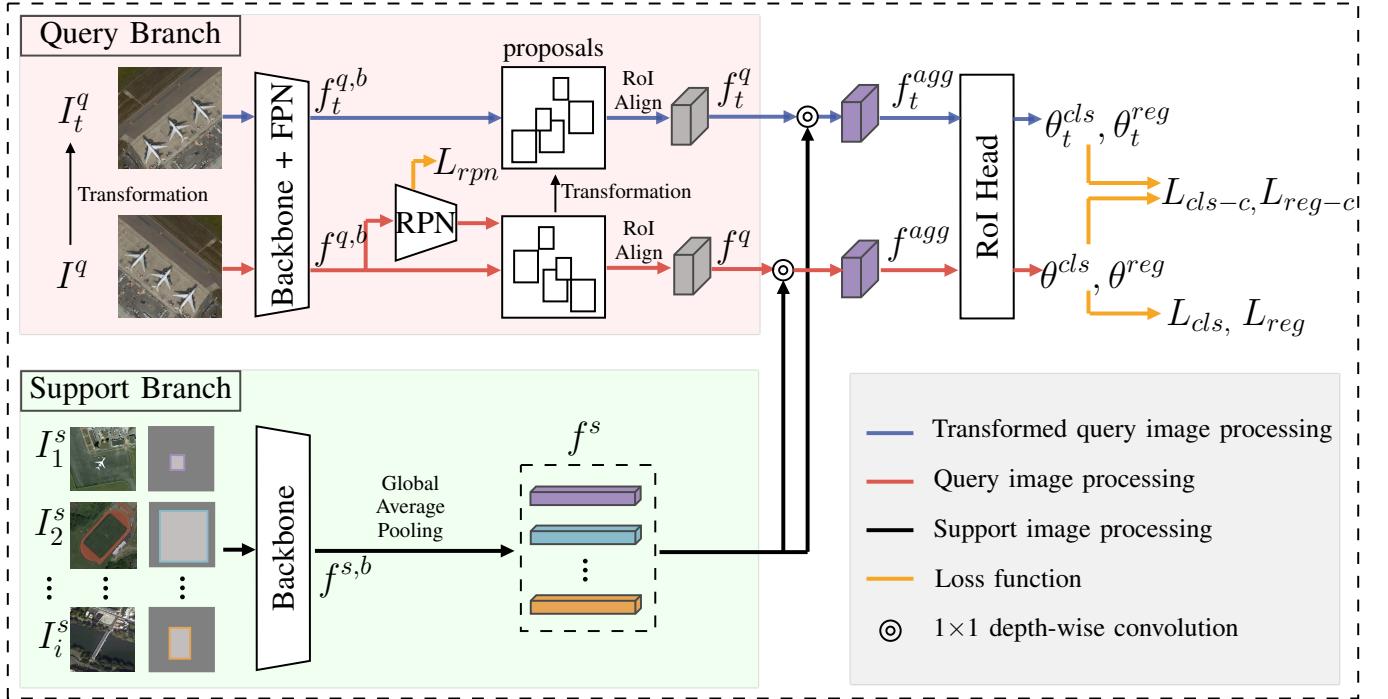


Fig. 3. The overall architecture of TINet. I^q and its transformed version I_t^q are fed into a shared backbone network and FPN to obtain query feature map $f^{q,b}$ and $f_t^{q,b}$. After subsequent processing, such as proposal generation by region proposal network (RPN) and the RoI Align, $f^{q,b}$ and $f_t^{q,b}$ are aggregated with the support feature f^s , which is generated by the support branch. The aggregated features f_t^{agg} and f_t^{qagg} are fed to the ROI Head to obtain regression parameters (θ_t^{reg} and θ_t^{reg}) and classification scores (θ_t^{cls} and θ_t^{cls}). The supervision loss of the entire network consists of the consistency loss (L_{cls-c} , L_{reg-c}) and the losses of the Faster-RCNN (L_{rpn} , L_{cls} and L_{reg}). In the testing phase, the transformed image processing branch is not used.

B. Strong Baseline

The most meta-based few-shot object detection algorithms [24, 22, 21] are built upon Faster-RCNN [12] which uses the backbone's C4 layer features for detection. Because the size variation of objects in natural images is small. So only using the C4 layer of the backbone can play a very good role. However, objects in RSIs often experience a larger variation in scale. Thus, we are motivated to incorporate multi-level feature extraction to account for the scale variation. The most intuitive idea is to add feature pyramid network (FPN) [11] to the few-shot detector e.g. Meta-RCNN [21], which has a query branch and a support branch. We only add the FPN to the query branch because complex feature fusion problems will be involved when the support branch produces multiple features. As shown in Tab. I, when FPN is added to Meta-RCNN, the performance is greatly improved, suggesting the importance of handling large-scale variation. In prior works [24, 22, 21] resize both the query feature and support feature to a size of 1×1 for element-wise multiplication. In our case, we only process the support feature, resizing it to 1×1 while keeping the query feature unchanged. This allows the ROI head to obtain more information for accurate object identification. Aggregation operation is then performed using depth-wise convolution. Additionally, during the test phase, we increase the IoU threshold in the non-maximum suppression (NMS) step of the region proposal network (RPN) from 0.7 to 0.9. This adjustment helps prevent the removal of bounding boxes due to mistakes, particularly for novel categories. From Tab. I, it can be observed that the addition of depth-wise

convolution and the increased IoU threshold slightly improve the results.

C. Transformation-Invariant Network

The proposed Strong Baseline effectively tackles the challenge of handling significant scale changes in objects. However, it falls short of addressing the issue of varying object orientations. Although data augmentation can mitigate this problem by introducing orientation transformations to the input image, it fails to address the inconsistency in aggregation between query features and support features. To overcome this limitation, we introduce a transformation-invariant few-shot object detection network (TINet) based on the Strong Baseline. The overall architecture of TINet is illustrated in Fig. 3. In the subsequent sections, we provide a comprehensive explanation of the query branch, support branch, feature aggregation, loss designs, and testing procedure employed in TINet.

