Few-Shot Object Detection With Self-Supervising and Cooperative Classifier

Di Qi, Jilin Hu, *Member, IEEE*, and Jianbing Shen, *Senior Member, IEEE*

***Abstract*— Few-shot object detection (FSOD), which detects novel objects with only a few training instances, has recently attracted more attention. Previous works focus on making the most use of label information of objects. Still, they fail to consider the structural and semantic information of the image itself and solve the misclassification between data-abundant base classes and data-scarce novel classes efficiently. In this article, we propose FSOD with Self-Supervising and Cooperative Classifier (FS^{3}C) approach to deal with those concerns. Specifically, we analyze the underlying performance degradation of novel classes in FSOD and discover that false-positive samples are the main reason. By looking into these false-positive samples, we further notice that misclassifying novel classes as base classes are the main cause. Thus, we introduce double RoI heads into the existing Fast-RCNN to learn more specific features for novel classes. We also consider using self-supervised learning (SSL) to learn more structural and semantic information. Finally, we propose a cooperative classifier (CC) with the base–novel regularization to maximize the interclass variance between base and novel classes. In the experiment, FS^{3}C outperforms all the latest baselines in most cases on PASCAL VOC and COCO.**

***Index Terms*— Cooperative classifier (CC), few-shot object detection (FSOD), self-supervised learning (SSL).**

I. Introduction

THE vision community has achieved remarkable success in image classification [5] and object detection [6], due to the emergence of deep-learning frameworks. However, such frameworks always require many manually annotated training datasets that are labor-intensive and sometimes inaccessible. This observation induces many researchers to rethink the generalization of deep-learning methods [7]. Thus, few-shot learning (FSL) and self-supervised learning (SSL) that learn from data with few or no labels attract more and more attention. There are already many attempts of FSL on image classification [8]. However, few efforts have been put into the more challenging task of few-shot object detection (FSOD) [9], [10], [11], [12], [13] as it involves not only class classification, but also the location prediction of objects.

The existing studies of FSOD can be classified into the following five categories: 1) transfer learning [14]; 2) metric learning [1], [10], [15]; 3) meta learning [9], [13], [15]; 4) fine tuning [2], [3], [11]; and 5) multiscale refinement [12]. All those methods try to learn feature representations from base classes, such that they can be applied in novel classes [11], [16]. However, these powerful and comprehensive pretrained features can do evil in FSOD. For example, in Fig. 1, the background (BG) of the query image is *green grass*, but if none of the training shots for *horse* contain this BG, then the detector is likely to mistake the *horse* on the grass as a *cow* which is usually contained in the BG of *green grass*. How to ignore meaningless features remains an ongoing challenge in FSOD. Moreover, the existing FSOD methods are mainly trying to learn a metric function by only minimizing the intraclass variance [11]. However, this may not be enough for novel classes with only a few labeled data available.

In this article, we propose a simple yet effective fine-tune-based approach—cooperative classifier (CC)—to facilitate FSOD, named FSOD with Self-Supervising and Cooperative Classifier (FS^{3}C). First, we try to figure out the reason for the performance degradation of FSOD in novel classes. Based on this analysis, we change the shared RoI head to double RoI heads for object localization (Loc) and classification [4], respectively. This change is beneficial for novel classes with few labeled data, such that each RoI head can learn more specific features for two different tasks. Next, we introduce SSL into the existing object detection framework without additional training data. It can then improve the structural information in feature maps, reducing the effect of inessential but confusing features. To integrate with the detection task, we adopt three different SSL tasks: pixel-level contrastive learning [17], patch-level *Jigsaw puzzle* [18] prediction, and instance-level image *Rotation* [19], and carefully analyze their characteristics. Finally, we incorporate the CC into the primary classification head to exploit the relationships between base and novel classes. A supervised classification objective for a CC is optimized to push novel classes away from base classes, thereby alleviating the misclassification between the base and novel objects. To sum up, our contributions are as follows.

1) We figure out that existing fine-tuning-based methods can easily classify novel objects into base classes, which

Manuscript received 20 December 2021; revised 30 May 2022; accepted 20 August 2022. This work was supported in part by the Science and Technology Development Fund (FDCT) under Grant 0154/2022/A3 and Grant SKL-IOTSC(UM)-2021-2023, in part by Grant MYRG-CRG2022-00013-IOTSC-ICI, and in part by Grant SRG2022-00023-IOTSC. *(Corresponding authors: Jilin Hu; Jianbing Shen.)*

Di Qi is with the School of Computer Science, Beijing Institute of Technology, Beijing 100811, China.

