Applied Intelligence (2023) 53:351–368
https://doi.org/10.1007/s10489-022-03399-2

Object detection based on few-shot learning via instance-level feature correlation and aggregation

Meng Wang[1,2] · **Hongwei Ning**[1,2] · **Haipeng Liu**[1,2]

Accepted: 15 February 2022 / Published online: 18 April 2022

Abstract

The detection of novel foregrounds only utilizing scarce annotated images, namely few-shot object detection, makes a detector no longer dependent on large-scale instantiated sets. The realistic challenge might lie in establishing the correlation of few instances and balancing the sensitivity between base and novel categories. In this paper, we propose a few-shot detector using instance-level feature correlation based on an interactive self-attention module to deeply mine the discriminating representations from scarce novel instances. Besides, using an extended soft threshold shrinkage, a feature aggregation procedure is introduced to eliminate redundant information while enhancing the representation sensitivity between base and novel categories. In the training phase, an orthogonal loss is applied to further enhance the feature distinguishability of inter-categories. Finally, we evaluate related competitive detectors on both benchmarks PASCAL-VOC07/12 and MS-COCO, with the results verifying the superior detection precision on AP, mAP and AR measurements of the proposed approach.

Keywords Few-shot learning · Object detection · Self-attention · Feature aggregation · Orthogonal loss

1 Introduction

To detect both location and classification of interesting objects in vision scenarios has been excellently developed utilizing deep neural architectures [25, 30, 32]. Some of the general detectors have even achieved the same ability as human eyes [28]; however, a prerequisite is that their network weights should be fully trained on a large-scale set with annotated instances [13]. In reality, taking for instance, the detection of rare animals and plants or the lesion detection, there are not so many detailed annotated examples to satisfy the model training; In addition, the manual annotations are often costly [18, 45]. Therefore, few-shot learning [42, 43] has been applied to detection scenarios in order to eliminate this bottleneck and incrementally learn novel foregrounds with a scarce corresponding resource [39]. At present, the realistic challenge might lie in establishing the representation correlation among these scarce instances while balancing their discriminating sensitivity between base and novel categories. Focusing on these capabilities, this paper will deeply mine the salient instance-level representations in order to formulate an extended detection architecture based on few-shot learning scenarios.

The few-shot learning aims to utilize prior knowledge to quickly generalize the base model to novel tasks, in terms of only a few labeled training data [41]. In general, the methods of few-shot learning contain three stages. The first-stage is to learn prior knowledge from a set with abundant samples and detailed annotations [26]. In the second-stage, the samples with visible and invisible categories are formulated as a support set and a query set respectively, and different strategies are applied to learn the category representations from typical K samples of each category [33]. In the third-stage, the samples involving both visible and invisible classes are fed into the previously trained model

✉ Meng Wang
wangmeng@kust.edu.cn

Hongwei Ning
Rui.Ning0807@gmail.com

Haipeng Liu
42227324@qq.com

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

[2] Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming 650500, China

to obtain final predictions. According to vision scenes, few-shot learning can be implemented [1, 4, 8] mainly via meta-learning [2, 6], metric-learning [3, 7], and etc. Recently, studies on few-shot object detection have emerged using different solutions based on the above mentioned approaches. The few-shot object detection based on meta-learning [12, 15] is devoted to designing effective network structures to evaluate and align the categorical feature between the support set and query set, and then iteratively update the network weights with few data batches. Moreover, the models based on metric-learning [11, 14] attempt to extract the embedding representation of features, and then to calculate the distance between these embedded feature vectors to determine which category the object belongs to. This type of solution might be affected by the metric function and the coupling features generated by the network. In addition, the detector based on transfer learning [10, 19] is dedicated to aggregating the context information of the source domain into the target domain. These transfer methods often have a strong dependence on source domain samples. Most of the above approaches directly adopt fragmented features for learning, which brings more background clutters and confusion features, and also has a great impact on the performance of few-shot object detection.

In recent years, a few works [45] have implied that though the localization of few-shot object detection can be greatly improved by pretraining on the base-category set and then fine-tuning on the novel-category set [19], nevertheless this modification is more challenge for instance categorization. The further solution might consist in how to obtain more discriminating instance-level features. The efficient learning of these instance-level features can prevent a lot of confusing backgrounds or prior-boxes with a small object proportion from entering the final classifications, thereby affecting the discriminating performance [15]. In general, the instance-level features are generated by the prior object box through non-maximum suppression [25, 30], and it is inevitable that there are many fragmented candidate boxes that contain multiple objects or merge backgrounds into the classification layer. It will lead to misclassification due to the lack of key category information that prevents the classifier from acquiring global recognition of images [19]. Therefore, when the visual information of the novel category samples is difficult to obtain, it is necessary to intensively formulate the inherent representations of instance-level features. Secondly, in the current approaches, the fusion or matching strategy of the support-set and the query-set features does not seem to be applicable [19, 50]. Some methods suffer the imbalance between the base categories and the novel categories in the final weight updating [17, 50]. Either too much attention is paid to the detection performance of base categories with the relatively poor detection performance for novel categories [15, 17], or only the performance improvement of novel categories is considered while the stability of base categories is ignored [12, 21] (Fig. 1).

This paper will chiefly intend to utilize the feature correlation and aggregation on instance-level to avoid the negative effects of excessive feature clutters and discriminating confusions brought by scarce novel instances. Inspired by DANet [48] and Nonlocal [46], a feature correlation will be discussed from the high-level semantics and fine-grained aspects of instance features using an interactive self-attention module, so that the network can effectively mine discriminating information from scarce training samples. We will further introduce an adaptive feature aggregation module, which aggregates the inferred support-set features into the query-set features to highlight the aligned category representations contained in these features. Furthermore, an improved soft-threshold shrinkage based on the adaptive function adopted in [23] will be integrated to filter out redundant information from these aggregated features, thus balancing the feature sensitivity and stability for both the base and the novel categories. In order to alleviate the final discriminating confusions due to ineffective few-shot learning, as a supplement to the above modules, an orthogonal category loss will be applied to constrain the classifier optimization as recent suggested in [35] which can make the features of the same category are aggregated more closely when the features of different categories remain orthogonal. In experiments, we will perform evaluations of recent competitive few-shot detectors on the benchmarks PASCAL VOC 07 [24]/12 [20] and MS COCO [29], and the results will verify that the proposed architecture achieves better performance compared with the comparison detectors. The main contributions of this paper can be summarized as follows.

1) An extended detection architecture of few-shot scenarios is formulated based on instance-level feature correlation (IFC) using a novel interactive self-attention module, which aims to deeply mine the discriminating features from scarce novel instances.
2) Besides, a feature aggregation mechanism is introduced into the backbone network through an improved adaptive soft-threshold shrinkage (ASA) to eliminate redundant information among these features, while enhancing the feature sensitivity and stability for both the novel and the base categories.
3) In addition, an orthogonal loss is suggested based on the similarity measurement between the instance features in training batches, so that these instantiated representations can be constrained to further enhance the final foreground distinguishability.

The rest of this paper are arranged as follows. Section 2 reviews the related works including general object detection,

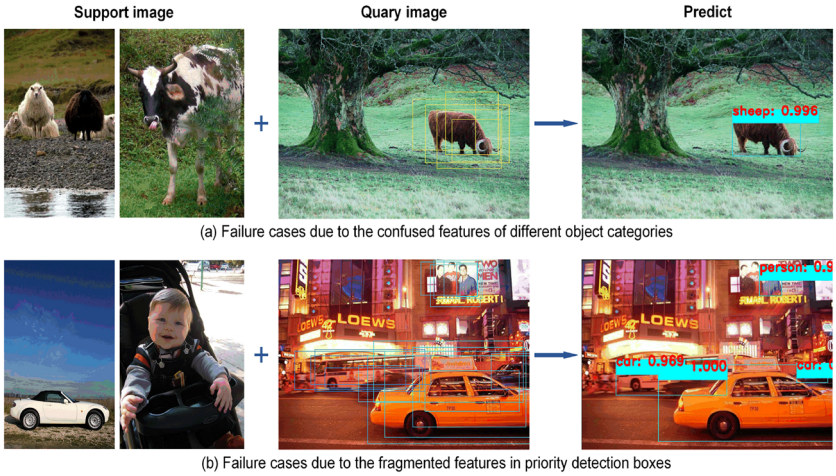


Fig. 1 The failure cases due to the above mentioned problems that often bring great challenges to few-shot object detection

few-shot learning and few-shot object detection. The proposed architecture of few-shot object detection is detailed in Section 3. The next section presents experimental settings, as well as the comparison evaluations and the discussions. The last section summarizes this paper.

2 Related work

2.1 General object detection

The deep network architecture of object detection can be divided into single-stage and two-stage categories according to the operation process. For instance, SSD [30] and YOLO [31, 32, 34] are single-stage detectors, which directly predict the classification confidence of object categories and the coordinates of regression boxes on the dense grid. The approaches to two-stage detection represented by Faster R-CNN [25] and its variants roughly extract the object location at first, and then obtain refined classification and location results through one or more refinements. Although the single-stage object detection network has a lightweight architecture and fast running speed, its detection accuracy is indeed far behind that of the two-stage object detector [22]. The above methods are based on object proposals, and the detection results are obtained by further optimizing the predefined rough object bounding box. Recently, proposal-free detectors have gradually become a new trend and made some progress, such as CornerNet [38], ExtremeNet [47] and Centernet [37] and their improved methods, which accomplish locations and classifications by predicting the key points of image foregrounds [22]. Although this kind of solution avoids extracting proposals and setting sensitive parameters, the use of key points for object detection itself requires more detailed location information and even more advanced instance-level semantic masks, thus setting strict requirements on the design of the network and making it not easy to achieve [27]. Taken together, the basic architecture of two-stage detectors is more suitable for improving few-shot object detection.