1) *Query branch:* Given a query image $I^q \in \mathbb{R}^{C \times H \times W}$, we first generate its transformed version $I_t^q \in \mathbb{R}^{C \times H \times W}$. Subsequently, both I^q and I_t^q are passed through a shared backbone network and Feature Pyramid Network (FPN) to produce feature maps $f^{q,b}$ and $f_t^{q,b}$, respectively. To ensure consistency in the generated proposals, only $f^{q,b}$ is utilized as input for the Region Proposal Network (RPN). The same transformation is applied to generate transformed proposals for I_t^q . Finally, ROI features $f^q \in \mathbb{R}^{C_q \times H_q \times W_q}$ and $f_t^q \in \mathbb{R}^{C_q \times H_q \times W_q}$ are extracted using the ROI Align operation, with H_q and W_q set to 7.

2) *Support branch*: Similar to previous works [21, 23], the support branch takes 4-channel N-way K-shot support images I_i^s as input, consisting of an RGB image and its binary mask derived from the object bounding box. The support feature maps $f^s \in \mathbb{R}^{C_s \times 1 \times 1}$ are obtained through backbone feature extraction and global average pooling (GAP) operation.

3) *Feature aggregation*: For feature aggregation, we employ a 1×1 depth-wise convolution, which is both simple and effective. Specifically, the support feature $f^s \in \mathbb{R}^{C_s \times H_s \times W_s}$ serves as a convolution kernel for a 1×1 depth-wise convolution. The resulting aggregated features, denoted as f^{agg} and f_t^{agg} , are then fed into the RoI Head to obtain classification scores θ^{cls} , θ_t^{cls} , and regression parameters θ^{reg} , θ_t^{reg} .

Algorithm 1 Consistency loss computation

Input: Query image I^q and its transformed version I_t^q support images I_1^s, \dots, I_i^s .

Output: Consistency loss L_{reg-c} , L_{cls-c} .

Query Branch:

- 1: Feed I^q and I_t^q to the shared backbone and FPN to obtain output features $f^{q,b}$ and $f_t^{q,b}$;
- 2: Generate proposals using $f^{q,b}$ by RPN and obtain the flipped proposals;
- 3: Extract query box feature f_t^q and f^q by RoI Align.

Support Branch:

- 4: Feed I_1^s, \dots, I_i^s to the shared backbone to obtain output feature $f_1^{s,b}, \dots, f_i^{s,b}$;
 - 5: Generate support features f^s by GAP;
 - 6: Random sample a support feature f_r^s from f^s .
 - 7: $f_t^{agg} \leftarrow f_t^q \otimes f_r^s$; $f^{agg} \leftarrow f^q \otimes f_r^s$;
 - 8: Feed f_t^{agg} and f^{agg} to RoI head to obtain θ^{cls} and θ^{reg} , θ_t^{cls} and θ_t^{reg} ;
 - 9: Compute L_{cls-c} according to Eq. (2) and L_{reg-c} according to Eq. (3).
-

4) *Consistency loss*: By minimizing the consistency loss, the network can perform consistency detection results on both query image processing and its transformation version so that the aggregation feature f^{agg} and f_t^{agg} are forced to the same distribution. The consistency loss comprises two components: the classification consistency loss L_{cls-c} and the regression consistency loss L_{reg-c} . Let σ^j and σ_t^j denote the classification distribution for the j -th proposal in θ^{cls} and θ_t^{cls} , respectively. Different metrics, such as L_2 loss, Jensen–Shannon divergence (JSD), and Kullback–Leibler divergence (KLD), can be used to measure the distance between these distributions. Through experiments, we find that L_2 loss performs the best (as discussed in Section IV-D3). Therefore, the classification consistency loss can be defined as follows:

$$L_{cls-c} = \sum_{j=1}^J \|\sigma^j - \sigma_t^j\|^2 \quad (2)$$

In contrast to the classification distribution, regression parameters change with image transformation. Let $[\Delta x^j, \Delta y^j, \Delta w^j, \Delta h^j]$ represent the regression results for the j -th proposal in θ^{reg} , representing the offset of the center point and the scale coefficients for width and

height. Similarly, $[\Delta x_t^j, \Delta y_t^j, \Delta w_t^j, \Delta h_t^j]$ represents the j -th parameters in θ_t^{reg} . In this paper, we focus on the effect of three flipping transformations: horizontal flipping, vertical flipping, and diagonal flipping. Since the flipping transformation causes Δx_t^j or Δy_t^j to move in the opposite direction, a negative operation is applied to correct it. For example, in diagonal flipping, Δx^j and Δy^j correspond to $-\Delta x_t^j$ and $-\Delta y_t^j$. The regression consistency loss with L_2 regularization can be defined as:

$$\begin{aligned} L_{reg-c} = & \sum_{j=1}^J \left[\left\| \Delta x^j - (-\Delta x_t^j) \right\|^2 + \left\| \Delta y^j - (-\Delta y_t^j) \right\|^2 \right. \\ & \left. + \left\| \Delta w^j - \Delta w_t^j \right\|^2 + \left\| \Delta h^j - \Delta h_t^j \right\|^2 \right] \end{aligned} \quad (3)$$

For the other two flipping transformations, Δx^j and Δy^j should correspond to $-\Delta x_t^j$ and Δy_t^j for horizontal flipping and Δx_t^j and $-\Delta y_t^j$ for vertical flipping.

The complete procedure to compute the consistency loss is presented in Algorithm 1. Both the base training phase and the few-shot fine-tuning phase utilize the consistency loss.

5) *Total loss*: We optimize the total loss L_T by combining the standard supervised loss terms with the consistency losses, as follows:

$$L_T = L_{rpn} + L_{cls} + L_{reg} + \lambda_1 L_{cls-c} + \lambda_2 L_{reg-c} \quad (4)$$

Here, L_{rpn} , L_{cls} , and L_{reg} represent the losses (cross-entropy loss and L1 loss) for the Region Proposal Network (RPN), which are the classification loss and regression loss for the original query image, respectively. The terms λ_1 and λ_2 control the strength of the transformation consistency. As object detection consistency is not stable in the early training stage, we choose relatively smaller weights: $\lambda_1 = 0.05$ and $\lambda_2 = 0.02$.

6) *Testing procedure*: During training, the support feature is randomly selected for each iteration. However, during testing, all support features must be involved in the process. The testing phase is outlined in Algorithm 2. Importantly, the transformed images are not included in the testing phase, ensuring that there is no impact on the overall inference time. The impact of different components in the consistency loss on both training and testing time is discussed in Section IV-D6.