Jilin Hu is with the Department of Computer Science, Aalborg University, 9220 Aalborg, Denmark (e-mail: hujilin1229@gmail.com).

Jianbing Shen is with the State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, Macau, China (e-mail: shenjianbingcg@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3204597.

Digital Object Identifier 10.1109/TNNLS.2022.3204597

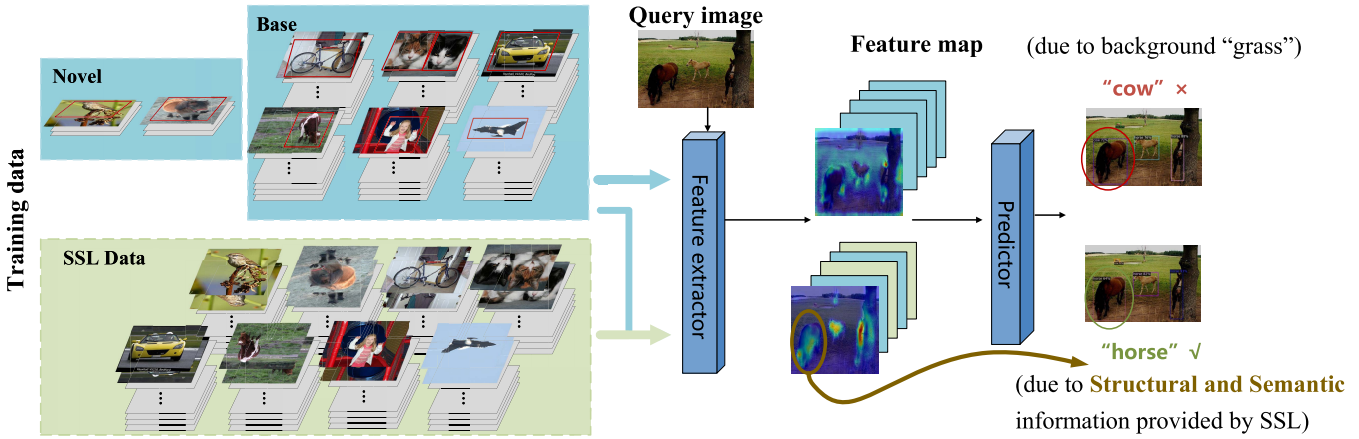


Fig. 1. **Illustration of the proposed unifying FS^{3}C.** The training data consists of base classes data, novel classes data, and SSL data. Then, the feature extractor is learned by taking input as the training data, where the SSL data helps to strengthen the structural and semantic information as is shown in the lower part of the "Feature map." This information can help to classify the object correctly, for example, "horse" in the query image, while not "cow" that is misled by the "green grass" feature in the BG.

is the main reason for the performance degradation in FSOD.

2) We introduce SSL into FSOD with auxiliary tasks in terms of pixel-level, patch-level, and instance-level to learn more structural and semantic information, reducing the impact of inessential features from the base classes.
3) We introduce a CC as a regularization for the base–novel classes. So, it can maximize the interclass variance between base and novel classes, reducing misclassification.
4) We further conduct extensive experiments on two popular object detection datasets, the COCO [20] and the PASCAL VOC [21], in few-shot settings to demonstrate our performance that can outperform all the latest baselines.

II. Related Work

A. Few-Shot Learning

FSL aims to imitate the human ability to transfer prior knowledge to new tasks in a small data regime. Fei-Fei *et al.* [8] generalized knowledge from a pretrained model to perform one-shot learning by using Bayesian inference. Lake *et al.* [22] proposed a hierarchical Bayesian one-shot learning system that exploits compositionality and causality. Recently, metric learning is gradually becoming a promising solution for FSL. The most intuitive metrics include cosine similarity [23], [24], [25], [26], Euclidean distance to class center [27], and graph distances [28]. In particular, prototypical models [27] converted the spatial semantic information of objects to convolutional channels. Matching Network [29] performed weighted nearest neighbor matching by encoded features to classify query images. Relation Network [30] learned a distance metric to compare the target image with a few labeled images. Existing research observed that class margin greatly influences classifiers when the model discriminability needs to be guaranteed. However, the novel classes to be reconstructed would improve the diversity of novel classes, so only pursuing max-margin could be infeasible. Li *et al.* [31] proposed the adaptive margin loss and a class-relevant additive margin loss to improve generalization ability. Liu *et al.* [32] introduced a negative class margin to benefit the representation of novel classes. Interestingly, some other methods utilized data argumentation to generate additional examples for unseen categories. Wang *et al.* [33] solved the data deficiency via learning to generate fake data. Moreover, optimization-based methods [34], [35] were proposed for fast adaptation to new few-shot tasks. Hou *et al.* [36] proposed a cross-attention mechanism to learn correlations between support and query images. The above methods offer promising solutions to image classification tasks. In comparison, few-shot detection is a more challenging research topic.