2.2 Few-shot learning

In recent years, few-shot learning has been applied in many vision fields based on two basic mechanisms, namely meta-learning and metric-learning. The approaches based on meta-learning [26] adopt a general strategy to capture the structural changes of different tasks through accumulating knowledge, to learn the distribution characteristics of samples through one or several gradient descent steps and to quickly generalize the model to novel categories [33]. Specifically speaking, the representative MAML [2] uses meta-knowledge to obtain a reliable initialization parameter, and quickly adapts to new tasks through gradient adjustment. MetaOptNet [4] analyzes the implicit differentiation of the optimal conditions of the convex problem of linear classifiers, enabling networks to embed high-dimensional features for rapid generalization. On the other hand, the key to metric-learning [26] is to construct a suitable embedding space, and then to distinguish different classes by designing clever loss functions and metric functions. PrototypeNet [3] introduces prototype clustering to construct the embedding space, and uses Euclidean distance to measure the class similarity of the embedding vector. MatchingNet [1] tries to combine feature extraction with differentiable K-NN, and uses cosine similarity to calculate the feature matching. DFL [9] proposes a few-shot classification weight generator based on the attention mechanism and cosine similarity discriminator to solve the problem of catastrophic forgetting. These above models have made great progress in few-shot classifications; however, the scene of object detection

involved both classification and localization, which is more complicated and challenging. Therefore, the above solutions cannot be directly applied to the few-shot object detection.

2.3 Few-shot object detection

As mentioned above, the detectors applied in few-shot scenes should be extended to efficiently capture more complex information for novel categories and instances. According to metric learning, RepMet [11] refers to the practice in PrototypeNet [3] to calculate the distance between the embedded feature and the support set category representation by designing the prototype space. TFA [17] refers to the practice in DFL [9] and replaces the discriminator in the second training phase with a cosine classifier. However, most of the existing metrics cannot resolve the negative impact caused by the interaction between high infra-category variance and the coupling features generated by the network [14]. As for the detectors using meta-learning, FSRW [12] applies a meta-weighted model to fusing the support features into the query features by channel multiplication. Meta R-CNN [15] integrates a predictive remodeling network of meta-weight into the instance-level ROI of the query set. A weight prediction meta-model is suggested in MetaDet [40], which unifies the classification and localization problems in few-shot object detection. Nevertheless, most of these methods require additional branches and directly use fragmented features for detection, which undoubtedly increases the parameters of the model. Inspired by transfer learning, Context-Transformer [19] proposes to guide the model to mine the context information of the target domain image by aggregating the features of the source domain image to avoid object confusion. MPSR [18] designs a positive example refinement branch to refine multi-scale features. In general, there are many factors affecting the performance of this type of method, which is also not easy to implement. Using dynamic graphs driven by image data, SRR-FSD [54] is proposed to construct the relationship between visual information and text information. DCNet [52] is put forward to utilize an improved self-attention module for relational distillation to mine fine-grained features. Using self-attention and multi-scale positive sample augmentation, FSSP [53] is formulated to solve the overfitting caused by hard samples and insufficient sample numbers. DAnA [49] alleviates the problem of spatial misalignment of features and entanglement of aggregated information by generating paired robust features that are sensitive to spatial location. FSAP [55] aggregates the query and support data based on meta-learning, and introduces attention mechanisms on support branches to refine few-shot information. DRL [56] is suggested via a dynamic relevance learning model using the relationship between the support images and the interesting regions on the query images to construct a dynamic graph convolutional network to guide the detector output. Although the above methods improve the learning efficiency of the detection network for each sample, they fail to consider the negative effects of redundant features and confusion between instances, or effectively solve the performance gap between the novel and the base classes.

3 Detection architecture of few-shot learning

In this section, we focused on presenting the overall architecture and the learning strategy for few-shot object detectors. Moreover, the related few-shot modules were shown in Fig. 2, in the hope of improving the existing detection baselines in terms of meta learning [15].

3.1 Preliminary

According to general few-shot configurations, a full training dataset $\mathcal{D}_{\text{train}}$ contained C_{train} categories, which was divided into a base dataset $\mathcal{D}_b$ contained C_b categories and a novel dataset $\mathcal{D}_n$ contained C_n categories, where $C_{\text{train}} = C_b \cup C_n$. Besides, there was no shared category between $\mathcal{D}_b$ and $\mathcal{D}_n$, namely $C_b \cap C_n = \emptyset$.

As previously mentioned, this paper concentrated on the training strategy of a two-phase detector as in FSRW [12]. Following the setting of meta-learning, we further formulate these datasets into tasks $T_j = S_j \cup Q_j = \{(I_1^S, Y_1^S), ..., (I_L^S, Y_L^S)\} \cup \{(I_j^Q, Y_j^Q)\}$, each task contained a support set S_j and a query set Q_j, where I denotes images with their associated annotations Y. In the first stage, the task T_j was randomly divided on the dataset $\mathcal{D}_b$ contained only the base categories C_b, and the detector was wholly trained to learn the prior knowledge on $\mathcal{D}_b$. In the second stage, the task T_j is randomly divided on the $\mathcal{D}_{\text{train}}$ that contained both the categories C_b and C_n, then the weight of the previous model was fine-tuned, and each category typically contained K labeled instances. Then the detection performance was evaluated on the test set $\mathcal{D}_{\text{test}}$, which also contained test samples with both the categories C_b and C_n.

3.2 Proposed modules for few-shot detection

3.2.1 Instance-feature correlation based on interactive self-attention

The two-stage detection baseline as first applied in Faster R-CNN [25], adopted the region proposal network (RPN) to generate various candidates of object locations in the first stage. Afterwards, the object proposals were aligned through the ROIAlign layer. The detector extracted instance features within these candidate regions corresponding to the boundary boxes, and then performed classification and

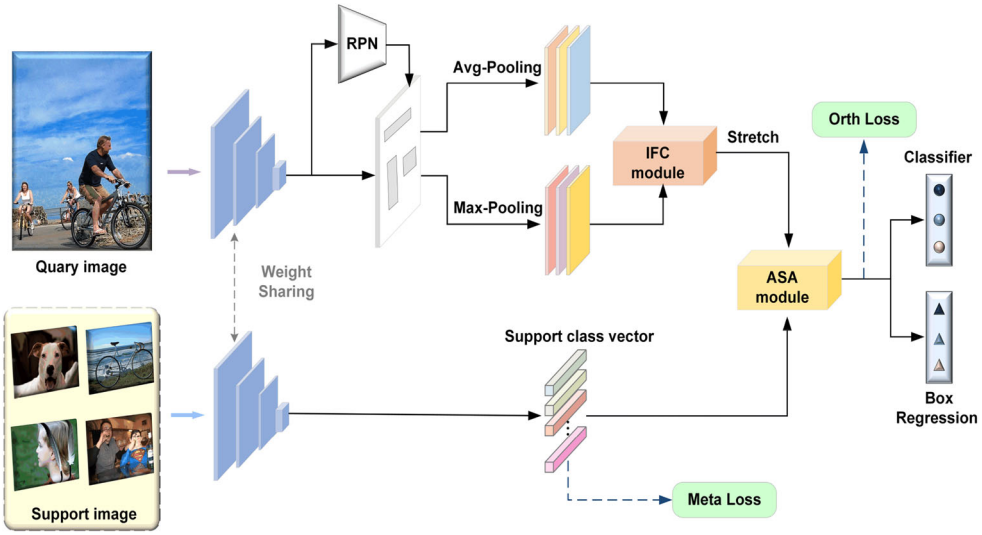


Fig. 2 The overall framework of our proposed module. It consists of two branches. The query branch receives query images to generate instance-level features. The support branch is used to generate support class vectors. We further introduced the Instance-level feature correlation module (IFC), which is used to construct the correlation of instance features. Then introduced an adaptive shrinking aggregation module (ASA) to enhance the parameters sensitivity of the features of the query set and support set, and use orthogonal loss to avoid inter-class and intra-class confusion

location by the classification layer and the box regression layer respectively.

In general, for the query branch these instance-level features could represent the regional semantics of candidate objects in a concentrated manner. Thus, they barred excessive messy backgrounds from negatively affecting the learning process. Typically, the high-level semantic features generated by avg-pooling and the detailed texture features generated by max-pooling were of great use for object detection or classification. Therefore, the extended interactive operations using two independent forward-paths to complement and enhance the two types of features would be highly helpful for the scarce novel instances under few-shot scenes. It could be seen from Fig. 3 that we employed an interactive self-attention structure in the proposed IFC module to achieve the instance-level feature correlation. Driven by both the local semantic feature $Z^A \in \mathbb{R}^{N\times C\times H\times W}$ based on avg-pooling operation, and the fine-grained representation of the texture feature $Z^R \in \mathbb{R}^{N\times C\times H\times W}$ based on max-pooling operation, this module structure could be regarded as a supplementary enhancement setting for few-shot instance learning, which improves the learning efficiency of the network. Next, a 3×3 convolution layer was applied to increasing the receptive field to obtain both features $\tilde{Z}^A$ and $\tilde{Z}^R$, which are then reshaped into $\hat{Z}^A \in \mathbb{R}^{N\times C\times V}$ and $\hat{Z}^R \in \mathbb{R}^{N\times C\times V}$ as a vectorizing operation, where $V = H \times W$, and thus $\hat{Z}^A$ can be operated on the channel dimension to formulate a correlation matrix $P \in \mathbb{R}^{N\times C\times C}$ as

$$P_{j,i} = \frac{\exp(\hat{Z}_j^A \times \hat{Z}_i^A)}{\sum_{i=1}^{C}\exp(\hat{Z}_j^A \times \hat{Z}_i^A)} \tag{1}$$