IV. EXPERIMENTS

A. Dataset and Experimental Setting

We conduct experiments on three extensively used remote sensing datasets, i.e., NWPU VHR-10.v2 [26], DIOR [13] and HRRSD [25]. NWPU VHR-10.v2 contains 1172 annotated images distributed into ten categories, which are divided into 75% for training and 25% for testing. For the DIOR dataset, 11,725 images are used as the training set, and the remaining 11,738 images are employed as the test set. Likewise, the HRRSD data are divided into three parts (the training, validation, and test sets), with 5,401, 5,417, and 10,913 images, respectively.

To establish the few-shot learning setup, we further divide each dataset into two parts, the novel class and the base class

TABLE II
DIFFERENT NOVEL/BASE CLASSES SPLIT SETTINGS FOR OUR EXPERIMENTS.

Dataset(split)	Novel classes	Base classes
DIOR(split1)	airplane, baseball field, train station, tennis court, windmill	rest
DIOR(split2)	airplane, airport, expressway toll station, harbor, ground track field	rest
HRRSD	airplane, baseball diamond, ground track field, storage tank	rest
NWPU VHR-10.v2	airplane, baseball diamond, tennis court	rest

TABLE III
COMPARING RESULTS OF DIFFERENT FEW-SHOT OBJECT DETECTION METHODS ON THE DIOR TEST SET IN SPLIT1(NAP/BAP/MAP). COLORED RESULTS REPRESENT THE **BEST** AND **SECOND-BEST**. * INDICATES RESULTS REPORTED IN [32].

Method	Backbone	Combination	5-shot			10-shot			20-shot			30-shot		
			nAP	bAP	mAP	nAP	bAP	mAP	nAP	bAP	mAP	nAP	bAP	mAP
FRCN-ft [12]	ResNet-50	FPN	15.9	33.7	29.3	20.4	43.4	37.7	24.8	49.4	43.2	26.5	50.7	44.7
FsDetView [22]	ResNet-50	C4	17.0	37.8	32.6	21.9	39.7	35.3	24.9	41.8	37.6	27.6	46.7	41.9
TFA [28]	ResNet-50	FPN	19.4	36.1	31.9	23.3	40.9	36.5	30.1	48.6	43.9	34.0	50.3	46.2
FSCE [29]	ResNet-50	FPN	20.3	38.0	33.6	26.2	43.5	39.1	34.0	49.6	45.7	37.2	53.4	49.3
Meta-RCNN [21]	ResNet-50	C4	20.7	47.1	40.5	24.7	46.7	41.3	27.9	48.1	43.0	30.0	49.3	44.5
RepMet* [49]	InceptionV3	-	8.0	-	-	14.0	-	-	16.0	-	-	-	-	-
FSRW* [23]	Darknet-19	-	22.0	-	-	28.0	-	-	34.0	-	-	-	-	-
FSODM* [32]	Darknet-53	-	25.0	-	-	32.0	-	-	36.0	-	-	-	-	-
Zhang et al. [36]	ResNet-101	FPN	34.0	-	-	37.0	-	-	42.0	-	-	-	-	-
Strong Baseline (Ours)	ResNet-50	FPN	22.0	48.0	41.5	26.9	52.1	45.8	32.1	55.5	49.6	36.8	55.4	50.7
TINet (Ours)	ResNet-50	FPN	29.5	56.2	49.5	35.2	56.8	51.4	41.6	59.8	55.3	42.8	62.6	57.7
TINet (Ours)	ResNet-101	FPN	28.6	57.8	50.5	38.4	57.4	52.7	43.2	62.1	57.4	44.6	63.6	58.9

Algorithm 2 The overall test procedure (one image)

Input: Query image I^q and support images I_1^s, \dots, I_i^s .
Output: Classification score θ_i^{cls} , regression parameters θ_i^{reg} .

Query Branch:

- 1: Feed I^q to the shared backbone and FPN to obtain output feature $f^{q,b}$;
- 2: Generate proposals using $f^{q,b}$ by RPN;
- 3: Extract query box feature f^q by RoI Align.

Support Branch:

- 4: Feed I_1^s, \dots, I_i^s to shared backbone to obtain output feature $f_1^{s,b}, \dots, f_i^{s,b}$;
- 5: Generate support features f^s by GAP.
- 6: **for** $i \leftarrow 1$ to N **do**
- 7: $f_i^{agg} \leftarrow f^q \otimes f_i^s$;
- 8: Feed f_i^{agg} to RoI head to obtain θ_i^{cls} and θ_i^{reg} .
- 9: **end for**

following the practice adopted in [32][37]. A detailed split setting is presented in Tab. II. It should be noted that the number of shots denotes the number of instances that are not images because one image contains several instances. We evaluate the testing images, which contain both base and novel classes.

For all experiments conducted with our proposed detector, we utilized a ResNet [51] backbone network pre-trained on ImageNet. During the training process, we employed an SGD optimizer with a momentum coefficient of 0.9 and a weight decay of 0.0001. The batch size was set to 4 for all datasets. In the base training stage, the initial learning rate was set to 0.01, with a 0.1 decrease at 80% of the total iterations. In the few-

TABLE IV
COMPARING RESULTS OF DIFFERENT FSOD METHODS ON THE DIOR TEST SET IN SPLIT2 (NAP). COLORED RESULTS REPRESENT THE **BEST** AND **SECOND-BEST**. * INDICATES RESULTS REPORTED IN [37],

Method	Backbone	5	10	20	30
FRCN-ft [12]	ResNet-50	14.9	17.6	22.8	23.5
FsDetView [22]	ResNet-50	14.2	16.2	19.1	21.9
TFA [28]	ResNet-50	18.0	20.9	23.0	26.4
FSCE [29]	ResNet-50	19.9	22.7	26.9	30.6
Meta-RCNN [21]	ResNet-50	14.1	17.6	21.0	21.2
RepMet* [49]	InceptionV3	5.6	5.9	6.8	6.5
FSRW* [23]	DarkNet-19	7.0	9.0	14.1	14.4
P-CNN* [37]	ResNet-101	14.9	18.9	22.8	25.7
Zhang et al. [36]	ResNet-101	15.5	19.7	23.8	29.6
G-FSDet [50]	ResNet-101	15.8	20.7	22.7	-
Strong Baseline	ResNet-50	20.1	23.3	26.5	28.1
TINet	ResNet-50	21.7	24.1	28.0	31.9
TINet	ResNet-101	22.8	25.1	29.4	33.2

shot fine-tuning stage, the initial learning rate was set to 0.001, with a 0.1 decrease at 80% of the total iterations. For NWPU VHR-10.v2, we trained for 9,000 iterations in the base training stage and 3,000 iterations in the few-shot fine-tuning stage. For HRRSD and DIOR, we trained for 36,000 iterations in the base training stage and 6,000 iterations in the few-shot fine-tuning stage. Additionally, we employed multiscale training and random flipping to enhance the detection performance. The scale range of the input images varies (440, 472, 504, 536, 568, 600). We perform the experiments under the PyTorch framework on a PC with an Intel single-core i7 CPU and a GeForce RTX 3090 GPU.