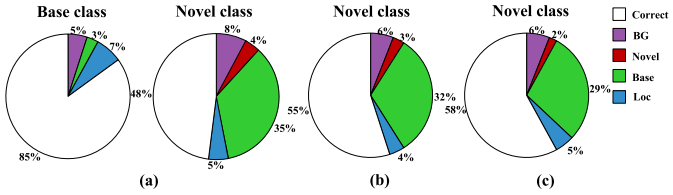


Fig. 2. **Error analysis of false positives in the three-shot setting of VOC split 1.** Pie charts indicate the fraction of top-ranked false positives that are due to poor Loc, confusion with base objects (base), confusion with novel objects (novel), or confusion with BG or unlabeled objects (BG). Note that "Correct" refers to the correct samples. (a) TFA w/cos. (b) SSL. (c) SSL + CC.

B. Few-Shot Object Detection

FSOD is a task of locating and classifying objects in images with very few training examples. Chen *et al.* [14] first proposed a novel FSOD framework via transfer learning. Karlinsky *et al.* [10] applied metric learning to model multimodal distribution for each object class. Recent works attempted to solve this problem with meta-learning. To learn a meta-learner, Kang *et al.* [9] and Yan *et al.* [13] integrated a re-weighting module to YOLOv2 [38] and Faster R-CNN [6], respectively. To take advantage of both meta-learning and metric learning, Fan *et al.* [15] adjusted the position of

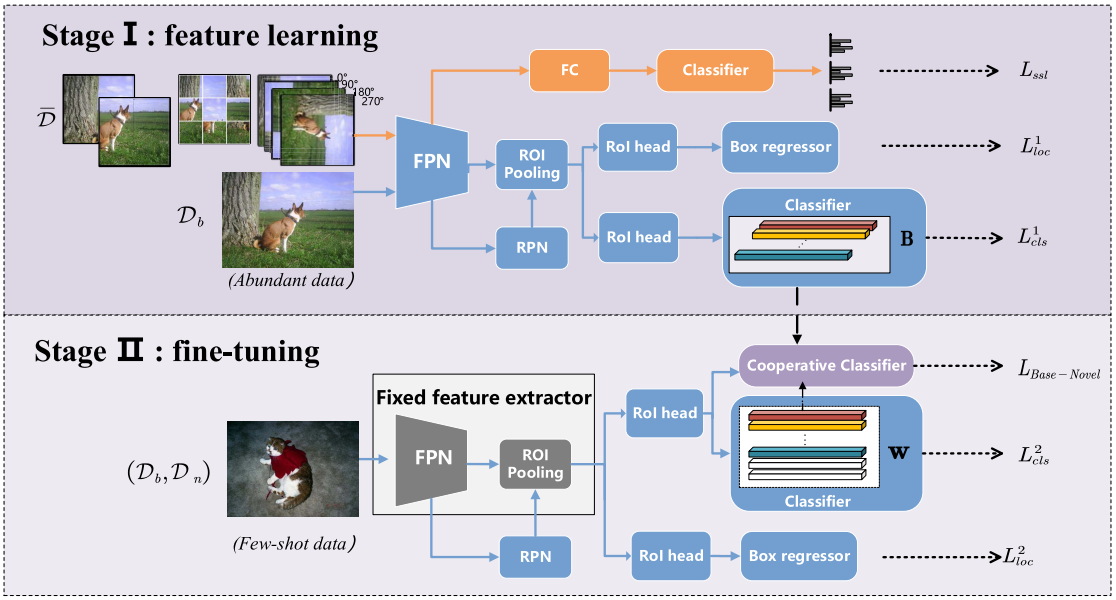


Fig. 3. **Overview of our approach, FS^{3}C. Top: training self-supervised tasks (image rotation, jigsaw puzzle prediction, and pixel-level contrastive learning) and supervised base class object detection in a multitask setting.** Visual features of input images from $\bar{\mathcal{D}}$ and $\mathcal{D}_b$ are first extracted by the weight-shared backbone and then fed into the SSL classifier following the orange line and box predictor following the blue line, respectively. **Bottom: introducing novel CC into Faster R-CNN during the few-shot fine-tuning stage.** In the fine-tuning stage, proposal features from the fixed pretrained feature extractor for few-shot data are sent into the predictor, where a CC is proposed paralleling the original classification branch.