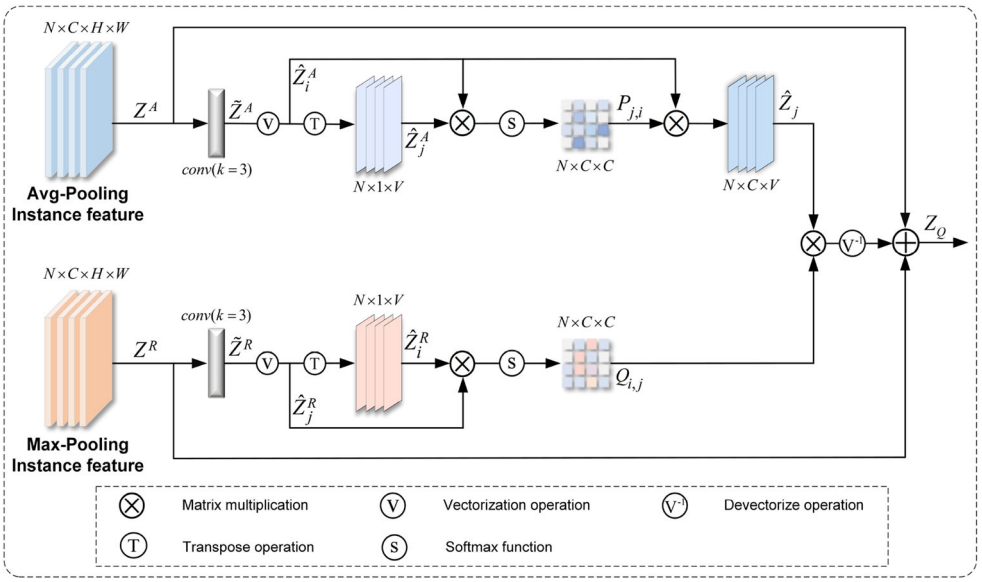


Fig. 3 Illustration of the instance-level feature correlation (IFC) module based on interactive self-attention. This module uses self-attention based on avg-pooling and max-pooling to separately mine local semantic information and detailed texture information in features to complement each other to achieve relational distillation

where the elements $P_{j,i}$ represented the relation metric of the instance feature $\hat{Z}^A$ from the $\hat{Z}_i^A \in \mathbb{R}^{N\times 1\times V}$ on i^{th} channel to the $\hat{Z}_j^A \in \mathbb{R}^{N\times 1\times V}$ on j^{th} channel, and each row of this matrix denoted the semantic correlation degree of the instance features. Then, the correlation matrix P was mapped to weight the feature $\hat{Z}_i^A$ and achieve the feature correlation on each channel.

$$\hat{Z}_j = \sum_{i=1}^{C} P_{j,i} \times \hat{Z}_i^A \tag{2}$$

To further secure the detailed features, the ROI feature Z was simultaneously fed to the max-pooling layer to obtain the feature Z^R. Then, the 3×3 convolution and vectorizing operation were also applied to obtain the feature $\hat{Z}^R$. To acquire fine-grained correlation, $\hat{Z}_j^R$ and its transposed $\hat{Z}_i^R$ were employed to perform matrix multiplication and get a finer attention matrix $Q \in \mathbb{R}^{N\times C\times C}$, and it was normalized through the following operations to obtain fine-grained weights $Q_{i,j}$.

$$Q_{i,j} = \frac{\exp(\hat{Z}_i^R \times \hat{Z}_j^R)}{\sum_{j=1}^{C}\exp(\hat{Z}_i^R \times \hat{Z}_j^R)} \tag{3}$$

To supplement this fine texture representation with the former semantic correlation, we further mapped the weights $Q_{i,j}$ onto the former correlation feature $\hat{Z}_j$ to highlight the fine discriminating features, and a scale coefficient $\beta \in [0, 1]$ was used to affect the mapping results as follows.

$$\hat{G}_i = \beta \sum_{j=1}^{C} Q_{i,j} \times \hat{Z}_j \tag{4}$$

Finally, the dimension of feature matrix $G = [\hat{G}_1, \hat{G}_2, ... \hat{G}_C]$ was recovered as $G \in \mathbb{R}^{N\times C\times H\times W}$ with the same size of Z^A and Z^R, then an element-wise sum operation was applied to the extended features G, Z^A and Z^R to obtain the final output of this proposed IFC module with the network parameters θ_Q, which is initialized by a random normal distribution and updated with all query samples. The final feature was formulated as below.

$$Z_Q = Z^A + Z^R + G \tag{5}$$

In this section, we integrated the interactive self-attention to obtain the instance feature correlation based on both the local high-level semantics and the fine texture information. In the following sections, we would further use the output Z_Q of this module as the input of the ASA module to present our approach.

3.2.2 Feature aggregation using adaptive soft-threshold shrinkage

As previously presented, the unsatisfactory detection performance of the base category and the novel category samples and the serious imbalance could be attributed to the fact that the model was not sensitive enough to the class parameters. Moreover, the excessive redundant information and insignificant features in the network were the main reasons for this problem. In this section, we attempted to solve this problem and propose the ASA module, which employed the feature multiplication between the query set instance vector and the support set class vector to aggregate the features of the same classes, and adopted the feature subtraction to highlight the features of different classes. Furthermore, in order to filter out the insignificant features in the aggregated features, we integrated a soft-threshold shrinking operation as in DRSN [44] proposed for the signal processing, and improved and formulated it as an adaptive shrinkage function performed in the feature aggregation module.

The block diagram of the proposed module ASA was shown in Fig. 4. Specifically, we standardize the space size of the support images into 224×224, and combine the support images and their associated annotations $S = \{(I_l^S, Y_l^S), l = 1, 2, ...L\}$ into a vector with 4-channel and fed into the feature extractor, which shares the entire backbone with the query branch except the first layer, then generate the support-category vector $V_{\text{cls}} \in \mathbb{R}^{M\times U}$ after the pooling and activation operations, where M represented the number of categories contained in the support-set images with the vector dimension $U = C \times H \times W$, and each category vector denoted as $V_{\text{cls},i} \in \mathbb{R}^{1\times U}, i = 1, 2, ...M$. This meta-learning branch should be constrained by loss $\mathcal{L}_{\text{meta}}$ to maintain the category spacing among the category vectors of support sets. This loss function was expressed as follows.

$$\mathcal{L}_{\text{meta}} = L_{\text{CE}}(\, W_m\, (\, E_b(\, I_l^S\,)\,),\; Y_l^S\,) \tag{6}$$

where $L_{\text{CE}}(\cdot)$ denoted the cross-entropy loss, E_b and W_m represented a feature extractor and a linear classifier, respectively. In addition, Z_Q represented the instance-level features of each query-set image generated by the former feature correlation module. It was stretched into a 1-dimensional vector denoted as $\hat{Z}_Q \in \mathbb{R}^{N\times U}$, where N was the number of instance features contained in each image. We first performed a broadcasting convolution to weight the support-set feature V_{cls} onto $\hat{Z}_Q$ and obtained $F_Z \in \mathbb{R}^{N\times U}$ to enhance the channel weights of the contained categories.

$$F_Z = \sum_{i=1}^{M} \hat{Z}_Q \otimes V_{cls,i} \tag{7}$$

Then, an element-level subtraction was performed to highlight different category information contained in the query-set features and the support-set features, and $F_D \in \mathbb{R}^{N\times U}$ was obtained as follows.

$$F_D = \sum_{i=1}^{M} \hat{Z}_Q - V_{cls,i} \tag{8}$$

The proposed adaptive feature shrinking module was indicated as $S(\cdot)$. As shown in the lower part of Fig. 4, the different convolution layers were implemented to formulate a learnable forward-path, and a scale parameter was obtained through averaging operation, then the absolute value was taken for these features.

$$x = \left| \frac{1}{N} \sum_{i=1}^{N} h_{\mathrm{ReLu}} \circ h_{\mathrm{BN}} \circ h_{\mathrm{conv}}(F_i) \right| \tag{9}$$

where the single instance feature $F_i \in \mathbb{R}^{1\times U}, i = 1, 2, ...N$ and $\circ$ denotes the operator of function composition between the adjacent layer functions. Afterwards, a 1-dimensional convolution layer was performed to achieve the interaction of local features on the channel, and then the sigmoid function was applied to map these scale metrics and obtain the normalized vector s.

$$s = \frac{1}{1 + \exp^{-h_\rho(x)}} \tag{10}$$

where $h_\rho(\cdot)$ represented the 1-dimensional convolution layer. Then, the score vector s and the absolute value vector x were multiplied to obtain the adaptive threshold value $\tau = s \cdot x$. Moreover, the threshold τ and the feature vector $F_\varphi = h_{\mathrm{ReLu}} \circ h_{\mathrm{BN}} \circ h_{\mathrm{conv}}(F_i)$ were both fed into the operator D.

Although this form of soft-threshold operator could dynamically filter out the fragmented representations, it would also make feature details smooth and degraded, thus leaving the reformulated feature deviating from the original feature. In order to relieve this difficulty, we revised the soft-threshold operator into the following formula.

$$d(F_\varphi) = \begin{cases} \mathrm{sign}(F_\varphi) max\left\{ \left|F_\varphi\right| - \tau e^{-\tau},\ 0 \right\} & , \quad \left|F_\varphi\right| > \tau \\ 0 & , \quad \left|F_\varphi\right| < \tau \end{cases} \tag{11}$$

As can be inferred from the characteristic curves in the lower-right corner of Fig. 4, the threshold τ in this function will dynamically adapt to the information learned from the aggregated features, thus filtering out clutter and insignificant features. This nonlinear characteristics of the function enables the reconstructed features to asymptote the original features without being too smooth and degraded. To maintain the stability of the model, we further adopted the residual structure, and the output of $S(\cdot)$ was expressed as the following equation.

$$\hat{F}_\varphi = d(F_\varphi) + F_\varphi \tag{12}$$

where F_Z and F_D were fed into $S(\cdot)$ to eliminate redundant information from features and obtain the shrinking features $\hat{F}_Z = S(F_Z)$ and $\hat{F}_D = S(F_D)$.