TABLE V
COMPARISON RESULT OF DIFFERENT FEW-SHOT OBJECT DETECTION METHODS ON HRRSD TEST SET (NAP/BAP/mAP). COLORED RESULTS REPRESENT THE **BEST** AND **SECOND-BEST**.

Method	Backbone	Combination	5-shot			10-shot			20-shot			30-shot		
			nAP	bAP	mAP	nAP	bAP	mAP	nAP	bAP	mAP	nAP	bAP	mAP
FRCN-ft [12]	ResNet-50	FPN	26.9	79.4	63.3	38.1	80.8	67.7	44.0	82.1	70.4	46.2	82.9	71.6
FsDetView [22]	ResNet-50	C4	35.6	62.2	54.0	42.0	67.8	59.8	48.1	69.8	63.1	52.8	70.0	64.7
TFA [28]	ResNet-50	FPN	36.0	75.3	63.2	45.1	79.3	68.8	51.4	80.7	71.7	53.0	81.0	72.4
FSCE [29]	ResNet-50	FPN	37.5	75.7	64.0	46.3	79.8	69.5	54.5	80.9	72.8	61.9	80.6	74.9
Meta-RCNN [21]	ResNet-50	C4	30.5	71.1	58.6	41.1	73.2	63.3	47.7	73.7	65.7	51.5	75.5	68.1
Strong Baseline	ResNet-50	FPN	32.3	71.2	59.2	43.7	75.4	65.6	53.3	75.1	68.4	61.4	77.2	72.3
TINet	ResNet-50	FPN	38.3	81.8	68.4	47.3	80.2	70.4	58.9	80.5	73.9	64.3	80.8	75.3

TABLE VI
COMPARISON RESULT OF DIFFERENT FSOD METHODS ON NWPU VHR-10.v2 TEST SET (NAP). COLORED RESULTS REPRESENT THE **BEST** AND **SECOND-BEST**.

Method	Backbone	2	3	5	10
FRCN-ft [12]	ResNet-50	35.1	44.8	48.9	57.3
FsDetView [22]	ResNet-50	40.8	52.2	58.6	65.2
TFA [28]	ResNet-50	42.8	50.7	53.1	60.5
FSCE [29]	ResNet-50	53.4	56.4	60.6	68.7
Meta-RCNN [21]	ResNet-50	43.1	50.6	55.1	62.6
OFA [38]	ResNet-101	34.0	43.2	60.4	66.7
G-FSDet [50]	ResNet-101	-	49.1	56.1	71.8
Strong Baseline	ResNet-50	45.8	55.1	59.8	64.0
TINet	ResNet-50	53.7	55.8	63.5	71.8

In the subsequent experimental results, we adopt the evaluation protocol of the PASCAL visual object classes (VOC) [52]. The mean average precision (mAP) represents the average precision across all object categories, including both base and novel categories. The novel class average precision (nAP) indicates the average precision for the novel categories, while the base class average precision (bAP) indicates the average precision for the base categories.

B. Reproducing Generic FSOD Methods

To make a fair comparison, we first reproduce several state-of-the-art generic few-shot object detection methods based on the open-source framework MMFewShot [53] which is tailored for few-shot learning. The reproduced methods include FRCN-ft [12], TFA [28], FSCE [29], Meta-RCNN [21], FsDetView [22], as well as our proposed Strong Baseline and TINet. Specifically, the Strong Baseline is referred to in Section III-B. FRCN-ft only uses base class objects to train the Faster-RCNN with FPN in the first phase and then uses combinations of the base class and novel class objects to fine-tune in the second phase. For TFA, we only freeze the backbone in the fine-tuning phase because we can obtain better results in this way, which is slightly different from the original paper. For FSCE, Meta-RCNN, and FsDetView, we keep the same setting as the original paper.

C. Few-Shot Object Detection Results

1) *DIOR*: We present the results of different methods for split1 and split2 in Tab. III and Tab. IV respectively. In split1,

as shown in Tab. III, in addition to the reproduced generic few-shot object detection methods, we further incorporate comparisons with RepMet [49] with InceptionV3 [54] as the backbone and FSRW with DarkNet-19 [55] as backbone according to [32]. We make the following observations from the results. First, the TINet outperforms all competing methods except for nAP@5shot. The gap is particularly significant compared with generic FSOD methods where more than 10% improvements in nAP/bAP/mAP are observed throughout 5 to 30 shots. This suggests the strong few-shot learning capability of TINet. Second, all the transfer-learning approaches and meta-learning approaches outperform the original fine-tuned Faster-RCNN (FRCN-ft). For the meta-learning approaches, we observe that Meta-RCNN and FsDetView only use the C4 layer for subsequent processing, thus they perform relatively worse compared with the Strong Baseline on the DIOR dataset which features large diversity in object scales. For the transfer-learning-based approaches, both FSCE and TFA are way behind TINet despite all are using FPN, probably due to the large intra-class variation in the DIOR dataset. Finally, compared with the results reported by the state-of-the-art methods in RSIs [32][36], we also demonstrate a competitive result, except for the nAP@5 shots. We further present the comparisons for split2 in Tab. IV, which is generally considered to be more challenging than split1. Under split2, we observe more significant improvements in TINet from the best-performing methods. These results again validate the effectiveness of the proposed method.