the re-weighting module slightly. Xiao and Marlet [39] refined the category-specific embedding by optimizing the re-weighting module. Li and Li [70] proposed Transformation Invariant Principle (TIP) to expand samples to various meta-learning models, which can boost the detection performance of novel class objects. DAnA-FasterRCNN [67] proposed dual-awareness attention to capture the correlations between objects. Recently, the fine-tuning approach has been demonstrated to outperform other complex methods [11]. Wu *et al.* [12] proposed a multiscale positive sample refinement strategy to deliver a solid FSOD solution. Sun *et al.* [2] added a contrastive branch to the primary RoI head to learn contrastive-aware object proposal encodings. Fan *et al.* [69] proposed to balance the bias in the pretrained RPN and propose a re-detector to find few-shot class objects. DeFRCN [66] performed decoupling among multiple components of Faster R-CNN by proposing GDL and PCB. However, almost all these previous methods ignore considering the relationships between novel and base classes. In this article, we further improve the performance with simple fine-tuning by alleviating the misclassification between novel and base classes.

C. Self-Supervised Learning

SSL is a feature learning strategy that exploits a variety of labels that come with data for free. There are many previous works devoted to designing effective pretext tasks to optimize SSL, which can be classified into the following categories.

1) Distortion, for example, predicting rotation [19].
2) Patches, for example, jigsaw puzzle [18], [40], divides an image into nine parts and then predicts the relative position to generate a loss.
3) Colorization [41], [42]. According to the grayscale image, it predicts the color of the image.
4) Contrastive learning, where the positive pair is usually formed with two augmented views of the same image, while negative ones are formed with different images. It holds the key to most state-of-the-art (SOTA) methods [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. 5) Generative modeling, which learns meaningful latent representation via reconstructing the original input [53].

Recently, Doersch and Zisserman [54] pointed out that combining different SSL tasks can be beneficial. More recently, Goyal *et al.* [55] and Kolesnikov *et al.* [56] concluded that jigsaw puzzles and rotations are the two most effective SSL tasks without considering contrastive learning. However, previous works have hardly been applied to the FSOD domain yet. Among these, TIP [70] simply treated contrastive learning as sample expansion. FSCE [2] proposed a contrastive branch to guide the RoI head and learn contrastive-aware object proposal embeddings, ignoring the SSL benefits in feature extraction. In this work, we exploit the complementary of these two domains. By unifying SSL into FSOD with auxiliary tasks, our model can extract a better-targeted feature map.

III. Preliminaries

Suppose we have a base class set C_b and a novel class set C_n, in which $C_b \cap C_n = \emptyset$. Correspondingly, we have a large-scale annotated base dataset $\mathcal{D}_b = \{(x_i, y_i)\}$, where x_i are input images and y_i are the labels for objects of base classes in x_i. Then following the same setting introduced in [9], we construct the novel-class dataset $\mathcal{D}_n = \{(x_i, y_i)\}$,

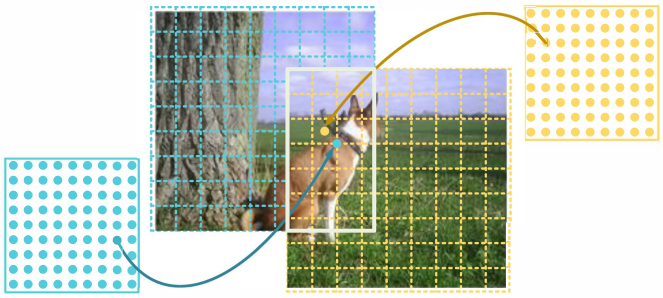


Fig. 4. **Illustration of pixel-level contrastive learning.** In our method, two views are randomly cropped from an image, and there is an intersection between them (outlined in white). We send these two views into the backbone to extract feature maps and regard each pixel vector in the two feature maps as a sample (shown as a yellow and blue dot). In the implementation, we use the diagonal length of a feature map bin in the original image space as the threshold and compute the distances between all pairs of pixels from the two feature maps to construct positive and negative pairs.

in which objects have labels belonging to C_n and the number of objects per category is K to meet the requirement of K-shot learning. In addition, we construct an SSL dataset $\bar{\mathcal{D}} = \{\text{pretext}(x_i)\}$, where pretext$(\cdot)$ denotes the pretext tasks that apply on images x_i from $\mathcal{D}_b$.