Next, the $\hat{F}_Z$ and $\hat{F}_D$ were aggregated through the concatenate operation with trainable network parameters θ_A which is initialized by a random normal distribution and updated via the ASA module. Also $\hat{Z}_Q$ was concatenated in this module, since $\hat{Z}_Q$ was important for stabilizing the detection performance of the base categories.

$$F_{\mathrm{con}} = \mathrm{Concat}\left(\hat{F}_Z,\ \hat{F}_D,\ \hat{Z}_Q\right) \tag{13}$$

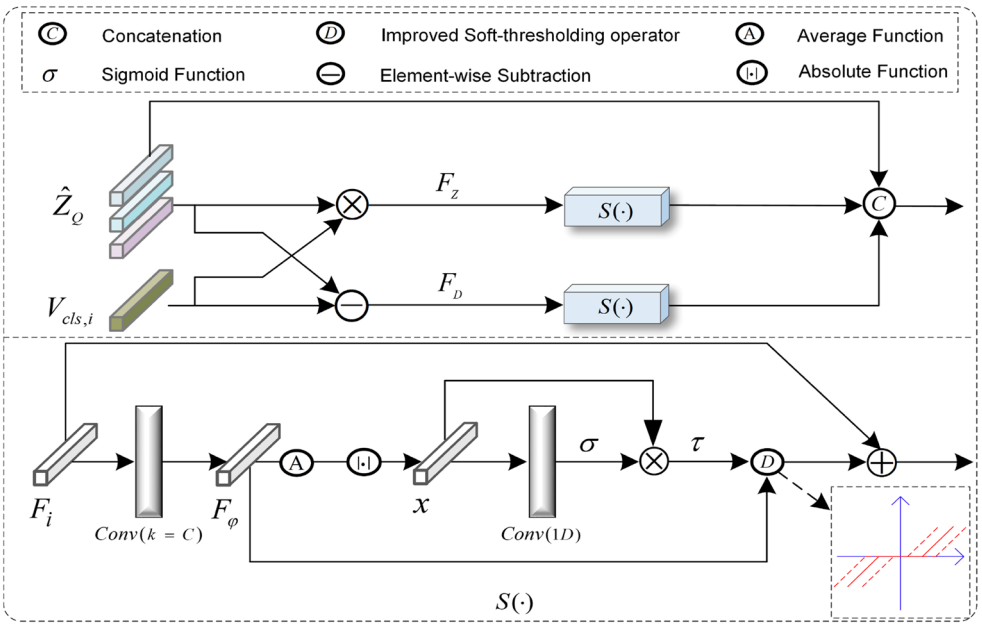


Fig. 4 Overall architecture of module ASA, where C was the number of feature channels, k was the size of the convolution kernel, and τ represented the threshold. The lower right corner was the characteristic curve of the improved soft-threshold function

Finally, these concatenated features were fed into the final prediction layers $\mathcal{P}(F_{\text{con}})$ with the independent parameters to obtain the outputs.

The above module used the element-wise multiplication and subtraction to obtain discriminating instance features, and the improved soft-threshold function was adopted to shrink the redundant features. Therefore, this proposed subnetwork could be efficiently integrated into the detection network for weight generalization.

3.2.3 Orthogonal category loss with cosine similarity

In the few-shot detection scenes, it was difficult for the basic detector to learn reliable feature representation only through a few novel instances, and the lack of accuracy location parameters caused great challenge to the classifier. In addition, there would inevitably be the situation of feature superposition and interaction during the previous calculation of the instance-feature correlation. In this case, if the hidden layer features were directly input to the classification layer, this would be affected by intra-category and inter-category confusion to a certain extent. To alleviate this negative influence, we adopt an orthogonal category loss [35] that calculates the cosine similarity of features as a constraint to impose orthogonality in the hidden layers in order to maintain the instance separation of different categories and the instance aggregation of the same categories.

For the sake of simplicity, the proposed architecture was summarized as a hidden feature block $\mathcal{F}(\cdot)$ and an output prediction block $\mathcal{P}(\cdot)$, and the features output by the hidden block were taken as the input of the prediction block. In a task of input $T_j = \{(I_1^S, Y_1^S), ..., (I_{L-1}^S, Y_{L-1}^S)\} \cup \{(I_j^Q, Y_j^Q)\}$, where I_l indicated the l^{th} input image, and Y_l denoted the corresponding labels, then the output of T_j through the hidden feature block was represented as $\mathcal{F}(I_l) = \left[f_{l,1}, f_{l,2}, \cdots, f_{l,N}\right]$, where $f_{l,i}$ denoted the i^{th} instance feature in the l^{th} batched image, and the category prediction of the k^{th} $(k = (l-1) \times N + i)$ instance feature was $\hat{c}_k = \mathcal{P}_c(f_k)$.

Assuming that the two different instance features f_m and f_n belonged to the same category, namely the category labels $c_m = c_n$, then we had

$$p = \sum \langle f_m, f_n \rangle, \qquad m, n \in \{1, 2, ..., N \times L\}, m \neq n \quad (14)$$

Considering these two instance features belonged to different categories, namely the category labels $c_m \neq c_n$, the function could be written as

$$q = \sum \langle f_m, f_n \rangle, \qquad m, n \in \{1, 2, ..., N \times L\} \quad (15)$$

where $\langle \cdot, \cdot \rangle$ was to calculate the cosine similarity of the two feature vectors illuminated as follows.

$$\langle f_m, f_n \rangle = \frac{f_m \times f_n^T}{\| f_m \|_2 \cdot \| f_n \|_2} \quad (16)$$

where $\| \cdot \|_2$ represented the $L2 - norm$ of the input vector. Thus, the orthogonal category loss was defined as the equation bellow.

$$\arg\min_{\theta_A} \mathcal{L}_{\text{orth}} = (1 - p) + \alpha \cdot |q| \quad (17)$$

where $\alpha \in [0, 2]$ was a weighting coefficient, and $|\cdot|$ denoted the absolute value. In (17), we constrained p to be close to 1, and q to be close to 0 to maintain the clustering of features of the same class, and keep the distance between features of different categories to reduce the impact of representation confusion on discrimination.

The training loss of the basic framework [25] adopted in our approach could be expressed as $\mathcal{L}_{\text{base}}$.

$$\arg\min_{\theta_B} \mathcal{L}_{\text{base}} = L\left(\mathcal{P}(\mathcal{F}_B(I^Q), Y^Q\right) \quad (18)$$

where $\mathcal{F}_B(\cdot)$ represented the hidden layers of the basic framework and $L(\cdot)$ denoted the predicted loss, defined as $\mathcal{L}_{\text{base}} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}$, and $\mathcal{L}_{\text{rpn}}$ was the loss used to distinguish foregrounds and backgrounds, as well as fine-tune anchors in the RPN. In addition, $\mathcal{L}_{\text{cls}}$ was the cross-entropy loss applied for the instance classification, and $\mathcal{L}_{\text{loc}}$ was the $Smooth - L1$ loss implemented for the bounding box regression in R-CNN.

According to the above, the complete training loss of the detector in this paper could be expressed as follows,

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{meta}} + \mathcal{L}_{\text{orth}} \quad (19)$$

where the scale coefficient $\lambda \in [0, 1]$ and these extended terms could be regarded as a supplement to the base loss, by introducing the inherent angular measurement among these scarce novel instances. In addition, through formulating the orthogonal category loss, a relatively stable geometric structure could be updated and then maintained in the feature space, which also reduced the sensitivity of the model to training configurations such as the data-batch size. To summarize this section, the training procedure of our detector is performed as following Algorithm 1.

Algorithm 1 Training few-shot detector with the proposed modules.

Input:
Support set: $\mathcal{S} = \{I^S,\ Y^S\}$;
Query set: $\mathcal{Q} = \{I^Q,\ Y^Q\}$.
Output:
Network parameters $\theta = \theta_A \cup \theta_B \cup \theta_Q$.
Training first-phase:
for $\left(I^S,\ Y^S,\ I^Q,\ Y^Q\ \ \textbf{in}\ \ \mathcal{D}_b\right)$ **do**
Predict:
(1) Obtain query instances Z via feeding $\{I^Q,\ Y^Q\}$ to RPN;
(2) Obtain Z^A and Z^R via feeding Z to ROIAlign layer;
(3) Obtain support vectors V_{cls} via feeding $\{I^S,\ Y^S\}$ to extractor;
(4) Obtain Z_Q via feeding Z^A and Z^R to module IFC, according to (1)˜(5);
(5) Obtain F_{con} via feeding stretched $\hat{Z}_Q$ and V_{cls} to module ASA, according to (7)˜(13);
(6) Predict $\mathcal{P}(F_{\text{con}})$ via feeding F_{con} to final layer;
Calculate base loss $\mathcal{L}_{\text{base}}$ by (18); meta loss $\mathcal{L}_{\text{meta}}$ by (6); orthogonal loss $\mathcal{L}_{\text{orth}}$ according to (14)˜(17);
Update θ to minimize the loss $\mathcal{L}$ by (19) with back-propagation;
end for
Training second-phase:
for $\left(\textbf{each}\ I^S,\ Y^S,\ I^Q,\ Y^Q\ \ \textbf{in}\ \ \mathcal{D}_b \cup \mathcal{D}_n\right)$ **do**
for $iter = 1, \ldots\ldots, j$ **do**
Predict:
(1) Obtain query instances Z via feeding $\{I^Q,\ Y^Q\}$ to RPN;
(2) Obtain Z^A and Z^R via feeding Z to ROIAlign layer;
(3) Obtain support vectors V_{cls} via feeding $\{I^S,\ Y^S\}$ to extractor;
(4) Obtain Z_Q via feeding Z^A and Z^R to module IFC, according to (1)˜(5);
(5) Obtain F_{con} via feeding stretched $\hat{Z}_Q$ and V_{cls} to module ASA, according to (7)˜(13);
(6) Predict $\mathcal{P}(F_{\text{con}})$ via feeding F_{con} to final layer;
Calculate base loss $\mathcal{L}_{\text{base}}$ by (18); meta loss $\mathcal{L}_{\text{meta}}$ by (6); orthogonal loss $\mathcal{L}_{\text{orth}}$ according to (14)˜(17);
Update θ to minimize the loss $\mathcal{L}$ by (19) with back-propagation;
end for
end for

4 Results and analysis

To evaluate the detector performance in different few-shot scenes, we performed training on benchmarks PASCAL VOC07 [24]/12 [20] and MS COCO [29], and compared the proposed detector with various latest baselines to verify its effectiveness. Aimed at making a fair comparison, the experimental settings were consistent as applied in FSRW [12], and all the experimental results were obtained through running 10 times of random data sampling to calculate the average values.