2) *HRRSD*: The quantitative results obtained by applying different methods to the HRRSD dataset are presented in Table V. Since no previous algorithms have been tested on the HRRSD dataset, we can only compare the results with our implemented algorithm. From the results, we can observe that most methods achieve good performance on this dataset, which is less complex compared to the DIOR dataset and has fewer object categories. Transfer learning approaches, especially FSCE, demonstrate strong performance in certain aspects. However, our method still outperforms FSCE in terms of nAP at 5 shots, 10 shots, 20 shots, and 30 shots, with improvements of 0.8%, 1.0%, 4.4%, and 2.4%, respectively.

3) *NWPU VHR-10.v2*: As shown in Table VI, since the NWPU VHR-10.v2 dataset is relatively simple, all the methods achieve relatively good results. It can be observed that our Strong Baseline has higher results than the general meta-learning methods but lower results than the transfer learning

TABLE VII

COMPARISON EXPERIMENTS WITH DATA AUGMENTATION ON DIOR TEST SET IN SPLIT1 (NAP). 5, 10, 20, AND 30 MEAN IN THE SETTING OF 5-SHOT, 10-SHOT, 20-SHOT, AND 30-SHOT, RESPECTIVELY. DATA AUG REPRESENTS DATA AUGMENTATION. THE EXPERIMENTAL RESULTS IN THE FIRST ROW ARE OBTAINED WITHOUT ANY STRATEGY.

Data Aug	L_{cls-c}	L_{reg-c}	5	10	20	30
✓	✓	✓	20.8	23.8	30.1	36.3
			22.0	26.9	32.1	36.8
			29.1	33.9	40.5	40.9
	✓	✓	27.8	31.8	39.3	41.6
		✓	28.8	34.3	41.1	42.2
✓	✓	✓	29.5	35.2	41.6	42.8

approach, FSCE. This is because this dataset is very small and the intra-class similarity is not large. TINet still outperforms FSCE on most metrics. Although OFA [38] improves object recognition in novel categories by rotating the support samples, increases the inference time, and does not use the oriented feature augmentation method in the base training phase, which may reduce the generalization performance.

D. Ablation Study

We conduct ablation experiments on the DIOR dataset (split1) to reveal the effectiveness of each individual component. Unless otherwise specified, the backbone network chosen is ResNet-50.

1) *Comparison with data augmentation:* There are two consistency losses (L_{cls-c} and L_{reg-c}) in the TINet. As shown in Tab. VII, we examine the influence of L_{cls-c} and L_{reg-c} and make a comparison with data augmentation, which augments the image before feeding it into the query branch. The experimental results in first row are obtained without any strategy. It should be noted that for data augmentation, we choose a combination of horizontal, vertical, and diagonal flipping. The consistency loss L_{cls-c} and L_{reg-c} here are both the L_2 loss, and the corresponding flipping method is diagonal flipping. It can be observed that data augmentation improves the performance of the network. However, as the number of shots increases, the impact of data augmentation diminishes significantly. On the other hand, adding only the consistency losses L_{cls-c} and L_{reg-c} outperforms the use of data augmentation alone. When either L_{cls-c} or L_{reg-c} is added, the network achieves stable improvements in performance, except in the 5-shot scenario. The best results are obtained when both the consistency losses and data augmentation are used together, indicating that these two techniques are complementary to each other.

2) *Alternative transformations:* We verify the effect of the different flipping transformations on the experimental results. As shown in Tab. VIII, we observe similar results for both the horizontal and vertical flips. The result of the diagonal flip is slightly better than the previous two. This is because the diagonal flip introduces fewer changes to the object's appearance so that the training is more stable compared to the horizontal and vertical flips.

TABLE VIII

DETECTION RESULTS OF ALTERNATIVE TRANSFORMATIONS ON DIOR TEST SET IN SPLIT1 (NAP). 5, 10, 20 AND 30 MEANS IN THE SETTING OF 5-SHOT, 10-SHOT, 20-SHOT AND 30-SHOT, RESPECTIVELY.

Flipping method	5	10	20	30
None	22.0	26.9	32.1	36.8
Vertical	28.1	34.1	39.1	41.3
Horizontal	28.5	34.5	39.4	41.2
Diagonal	29.5	35.2	41.6	42.8

3) *Alternative consistency regularizations.:* We verify the effect of different regularizations in the classification consistency loss L_{con-c} on the results. As shown in Tab. IX, the JSD and KLD represents Jensen–Shannon divergence and Kullback–Leibler divergence, respectively. The weight of JSD and KLD here we chose are 0.05 and 0.1. It can be observed that simply using the L_2 loss yields the best results in most metrics except in 5-shot. Because the L_2 loss is more sensitive to outliers, it can play a more restrictive role.

4) *Alternative hyper-parameters of loss L_T :* General object detectors always focus on two main sub-tasks (regression and classification) so that the weight of auxiliary losses should be relatively smaller. We carried out evaluations on the robustness of the choice of λ s in Tab. X. In general, stable performance is observed around the hyper-parameters we chose.

TABLE IX

DETECTION RESULTS WITH ALTERNATIVE CONSISTENCY REGULARIZATIONS ON DIOR TEST SET IN SPLIT1 (NAP). 5, 10, 20, AND 30 MEAN IN THE SETTING OF 5-SHOT, 10-SHOT, 20-SHOT, AND 30-SHOT, RESPECTIVELY.

Regularization method	5	10	20	30
None	22.0	26.9	32.1	36.8
JSD	27.8	34.8	38.1	41.4
KLD	30.3	34.7	37.9	40.2
L_2	29.5	35.2	41.6	42.8

5) *Pearson correlation coefficient:* We further measure the calibration of detection models by the Pearson Correlation Coefficient (PCC), defined as follows:

$$r = \frac{\sum (x - m_x)(y - m_y)}{\sqrt{\sum (x - m_x)^2 \sum (y - m_y)^2}} \quad (5)$$

x and y are respectively the IoU between ground-truth and predicted bounding boxes and the confidence score (the highest posterior). A high correlation indicates the confidence is well calibrated. m_x and m_y are the mean of the x and y respectively. The results (shown in Fig. 4) demonstrate that the TINet obtained a higher value of PCC than the Strong Baseline and Meta-RCNN in all the datasets, suggesting TINet is better calibrated than others.