In detail, we utilized Meta R-CNN [15] as the basic architecture. To build transferable scenes for detection, the ResNet-101 [28] pre-trained on ImageNet [36] was used as the default backbone network when trained on PASCAL VOC, while the ResNet-50 [28] pre-trained on ImageNet was adopted as the default backbone network when trained on MS COCO. The training strategy was the same as FSRW, a stochastic gradient descent (SGD) was used as the network optimizer, and the whole process was divided into two phases. In the first phase, we only trained the base dataset $\mathcal{D}_b$, the initial learning rate was set to 10^{-3} and the batch size was set to 4. 20 epochs were trained in total, with each epoch iterating 3000 times, and the learning rate decayed by 10^{-4} after every five epochs. The second phase was to train on the support set based on both $\mathcal{D}_b$ and $\mathcal{D}_n$. The initial learning rate was set to 10^{-3}, the batch size was set to 4, a total of 9 training epochs were performed, the learning rate decayed to 10^{-4} after training 5 epochs, and the training continued for 4 epochs. Moreover, the parameters $\alpha = 0.5$ and $\lambda = 1$ were set in the loss $\mathcal{L}$ in Eq. (19). All the experiments were implemented on Pytorch using GTX TITAN X.

4.1 Results on PASCAL VOC

According to the experimental settings, we used the Trainval set in PASCAL VOC07/12 to train the model, and evaluated the model performance on the test set of VOC07. Consistent with the standard evaluation settings [12, 15, 51], we considered three different split settings of dataset on VOC07/12. Each setting contained 15 base categories and 5 novel categories after random selection. These split settings were listed as follows: $Split1 \rightarrow (bird, bus, cow, mbike, sofa/rest)$; $Split2 \rightarrow (aero, bottle, cow, horse, sofa/rest)$; $Split3 \rightarrow (boat, cat, mbike, sheep, sofa/rest)$. In the first phase, the model was trained on the base dataset containing 15 categories; In the second phase, we performed further training on the full dataset with 20 categories containing both the base categories and the novel categories. Each category had K annotated samples, where $K \in \{1, 2, 3, 5, 10\}$. We followed the PASCAL Challenge protocol that the correct prediction should have an IoU of more than 0.5 with Ground Truth, and took the mean Average Precision (mAP) as the evaluation indicator.

Table 1 showed the comparison results on PSCAL VOC. It could be seen from Table 1 that most results of the

Table 1 Few-shot object detection mAP on VOC2007 Test set in novel classes

	Novel-class split1					Novel-class split2					Novel-class split3				
Methods/shot	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FSRW [12]	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet [40]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
Meta R-CNN [15]	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
FSOD-KT [50]	27.8	41.4	46.2	55.2	56.8	19.8	27.9	38.7	38.9	41.5	29.5	30.6	38.6	43.8	45.7
TFA w/fc [17]	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/cos [17]	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
FSDet View [16]	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
MPSR [18]	**41.7**	-	51.4	55.2	**61.8**	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7
AFD-Net [21]	33.1	39.6	51.6	55.3	60.7	24.7	29.2	40.4	**44.6**	48.4	27.1	35.0	43.1	48.4	53.6
CME(F-RCNN) [51]	41.5	47.5	50.4	**58.2**	60.9	27.2	30.2	**41.4**	42.5	46.8	34.3	39.6	45.1	48.3	51.5
DCNet [52]	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
FSAP [55]	24.3	36.5	44.9	52.0	59.2	20.5	27.5	33.1	40.9	47.1	22.4	33.0	37.8	43.9	51.5
Ours	41.0	**47.9**	**52.7**	55.0	61.5	**28.6**	**34.2**	36.8	42.4	**49.4**	**37.3**	**43.5**	**45.7**	**50.7**	**53.9**

We evaluate the baselines under three different splits of novel classes

Bold entries indicate the best performance achieved in that evaluation setting

proposed detector were superior to the latest baselines. In addition, all the evaluations of our model achieved excellent performance in the 10-shot settings, especially in split3, and reached the competitive evaluations with the state-of-the-arts models in all settings. Moreover, the proposed detector offered satisfactory results in the extremely-low-shot setups ($K \in \{1, 2, 3\}$) of the three split subsets, which illustrated the robustness and strong learning ability of our model under the condition of scarce resources with high variance. On the 1-shot and 10-shot experiments of split1, our method also achieved 41.0%(mAP) and 61.5%(mAP) respectively, which was slightly lower than that in MPSR [18]. Nonetheless, a multi-scale network architecture should be used in MPSR, while our architecture was only based on a single scale, which was more compact than MPSR. In general, the proposed detector was more competitive.

In Table 2, we reported the detailed evaluations on each novel category using the three default VOC splits. Compared with the baselines, it was obvious that when $K \in \{1, 2, 3\}$, Meta R-CNN performed poorly on "sofa" and "bottle". However, in extremely-low-shot setups ($K \in \{1, 2, 3\}$), our model improved the detection performance of "Sofa" in VOC split1 by 21%(mAP), 28.9%(mAP) and 34.6%(mAP) respectively, and the performance of "sofa" in VOC split2 and split3 was much higher than other baselines. For the tests of "bottle", when $K \in \{1, 2, 3\}$, the performance of our model in VOC split2 improved by 7.1%(mAP), 8.0%(mAP) and 15.5%(mAP) respectively. On the whole, compared with other models, the detection performance of our model for all categories maintained strong competitiveness. In essence, such a huge performance improvement was mainly thanks to the extended modules integrating feature correlation and aggregation mechanism.

After further exploring PASCAL VOC, we found that the categorized objects such as "sofa" varied greatly in appearance and sub-type. Therefore, if there were two categories of "person" and "sofa" that should be discriminated at the same time, the classification network tended to give higher weight to "person" and ignore "sofa". And the "bottle" in PASCAL VOC was often located in dense regions and occupied a relatively small proportion. These objective conditions made it difficult for other baselines to obtain high performance for these categories. Nevertheless, the experimental results implied that our detector enabled the network to capture more discriminating features to address this problem, and provided proper attention to each existing category in the image.

In addition, the evaluations of the base and the novel categories on PASCAL VOC split1 were illustrated in Fig. 5. It could be found that for the existing models, there was a certain imbalance between the performance of the base and the novel categories. In other words, the detector "forgot" the configuration of the base categories when training the novel categories, or the novel category samples were too few to learn reliable feature representations. However, it was necessary to maintain a high-level detection for both the base category and the novel category samples. On the basis of our model, the performance of the novel categories in the 3-shot setting achieved the optimal 52.7%(mAP), while that of the base categories reached 69.3%(mAP). The performance of the novel categories in 10-shot arrived at 61.5%(mAP), while that of the base categories reached 71.3%(mAP). As seen in Fig. 5, while the TFA [17] evaluations of the base categories was relatively high, the performance of the novel categories tended to degenerate. By contrast, our model demonstrated balanced performance, which was similar to MPSR [18]. However, the refined branch of the positive samples was manually decided in MPSR, which was actually not ideal. The results above illuminated that the proposed architecture could alleviate the inherent defects that the backbone network either failed to detect the novel categories or "forgot" the base categories.

4.2 Results on MS COCO

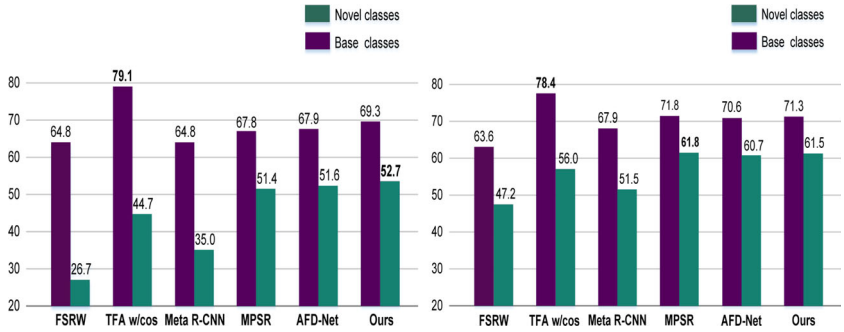
The results on MS COCO [29] were illuminated in Table 3. Compared with PASCAL VOC, MS COCO was more challenging for the task of few-shot object detection. In detail, this dataset contained 80 categories, 20 of which were the same as those in PASCAL VOC. As conducted in FSRW [12], we adopted 60 categories that did not intersect with the categories of PASCAL VOC as the base categories for training in the first phase. Then, in the second phase, 60 base categories and 20 novel categories were trained simultaneously. Each category had K object instances with labels, where $K \in \{10, 30\}$. Following the standard evaluation protocol, we jointly trained the model on both the Train-set and the Val-set benchmarks, and the performance was tested on the Test-set, which contained both the base and the novel categories. For a more comprehensive comparison, our evaluation standard setting were consistent with that in other works like [12, 18, 51], and the indicators consisted of average precision on AP, AP_{50}, AP_{75} and AP_S, AP_M, AP_L and average recall rate on AR, AR_{10}, AR_{100} and AR_S, AR_M, AR_L.