6) *Training and testing time:* The results are shown in Tab. XI. Here one iteration includes four multi-scale images, and in the testing phase, the images are resized to 600×600 . It can be observed that the training time of TINet increases

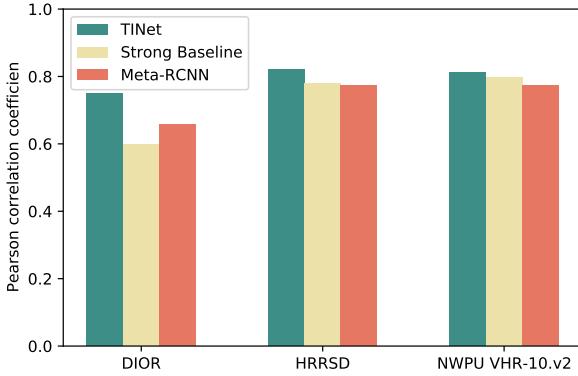


Fig. 4. Pearson correlation coefficient (PCC) between classification scores and IoUs.

TABLE X
COMPARING DIFFERENT λ ON DIOR TEST SET IN SPLIT1 (NAP) UNDER 5, 10, 20, AND 30 SHOTS.

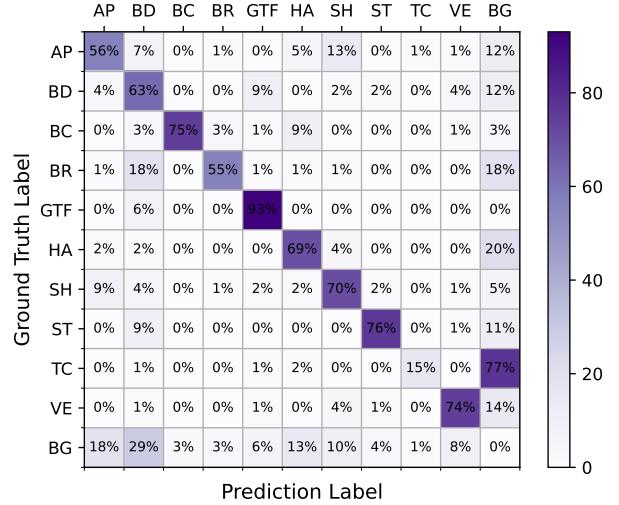
λ_1	λ_2	5	10	20	30
1	1	8.6	12.2	16.7	24.8
0.5	0.5	25.4	31.6	37.7	38.1
0.05	0.05	29.0	34.7	42.2	42.1
0.05	0.02	29.5	35.2	41.6	42.8
0.02	0.05	29.2	35.0	41.3	42.4
0.02	0.02	29.1	34.7	40.9	41.8

slightly compared with Strong Baseline because TINet has to process two images simultaneously. However, the inference time of all the comparison methods remains the same. In addition, the loss computation has little effect on the time of network training.

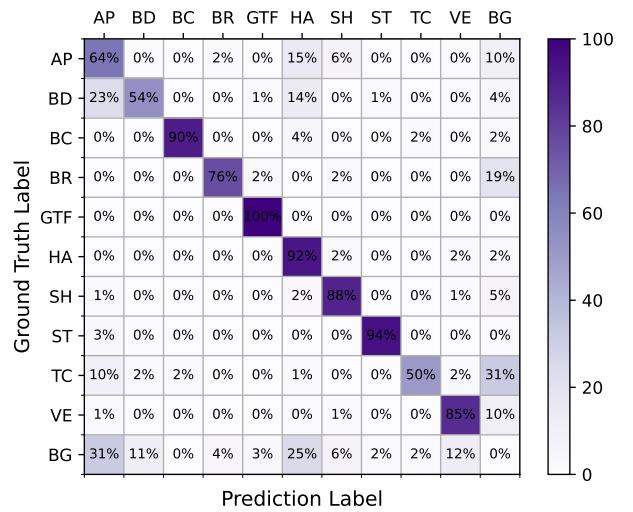
7) *Comparison with other transformation invariant methods:* We finally compared our method with two representative transformation invariant methods. ReResNet [44] achieves invariance by extracting rotation-invariant features, while TIP introduces Cutout and Gaussian Noise into the input images and utilizes consistency loss for achieving invariance. From Table XII, we can observe that the performance of the Strong Baseline deteriorates significantly after incorporating ReResNet. This is because ReResNet lacks a sufficient number of samples for training in few-shot setting, leading to convergence issues. Moreover, the inference speed is significantly slower when using ReResNet (4.4 FPS compared to 15.5 FPS).

TABLE XI
THE COMPARISON OF TRAINING AND TESTING TIME BETWEEN THE STRONG BASELINE AND TINET.

Method	Training (iteration/s)	Testing (frame/s)
Strong Baseline	4.92	15.5
TINet(w/o L_{cls-c})	4.29	15.5
TINet(w/o L_{reg-c})	4.35	15.5
TINet	4.27	15.5



(a)



(b)

Fig. 5. Comparison of confusion matrix between the Strong Baseline and TINet. (a) Confusion matrix of the Strong Baseline. (b) Confusion matrix of TINet. The abbreviations for the categories are AP-airplane, BD-baseball diamond, BC-basketball court, BR-bridge, GTF-ground track field, HA-harbour, SH-ship, ST-storage tank, TC-tennis court, VE-vehicle, and BG-background.

TABLE XII
COMPARING WITH OTHER TRANSFORMATION INVARIANT METHODS ON THE DIOR TEST SET IN SPLIT1 (NAP).* REPRESENTS WHAT WE REPRODUCED, AND MAY DIFFER SLIGHTLY FROM THE ORIGINAL PAPER

Method	Backbone	5	10	20	30	FPS
Strong Baseline	ReResNet[44]	12.1	15.3	18.7	21.0	4.4
TIP*[40]	ResNet	26.6	33.4	40.1	41.3	15.5
TINet(ours)	ResNet	29.5	35.2	41.6	42.8	15.5

As TIP [40] did not publish their source code, we managed to replace our geometric transformation with Cutout and Gaussian Noise, as proposed in TIP. The results suggest that geometric transformation is significantly superior to Cutout and Gaussian Noise, especially in the low-shot regime.



Fig. 6. Detection results of different methods of sample images including the detected novel objects and a small number of base objects on DIOR dataset (30-shot at split1).

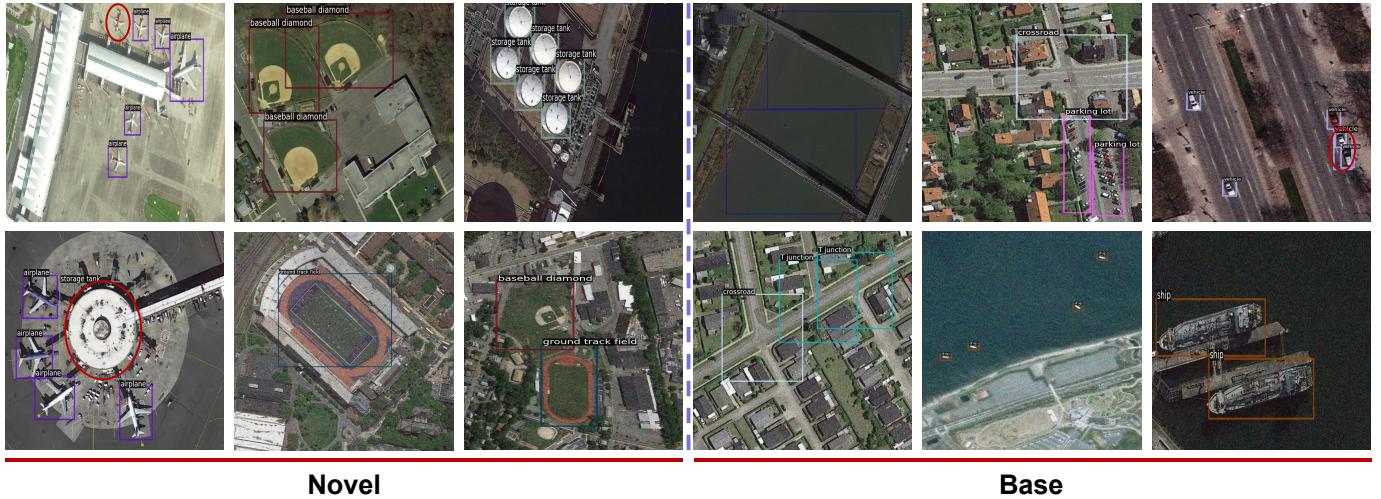


Fig. 7. Detection results of the TINet for the novel and base objects in HRRSD dataset (30-shot). Objects of different categories are represented by rectangular boxes of different colors. Failure cases are represented by red ellipses.

E. Visualization

To more intuitively demonstrate the effectiveness of our method, we visualize the confusion matrix and prediction results.

1) *Confusion matrix:* We generated a confusion matrix using the detection results of the NWPU VHR-10.v2 test set. The abbreviations for the categories are as follows: AP-airplane, BD-baseball diamond, BC-basketball court, BR-bridge, GTF-ground track field, HA-harbour, SH-ship, ST-storage tank, TC-tennis court, VE-vehicle, and BG-background. Unlike the classification task, the detection task involves cases of false positives and missing detections, making the inclusion of a background class necessary to cover all cases. It should be noted that the percentage sum along the horizontal axis is 100%, while the vertical axis is not 100% due to normalization based on the horizontal axis. From Fig. 5, it can be observed that the Strong Baseline has a low probability of correctly recognizing the novel classes (AP, BD, and TC). Our method, on the other hand, alleviates this problem to some extent. Additionally, it is worth mentioning that our method does not forget the characteristics of the base class while training on the novel class.

2) *Prediction results:* As shown in Fig. 6, We present a comparison of several FSOD methods on the DIOR dataset of 30-shot at split1, which contains objects of the novel category, including airplanes, baseball fields, train stations, tennis courts, and windmills, as well as a small number of objects from the base category.

The results highlight the strong generalization ability of our proposed method, TINet, attributed to its multi-scale feature structure and transformation invariant learning. Notably, we observe that for smaller objects like airplanes and windmills, Meta-RCNN without the FPN structure performs even worse than the original fine-tuning Faster-RCNN (FRCN-ft). However, the incorporation of the Strong Baseline significantly enhances the detection performance of Meta-RCNN, leading to similar results as FSCE. Moreover, leveraging the

transformation invariant strategy atop the Strong Baseline, TINet further improves the detection performance for objects with varying orientations, such as airplanes and tennis courts. For simpler objects like baseball fields, which lack scale and orientation diversity, all the compared algorithms achieve comparable detection results. Overall, our TINet outperforms all competing methods, producing the best detection results.

Furthermore, we provide additional qualitative results in Fig. 7, encompassing both novel and base objects on the HRRSD (30-shot) dataset. This dataset exhibits less complexity compared to DIOR, resulting in fewer false detections. For example, the appearance of the circular aircraft waiting hall is very similar to the storage tank, so it caused false detection. And the color of the missing airplane is overlaid with more red, which makes it different from other airplanes. Hence, this phenomenon motivates us to focus on designing modules to extract more discriminative features in future work.

V. CONCLUSION

In this paper, in light of the challenges in few-shot object detection (FSOD) for remote sensing images (RSIs), we first propose to modify from existing meta-learning-based FSOD method by incorporating FPN and depth-wise convolution. To improve the network's ability to align the feature of support branch and query branch, we further propose to incorporate transformation invariance into the baseline, which is then referred to as TINet. Extensive experiments demonstrate the effectiveness of our method, and the method achieved state-of-the-art performances on the vast majority of the metrics on three widely used optical remote sensing object detection datasets, i.e., NWPU VHR-10.v2, DIOR, and HRRSD. It is worth noting that our work is to demonstrate that the improvement of the FSOD in RSIs by geometric transformation is significant. In general, more geometric transformations may further improve performance, such as arbitrary rotation, scaling, translation, etc, which will be considered in detail in our future work. Among them, arbitrary rotation transformation

may introduce an artificial black border area and the risk of GT information leakage, which requires a special design.