As indicated in Table 3, under the 10-shot and 30-shot settings, the proposed model could achieve optimal performance on most indicators with Average Precision (AP) and Average Recall(AR). It implied that the proposed model enjoyed strong learning robustness in complex scenarios and hard samples. It was worth noting that our model achieved better performance than other baselines when detecting small objects, suggesting that the proposed model could mine discriminating representations through the extended module utilizing the correlation of instance-level features.

Table 2 The AP(%) and mAP(%) of the novel classes in the three splits of VOC

		Novel-class split1						Novel-class split2						Novel-class split3					
Shot	Methods	bird	bus	cow	mbike	sofa	mean	aero	bottle	cow	horse	sofa	mean	boat	cat	mbike	sheep	sofa	mean
1	FSRW [12]	13.5	10.6	31.5	13.8	4.3	14.8	11.8	9.1	15.6	23.7	18.2	15.7	10.8	44.0	17.8	18.1	5.3	21.3
	Meta R-CNN [15]	6.1	32.8	15.0	35.4	0.2	19.9	23.9	0.8	23.6	3.1	0.7	10.4	0.6	31.1	28.9	11.0	0.1	14.3
	MPSR [18]	33.5	**41.2**	**57.6**	**54.5**	**21.6**	41.7	21.2	**9.1**	**36.0**	30.9	25.1	24.4	14.9	47.8	**57.7**	**34.7**	22.8	35.6
	Ours	**49.3**	39.2	44.8	50.5	21.2	41.0	**32.5**	7.9	25.3	**38.3**	**38.8**	**28.6**	**19.3**	**55.5**	54.0	17.0	**40.9**	**37.3**
2	FSRW [12]	21.2	12.0	16.8	17.9	9.6	15.5	28.6	0.9	27.6	0.0	19.5	15.3	5.3	46.4	18.4	26.1	12.4	25.6
	Meta R-CNN [15]	17.2	34.4	43.8	31.8	0.4	25.5	12.4	0.1	44.4	50.1	0.1	19.4	10.6	24.0	36.2	19.2	0.8	18.2
	MPSR [18]	38.2	28.6	**56.5**	57.3	**32.0**	42.5	36.5	**9.1**	45.1	26.1	**34.2**	29.3	**17.9**	49.6	59.2	**49.2**	32.9	41.8
	Ours	**52.0**	**53.0**	46.4	**59.2**	29.3	**47.9**	**36.6**	8.1	**50.0**	**60.2**	16.3	**34.2**	15.4	**67.3**	**59.3**	37.6	**37.9**	**43.5**
3	FSRW [12]	26.1	19.1	40.7	20.4	27.1	26.7	29.4	4.6	34.9	6.8	37.9	22.7	11.2	39.8	20.9	23.7	33.0	28.4
	Meta R-CNN [15]	30.1	44.6	50.8	38.8	10.7	35.0	25.2	0.1	50.7	53.2	18.8	29.6	16.3	39.7	32.6	38.8	10.3	27.5
	MPSR [18]	35.1	60.6	**56.6**	61.5	43.4	51.4	**49.2**	9.1	**47.1**	**46.3**	**44.3**	**39.2**	14.4	60.6	**57.1**	37.2	**42.3**	42.3
	Ours	**50.8**	**64.1**	38.8	**64.7**	**45.3**	**52.7**	42.5	**15.6**	47.4	42.0	36.4	36.8	**23.3**	**65.5**	55.1	**48.0**	36.7	**45.7**
5	FSRW [12]	31.5	21.1	39.8	40.0	37.0	33.9	33.1	9.4	38.4	25.4	44.0	30.1	14.2	57.3	50.8	38.9	41.6	42.8
	Meta R-CNN [15]	35.8	47.9	54.9	55.8	34.0	45.7	28.5	0.3	50.4	**56.7**	38.0	34.8	16.6	45.8	53.9	41.5	**48.1**	41.2
	MPSR [18]	39.7	**65.5**	**55.1**	**68.5**	**47.4**	**55.2**	47.8	10.4	45.2	47.5	**48.8**	39.9	20.9	56.6	**68.1**	48.4	45.8	48.0
	Ours	**58.9**	65.2	42.5	66.8	41.5	55.0	**55.6**	**18.6**	**53.1**	55.1	29.4	**42.4**	**23.8**	**70.0**	63.1	**51.8**	44.9	**50.7**
10	FSRW [12]	30.0	62.7	43.2	60.6	39.6	47.2	43.2	13.9	41.5	58.1	39.2	39.2	20.1	51.8	55.6	42.4	36.6	45.9
	Meta R-CNN [15]	52.5	55.9	52.7	54.6	41.6	51.5	52.8	3.0	52.1	**70.0**	49.2	45.4	13.9	72.6	58.3	47.8	47.6	48.1
	MPSR [18]	48.3	**73.7**	68.2	**70.8**	**48.2**	**61.8**	51.8	16.7	53.1	66.4	**51.2**	47.8	24.4	55.8	67.5	**50.4**	**50.5**	49.7
	Ours	**58.6**	66.2	**68.5**	69.6	44.8	61.5	**59.3**	**18.8**	**58.7**	66.9	43.3	**49.4**	**35.6**	**73.4**	**69.6**	45.1	45.8	**53.9**

Bold entries indicate the best performance achieved in that evaluation setting



(a) $mAP@0.5\,(\%)$ on 3-shot setup. (b) $mAP@0.5\,(\%)$ on 10-shot setup.

Fig. 5 Few-shot detection performance of 3-shot and 10-shot for base and novel categories on PASCAL VOC split1

At the same time, the high recall rate indicated that this proposed model had fine generalization ability for the novel categories, and could quickly learn and detect instances with the novel categories. As the instance number of the novel categories increased, the performance advantage tended to be more obvious.

4.3 Results on MS COCO to PASCAL VOC

Furthermore, a cross-dataset evaluation was performed from MS COCO to PASCAL VOC. Specifically, following the setting in FSRW [12], we adopted 60 categories in MS COCO that did not intersect with PASCAL VOC as the base dataset for detector training, and then tested the trained detectors on VOC07. The network structure and training parameter settings of this evaluation were the same as those used in other experiments. Each category set was made up of 10 samples (10-shot). Then, we compared the cross-domain performance with that of the existing detectors such as FSRW [12], MetaDet [40], Meta R-CNN [15] and MPSR [18], and the like. It could be seen from Table 4 that our model achieved 43.6%(mAP), which was significantly higher than the performance of other models.

4.4 Effectiveness verification

In an ablation study, the contributions of each proposed module were discussed to verify the performance improvement, and the specific reasons were analyzed for the effect of these components. The architecture in this paper was inherited from Meta R-CNN [15] with ResNet-101 [28] as the backbone network. Then, we gradually applied the instance-level feature correlation module, the adaptive shrinkage aggregation module and the orthogonal loss to the backbone architecture, in order to evaluate the contribution of each component to PASCAL VOC split1, with the evaluations listed in Table 5.

Effect of IFC: As shown in Table 5, the baseline could achieve good results in the 5/10-shot setting, but the performance in the extremely-low-shot setups ($K \in \{1, 2, 3\}$) was not satisfactory. This was mainly due to the sudden decrease in visual information that could be learned. The proposed IFC module could further mine dependable representations in instances when the visual information was extremely lacking. Therefore, the model could achieve satisfactory performance when there were only a few learnable samples. Comparing rows 1 and 2 in Table 5, we could acquire significant performance gains of 15.7%(mAP), 17.0%(mAP), and 13.4%(mAP) in extremely-low-shot setups in comparison with the baseline. When $K \in \{5, 10\}$, the proposed model achieved 3.6%(mAP) and 6.2%(mAP) performance gains, respectively. It could be argued that as the amount of learnable visual information increased, the variance decreased, and the performance tended to be stable. The simple correlation construct module was not enough to guide the expended representations of the category distributions, so the gain obtained was not as high as in extremely-low-shot setups.

Effect of ASA: Comparing rows 2 and 3 in Table 5, the performance gains this module brought to the model were 2.9%(mAP), 3.6%(mAP), 3.2%(mAP), 3.9%(mAP), and 1.6%(mAP), respectively when $K \in \{1, 2, 3, 5, 10\}$. This implied the effectiveness of this module in performance improvement, which benefited from the fact that this module established an improved learnable soft-threshold operator to filter out invalid features and clutter signals from the aggregate features. On the other hand, though the meta-learning setting alleviated the influence of overfitting, such an impact still existed in the baseline detection results. The ASA module could further reduce the impact of overfitting, confirmed by the most powerful proof that our model achieved performance improvement in extremely-low-shot setups.