REFERENCES

- [1] K. Chen, W. Li, S. Lei, J. Chen, X. Jiang, Z. Zou, and Z. Shi, "Continuous remote sensing image super-resolution based on context interaction in implicit function space," *IEEE Transactions on Geoscience and Remote Sensing*, 2023.
- [2] N. Liu, T. Celik, and H.-C. Li, "Msnet: a multiple supervision network for remote sensing scene classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2020.
- [3] K. Chen, Z. Zou, and Z. Shi, "Building extraction from remote sensing images with sparse token transformers," *Remote Sensing*, vol. 13, no. 21, p. 4441, 2021.
- [4] Y. Yao, G. Cheng, G. Wang, S. Li, P. Zhou, X. Xie, and J. Han, "On improving bounding box representations for oriented object detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023.
- [5] N. Liu, T. Celik, T. Zhao, C. Zhang, and H.-C. Li, "Afdet: Toward more accurate and faster object detection in remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.
- [6] X. Yang, L. Hou, Y. Zhou, W. Wang, and J. Yan, "Dense label encoding for boundary discontinuity free rotation detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [7] G. Cheng, Y. Si, H. Hong, X. Yao, and L. Guo, "Cross-scale feature fusion for object detection in optical remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, 2021.
- [8] N. Liu, T. Celik, and H.-C. Li, "Gated ladder-shaped feature pyramid network for object detection in optical remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, 2022.
- [9] J. Han, J. Ding, J. Li, and G.-S. Xia, "Align deep features for oriented object detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [10] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, vol. 111, no. 3, pp. 257–276, 2023.
- [11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, 2015.
- [13] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS Journal of Photogrammetry and Remote Sensing*, 2020.
- [14] G. Cheng, L. Cai, C. Lang, X. Yao, J. Chen, L. Guo, and J. Han, "Spnet: Siamese-prototype network for few-shot remote sensing image scene classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–11, 2021.
- [15] C. Lang, G. Cheng, B. Tu, C. Li, and J. Han, "Base and meta: A new perspective on few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [16] G. Cheng, C. Lang, and J. Han, "Holistic prototype activation for few-shot segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [17] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 384–400.
- [18] X. Xu, T. Hospedales, and S. Gong, "Transductive zero-shot action recognition by word-vector embedding," *International Journal of Computer Vision*, vol. 123, pp. 309–333, 2017.
- [19] K. Chen, X. Jiang, Y. Hu, X. Tang, Y. Gao, J. Chen, and W. Xie, "Ovarnet: Towards open-vocabulary object attribute recognition," *arXiv preprint arXiv:2301.09506*, 2023.
- [20] A. Zareian, K. D. Rosa, D. H. Hu, and S.-F. Chang, "Open-vocabulary object detection using captions," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 393–14 402.
- [21] X. Yan, Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin, "Meta r-cnn: Towards general solver for instance-level low-shot learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [22] Y. Xiao and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," in *European conference on computer vision*, 2020.
- [23] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.
- [24] J. Han, Y. Ren, J. Ding, K. Yan, and G.-S. Xia, "Few-shot object detection via variational feature aggregation," *arXiv preprint arXiv:2301.13411*, 2023.
- [25] Y. Zhang, Y. Yuan, Y. Feng, and X. Lu, "Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection," *IEEE Transactions on Geoscience and Remote Sensing*, 2019.
- [26] K. Li, G. Cheng, S. Bu, and X. You, "Rotation-insensitive and context-augmented object detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2017.
- [27] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-rpn and multi-relation detector," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.
- [28] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," 2020.
- [29] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "Fsce: Few-shot object detection via contrastive proposal encoding," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2021.
- [30] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang, "Defrcn: Decoupled faster r-cnn for few-shot object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.
- [31] K. Guirguis, A. Hendawy, G. Eskandar, M. Abdelsamad, M. Kayser, and J. Beyerer, "Cfa: Constraint-based finetuning approach for generalized few-shot object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [32] X. Li, J. Deng, and Y. Fang, "Few-shot object detection on remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [33] Z. Zhao, P. Tang, L. Zhao, and Z. Zhang, "Few-shot object detection of remote sensing images via two-stage fine-tuning," *IEEE Geoscience and Remote Sensing Letters*, 2022.
- [34] Y. Zhou, H. Hu, J. Zhao, H. Zhu, R. Yao, and W.-L. Du, "Few-shot object detection via context-aware aggregation for remote sensing images," *IEEE Geoscience and Remote Sensing Letters*, 2022.
- [35] Z. Xiao, J. Qi, W. Xue, and P. Zhong, "Few-shot object detection with self-adaptive attention network for remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2021.
- [36] Y. Zhang, B. Zhang, and B. Wang, "Few-shot object detection with self-adaptive global similarity and two-way foreground stimulator in remote sensing images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022.
- [37] G. Cheng, B. Yan, P. Shi, K. Li, X. Yao, L. Guo, and J. Han, "Prototype-cnn for few-shot object detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2022.
- [38] Z. Zhang, J. Hao, C. Pan, and G. Ji, "Oriented feature augmentation," 2021.

- [39] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” *Advances in neural information processing systems*, 2017.
- [40] A. Li and Z. Li, “Transformation invariant few-shot object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.
- [41] S. Laine and T. Aila, “Temporal ensembling for semi-supervised learning,” in *International Conference on Learning Representations*, 2017.
- [42] X. Feng, X. Yao, G. Cheng, and J. Han, “Weakly supervised rotation-invariant aerial object detection network,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
- [43] X. Feng, X. Yao, G. Cheng, J. Han, and J. Han, “Saenet: Self-supervised adversarial and equivariant network for weakly supervised object detection in remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, 2021.
- [44] J. Han, J. Ding, N. Xue, and G.-S. Xia, “Redet: A rotation-equivariant detector for aerial object detection,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2786–2795.
- [45] G. Cheng, P. Zhou, and J. Han, “Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 12, pp. 7405–7415, 2016.
- [46] X. Xu, M. C. Nguyen, Y. Yazici, K. Lu, H. Min, and C.-S. Foo, “Semicurv: Semi-supervised curvilinear structure segmentation,” *IEEE Transactions on Image Processing*, 2022.
- [47] X. Xu and G. H. Lee, “Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020.
- [48] D. Marcos, M. Volpi, N. Komodakis, and D. Tuia, “Rotation equivariant vector field networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5048–5057.
- [49] L. Karlinsky, J. Shtok, S. Harary, E. Schwartz, A. Aides, R. Feris, R. Giryes, and A. M. Bronstein, “Repmet: Representative-based metric learning for classification and few-shot object detection,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
- [50] T. Zhang, X. Zhang, P. Zhu, X. Jia, X. Tang, and L. Jiao, “Generalized few-shot object detection in remote sensing images,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 195, pp. 353–364, 2023.
- [51] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [52] M. Everingham, L. V. Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, 2010.
- [53] mmfewshot Contributors, “Openmmlab few shot learning toolbox and benchmark,” <https://github.com/open-mmlab/mmfewshot>, 2021.
- [54] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
- [55] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.