Effect of Orth-loss: Rows 3 and 4 in Table 5 showed that after using orthogonal category loss, the model performance improved by 2.5%(mAP), 1.8%(mAP), 1.1%(mAP), 1.8%(mAP), and 2.2%(mAP), respectively with $K \in \{1, 2, 3, 5, 10\}$. It could be seen from Table 5 that the feature correlation module played an important role in the overall performance improvement. But with the increase in label samples $K \in \{5, 10\}$, the number of negative samples and hard samples also increased, exacerbating the confusion of

Table 3 Few-shot object detection performance on COCO minival set for novel classes

Shot	Methods	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
10	FSRW [12]	5.6	12.3	4.6	0.9	3.5	10.5	10.1	14.3	14.4	1.5	8.4	28.2
	MetaDet [40]	7.1	14.6	6.1	1.0	4.1	12.2	11.9	15.1	15.5	1.7	9.7	30.1
	Meta R-CNN [15]	8.7	19.1	6.6	2.3	7.7	14.0	12.6	17.8	17.9	7.8	15.6	27.2
	FSDetView [16]	12.5	27.3	9.8	2.5	13.8	19.9	**20.0**	25.5	25.7	7.5	27.6	38.9
	MPSR [18]	9.8	17.9	9.7	3.3	9.2	16.1	15.7	21.2	21.2	4.6	19.6	34.3
	CME [51]	15.1	24.6	**16.4**	4.6	**16.6**	26.0	16.3	22.6	22.8	6.6	24.7	**39.7**
	DCNet [52]	12.8	23.4	11.2	4.3	13.8	21.0	18.1	**26.7**	25.6	7.9	24.5	36.7
	FSSP [53]	9.9	20.4	9.6	-	-	-	-	-	-	-	-	-
	SRFS [5]	11.6	21.7	10.4	4.6	10.5	17.2	16.4	23.9	24.1	9.3	21.8	37.7
	DRL/normal [56]	11.9	27.4	7.9	3.8	12.6	17.7	19.7	25.2	25.3	8.6	26.7	35.5
	Ours	**16.7**	**29.2**	15.5	**5.7**	15.9	**26.5**	18.3	26.2	**26.8**	**12.5**	**30.1**	38.7
30	FSRW [12]	9.1	19.0	7.6	0.8	4.9	16.8	13.2	17.7	17.8	1.5	10.4	33.5
	MetaDet [40]	11.3	21.7	8.1	1.1	6.2	17.3	14.5	18.9	19.2	1.8	11.1	34.4
	Meta R-CNN [15]	12.4	25.3	10.8	2.8	11.6	19	15.0	21.4	21.7	8.6	20.0	32.1
	FSDetView [16]	14.7	30.6	12.2	3.2	15.2	23.8	22.0	28.2	28.4	8.3	30.3	42.1
	MPSR [18]	14.1	25.4	14.2	4.0	12.9	23.0	17.7	24.2	24.3	5.5	21.0	39.3
	CME [51]	16.9	28.0	**17.8**	4.6	**18.0**	**29.2**	17.5	23.8	24.0	6.0	24.6	42.5
	DCNet [52]	**18.6**	32.6	17.5	6.9	16.5	27.4	22.8	27.6	28.6	8.4	25.6	**43.4**
	FSSP [53]	14.2	25.0	13.9	-	-	-	-	-	-	-	-	-
	SRFS [5]	15.2	27.5	14.6	6.1	14.5	24.7	18.4	27.1	27.3	9.8	25.1	42.6
	DRL/normal [56]	14.6	31.3	11.3	4.8	15.5	22.3	22.1	28.7	28.8	10.8	30.0	40.4
	Ours	17.9	**33.4**	16.5	**7.1**	17.6	28.6	**23.3**	**30.1**	**30.4**	**12.6**	**32.0**	43.3

We evaluate the performance for different shot examples of novel classes under backbone with ResNet-50

Bold entries indicate the best performance achieved in that evaluation setting

Table 4 Few-shot object detection performance on MS COCO to PASCAL VOC

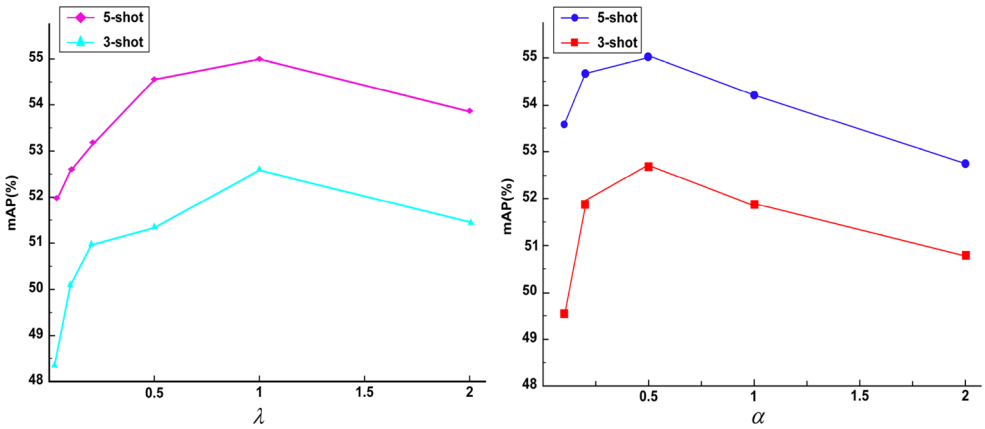
Methods	mAP
FSRW [12]	32.3
MetaDet [40]	33.9
Meta R-CNN [15]	37.4
MPSR [18]	42.3
SRFS [5]	43.2
Ours	**43.6**

We evaluate the performance on the 10-shot setting
Bold entries indicate the best performance achieved in that evaluation setting

Table 5 The results of ablation experiments of the various components proposed on the VOC split1

	VOC split1				
Components/shot	1	2	3	5	10
Baseline	19.9	25.5	35.0	45.7	51.5
Baseline + IFC	35.6	42.5	48.4	49.3	57.7
Baseline + IFC + ASA	38.5	46.1	51.6	53.2	59.3
Baseline + IFC + ASA + Orth-loss	**41.0**	**47.9**	**52.7**	**55.0**	**61.5**

Bold entries indicate the best performance achieved in that evaluation setting



(a) The effect of λ on performance. (b) The effect of α on performance.

Fig. 6 The comparison the impact of hyperparameters λ and α in the 3/5-shot setting of VOC split1

Table 6 Comparative analysis of the parameters of neural network for few-shot object detection

Methods	Receptive field size	Params/M	Inference speed(ms/per)
Meta R-CNN [15]	short600	45.98	165.42
FSDetView [16]	short600	50.39	174.84
TFA w/fc [17]	short800	60.08	168.33
MPSR [18]	short800	81.84	177.50
DCNet [52]	short800	100.96	–
CME [51]	416×416	66.81	213.85
Ours	short600	79.17	196.18

inter-category and intra-category of features. However, this loss had strong adaptability and effectiveness in solving this confusion problem. In addition, we attempted to modify the hyperparameters (α, λ) in the loss function to observe their impacts on detector performance. Experiments were evaluated on PASCAL VOC Split1. The analysis results were shown in Fig. 6. It could be noticed that the model enjoyed the best performance when $\alpha = 0.5$ and $\lambda = 1$.

Moreover, since different model architectures are usually performed on different platforms of GPU and also the corresponding sizes of input image are different, we reproduced some existing models following the original settings for a fair costing comparison. Table 6 illuminates the scales of network parameters for these models, as well as the inference speed of testing a single image at the 10-shot setting. Combined with Tables 1 and 3, we found it is false that the more parameters, the better the performance. This also implies that the key to few-shot learning might lie in how to determine the more efficient transformable configurations and then how to make the model have a higher learning efficiency for each novel sample. Obviously the proposed method has a certain improvement in both the aspects.

4.5 Visualization of results

In Fig. 7, we finally presented the detection visualization results of the novel categories on PASCAL VOC and MS COCO in a 10-shot setting. Both the successful cases and the failure cases are involved to analyze the inherent reasons. These results verified that the proposed model achieves excellent performance, but there still existed few failure cases. And we inferred that the model might generate false foreground instances when there were more interference backgrounds in the image. Besides, there would also exist missed instances when smaller and denser objects were exist in the regions.

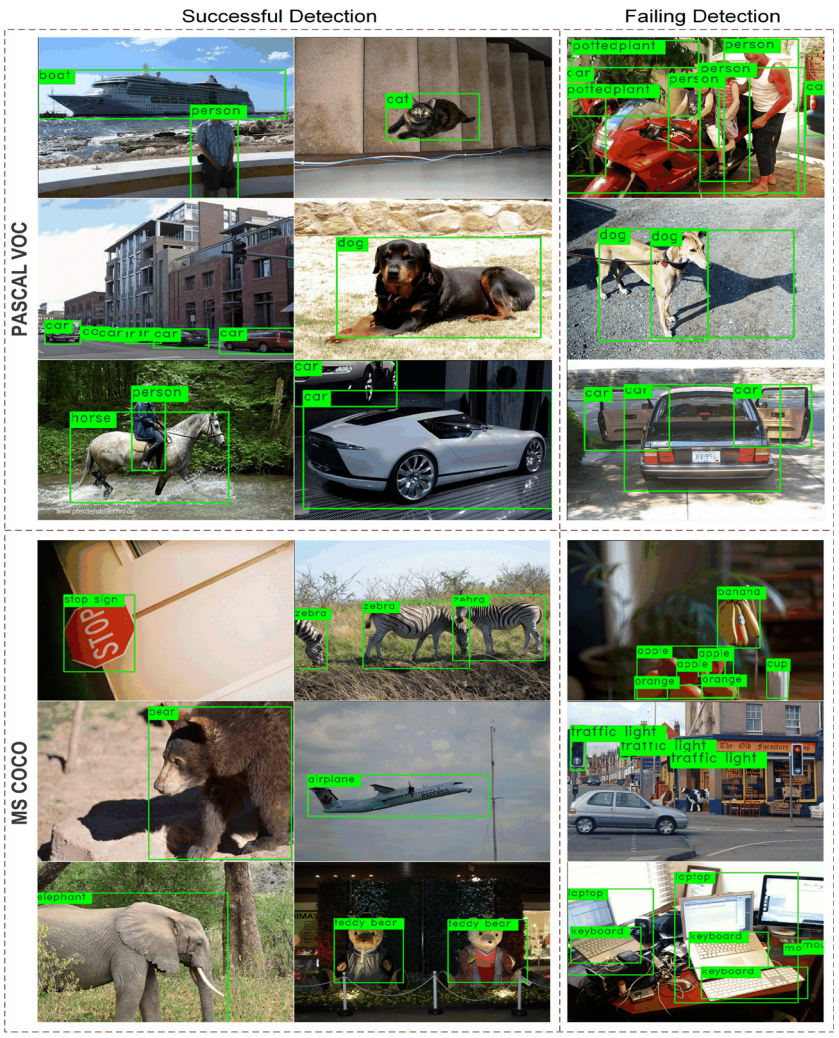


Fig. 7 Visualization results of 10-shot setting on PASCAL VOC and MS COCO datasets. The two columns on the left are success cases and the failure cases are on the right

5 Conclusion

This paper aims to formulate a few-shot object detector based on meta-learning. Most existing approaches perform inefficient baselines to learn features from few samples through single feature extraction strategies. In this paper, a feature correlation building module based on self-attention has been proposed to mine representative discriminating representations. By a learnable soft-threshold operator, the proposed architecture has been enhanced to shrink redundant information from aggregate features. In addition, orthogonal loss has been used to further constrain instance representations and avoid category confusion. The experimental results have verified that the proposed detector has higher detection precision compared with the existing related baselines through the comprehensive evaluations on AP, mAP and AR of two benchmark datasets. In the future, more researches can be conducted to reduce model complexity and alleviate high cost in model training. Moreover, specific dynamic memory mechanism can be considered in the field of few-shot detection so as to further enhance performance.

Acknowledgements The research was supported by the National Natural Science Foundation of China (62062048).

References

1. Vinyals O, Blundell C, Lillicrap T et al (2016) Matching networks for one shot learning[J]. Adv Neural Inf Process Syst 29:3630–3638
2. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks[C]//International Conference on Machine Learning. PMLR, pp 1126–1135
3. Snell J, Swersky K, Zemel RS (2017) Prototypical networks for few-shot learning[J]. arXiv:1703.05175
4. Lee K, Maji S, Ravichandran A et al (2019) Meta-learning with differentiable convex optimization[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 10657–10665
5. Kim G, Jung H G, Lee S W (2021) Spatial Reasoning for Few-Shot Object Detection[J]. Pattern Recogn:108118
6. Ravi S, Larochelle H (2016) Optimization as a model for few-shot learning[J]
7. Sung F, Yang Y, Zhang l et al (2018) Learning to compare: Relation network for few-shot learning[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1199–1208
8. Bertinetto L, Henriques JF, Torr PHS et al (2018) Meta-learning with differentiable closed-form solvers[J]. arXiv:1805.08136
9. Gidaris S, Komodakis N (2018) Dynamic few-shot visual learning without forgetting[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4367–4375
10. Chen H, Wang Y, Wang G et al (2018) Lstd: A low-shot transfer detector for object detection[C]. Proc AAAI Conf Artif Intell 32(1)
11. Karlinsky L, Shtok J, Harary S et al (2019) Repmet: Representative-based metric learning for classification and few-shot object detection[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5197–5206
12. Kang B, Liu Z, Wang X et al (2019) Few-shot object detection via feature reweighting[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 8420–8429
13. Fan Q, Zhuo W, Tang CK et al (2020) Few-shot object detection with attention-RPN and multi-relation detector[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4013–4022
14. Li Y, Feng W, Lyu S et al (2020) MM-FSOD: Meta and metric integrated few-shot object detection[J]. arXiv:2012.15159
15. Yan X, Chen Z, Xu A et al (2019) Meta r-cnn: Towards general solver for instance-level low-shot learning[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9577–9586
16. Xiao Y, Marlet R (2020) Few-shot object detection and viewpoint estimation for objects in the wild[C]. European Conference on Computer Vision. Springer, Cham, pp 192–210
17. Wang X, Huang TE, Darrell T et al (2020) Frustratingly simple few-shot object detection[J]. arXiv:2003.06957
18. Wu J, Liu S, Huang D et al (2020) Multi-scale positive sample refinement for few-shot object detection[C]. European Conference on Computer Vision. Springer, Cham, pp 456–472
19. Yang Z, Wang Y, Chen X et al (2020) Context-transformer: tackling object confusion for few-shot detection[C]. Proc AAAI Conf Artif Intell 34(07):12653–12660
20. Everingham M, Eslami SMA, Van Gool L et al (2015) The pascal visual object classes challenge: A retrospective[J]. Int J Comput Vis 111(1):98–136
21. Liu L, Ma B, Zhang Y et al (2020) AFD-Net: Adaptive Fully-Dual Network for Few-Shot Object Detection[J]. arXiv:2011.14667
22. Jiao L, Zhang F, Liu F et al (2019) A survey of deep learning-based object detection[J]. IEEE access 7:128837–128868
23. Isogawa K, Ida T, Shiodera T et al (2017) Deep shrinkage convolutional neural network for adaptive noise reduction[J]. IEEE Signal Process Lett 25(2):224–228
24. Everingham M, Van Gool L, Williams CKI et al (2010) The pascal visual object classes (voc) challenge[J]. Int J Comput Vis 88(2):303–338
25. Ren S, He K, Girshick R et al (2015) Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Adv Neural Inf Process Syst 28:91–99
26. Bendre N, Marín HT, Najafirad P (2020) Learning from few samples: A survey[J]. arXiv:2007.15484
27. Law H, Teng Y, Russakovsky O et al (2019) Cornernet-lite: Efficient keypoint based object detection[J]. arXiv:1904.08900
28. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
29. Lin TY, Maire M, Belongie S et al (2014) Microsoft coco: Common objects in context[C]. European conference on computer vision. Springer, Cham, pp 740–755
30. Liu W, Anguelov D, Erhan D et al (2016) Ssd: Single shot multibox detector[C]. European conference on computer vision. Springer, Cham, pp 21–37
31. Redmon J, Divvala S, Girshick R et al (2016) You only look once: Unified, real-time object detection[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
32. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7263–7271
33. Wang Y, Yao Q, Kwok JT et al (2020) Generalizing from a few examples: A survey on few-shot learning[J]. ACM Comput Surv (CSUR) 53(3):1–34

34. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement[J]. arXiv:1804.02767
35. Ranasinghe K, Naseer M, Hayat M et al (2021) Orthogonal Projection Loss[J]. arXiv:2103.14021
36. Deng J, Dong W, Socher R et al (2009) Imagenet: A large-scale hierarchical image database[C]. 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255
37. Duan K, Bai S, Xie L et al (2019) Centernet: Object detection with keypoint triplets[J]. arXiv:1904.08189
38. Law H, Deng J (2018) Cornernet: Detecting objects as paired keypoints[C]. Proceedings of the European conference on computer vision (ECCV), pp 734–750
39. Gidaris S, Bursuc A, Komodakis n et al (2019) Boosting few-shot visual learning with self-supervision[C]. Proceedings of the IEEE/CVF international conference on computer vision, pp 8059–8068
40. Wang YX, Ramanan D, Hebert M (2019) Meta-learning to detect rare objects[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9925–9934
41. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks[C]. International Conference on Machine Learning. PMLR, pp 1126–1135
42. Fei-Fei L, Fergus R, Perona P (2006) One-shot learning of object categories[J]. IEEE Trans Pattern Anal Mach Intell 28(4):594–611
43. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction[J]. Science 350(6266):1332–1338
44. Zhao M, Zhong S, Fu X et al (2019) Deep residual shrinkage networks for fault diagnosis[J]. IEEE Trans Industrial Inf 16(7):4681–4690
45. Chen X, Jiang M, Zhao Q (2020) Leveraging Bottom-Up and Top-Down Attention for Few-Shot Object Detection[J]. arXiv:2007.12104
46. Wang X, Girshick R, Gupta A et al (2018) Non-local neural networks[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
47. Zhou X, Zhuo J, Krahenbuhl P (2019) Bottom-up object detection by grouping extreme and center points[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 850–859
48. Fu J, Liu J, Tian H et al (2019) Dual attention network for scene segmentation[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3146–3154
49. Chen TI, Liu YC, Su HT et al (2021) Should I Look at the Head or the Tail? Dual-awareness Attention for Few-Shot Object Detection[J]. arXiv:2102.12152
50. Kim G, Jung HG, Lee SW (2020) Few-shot object detection via knowledge transfer[C]. 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp 3564–3569
51. Li B, Yang B, Liu C et al (2021) Beyond Max-Margin: Class Margin Equilibrium for Few-shot Object Detection[C]. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7363–7372
52. Hu H, Bai S, Li A et al (2021) Dense Relation Distillation with Context-aware Aggregation for Few-Shot Object Detection[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10185–10194
53. Xu H, Wang X, Shao F et al (2021) Few-Shot Object detection via sample Processing[J]. IEEE Access 9:29207–29221
54. Zhu C, Chen F, Ahmed U et al (2021) Semantic relation reasoning for shot-stable few-shot object detection[C]. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 8782–8791
55. Lee H, Lee M, Kwak N (2021) Few-Shot Object Detection by Attending to Per-Sample-Prototype. arXiv:2109.07734
56. Liu W et al (2021) Dynamic Relevance Learning for Few-Shot Object Detection. arXiv:2108.02235

Meng Wang received the Ph.D. degree in control theory and control engineering from the Beijing Institute of Technology, Beijing, China, in 2014. He is currently an Associate Professor with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. His research interests include image processing, computer vision, and pattern recognition.

Hongwei Ning is currently pursuing the master's degree with Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, China. His research interests include artificial intelligence, machine learning and computer vision, focusing on few-shot object detection technology.

Haipeng Liu received the Ph.D. degree in Power Electronics and Power Drives from the North China Electric Power University, Beijing, China, in 2015. He is currently an senior experimentalist with the Faculty of Information Engineering and Automation, Kunming University of Science and Technology. His research interests include image processing, and data forecast.