The final goal of general FSOD is to learn a detection algorithm via the training data $(\mathcal{D}_b, \mathcal{D}_n)$, such that it can have a good detection average precision (AP) on the novel classes (nAP) as well as base classes (bAP). In this article, we adopt a two-phase training process that consists of: 1) feature learning phase (FLP) and 2) fine-tuning phase (FTP).

A. Performance Degradation in Novel Class

We first seek the reason for detection performance differences between novel and base classes in FSOD. In the beginning, we take a closer look at results in TFA [11] and notice a very similar recall in both base and novel classes, which are 0.6 versus 0.59 in three shots, and 0.69 versus 0.68 in ten shots on PASCAL VOC Set1. Thus, the low precision can account for the performance degradation in novel classes, which is caused by the increased number of false positives. Next, we try to figure out the frequency and impact of different types of false positives using a diagnosing tool [57], whose results are shown in Fig. 2(a). In this figure, we cannot see a clear distinction between either poor Loc or confusion with BG. Yet, we can observe that the most significant difference between base and novel classes is misclassification, which increases from 3% to 39% (4% in red and 34% in green). This phenomenon is also observed in FSCE [2].

Due to the complementary between classification and Loc, previous object detectors usually share an RoI head for Loc and classification, trained with a large amount of labeled data. However, in FSOD, it is not easy to fine-tune this single RoI head to fit both Loc and classification with few annotations. For this reason, we adopt double RoI heads for Loc and classification, respectively. Furthermore, we unfreeze the double RoI heads and RPN during fine-tuning stage for better representation learning. As shown in Table I, our double RoI heads work better in a few-shot setting for novel classes.

Algorithm 1 Pixel-Level Contrastive Learning

Require: batch size N, total number pixels in a feature map S, constant τ, structure of $f, g, \mathcal{T}$

for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for $k = 1$ to N **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\hat{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\hat{\boldsymbol{x}}_{2k-1})$
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$
    # the second augmentation
    $\hat{\boldsymbol{x}}_{2k} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\hat{\boldsymbol{x}}_{2k})$
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$
    # pairwise similarity
    for pixels $\{(\hat{\mathbf{p}}_m, \hat{\mathbf{p}}_n)\}_{m,n=1}^{S}$ in $(\boldsymbol{z}_{2k-1}, \boldsymbol{z}_{2k})$ **do**
      $s_{m,n}^{2k-1,2k} = \cos(\hat{\mathbf{p}}_m, \hat{\mathbf{p}}_n)$
      compute the ground truth similarity $\hat{y}_{mn}$
    end for
  end for
  # define loss
  $\ell(2k-1, 2k) = -\frac{1}{S}\sum_{m=1}^{S}\log\frac{\sum_{n=1}^{S}\mathbb{1}_{[\hat{y}_{mn}=1]}\exp(s_{m,n}^{2k-1,2k}/\tau)}{\sum_{n=1}^{S}\exp(s_{m,n}^{2k-1,2k}/\tau)}$
  $\mathcal{L}_{pixel} = \frac{1}{2N}\sum_{k=1}^{N}[\ell(2k-1, 2k) + \ell(2\,k, 2\,k-1)]$
  update networks f and g to minimize the loss $\mathcal{L}_{pixel}$
end for
return encoder network $f(\cdot)$, throw away predictor $g(\cdot)$

Note that DeFRCN [66] also paid attention to this issue but adopted another implicit approach to solving it.

IV. Framework of FS^{3}C

As shown in Fig. 3, we adopt the widely used Faster R-CNN as our base model. To better adapt to a few-shot setting, we first decompose the shared RoI head into two separate RoI heads for both classification and Loc, respectively [4].

We then unify SSL with the backbone feature extractor during FLP to strengthen the structural information. We construct three auxiliary tasks at pixel-level, patch-level, and instance-level. We build a weight-shared network with two branches, where one branch is for $\bar{\mathcal{D}}$ and the other is for $\mathcal{D}_b$. The visual features of input images are extracted by the weight-shared backbone, which is further fed into the SSL classifier and box predictor to obtain the prediction, respectively.

Finally, we propose a CC branch and a novel regularization (namely base–novel regularization) to enhance the fine-tuning and alleviate the misclassification between base and novel classes. During FTP, we continue to use the learned feature extractor in FLP by removing the SSL branch and fixing the entire parameters in the feature extractor.

A. Self-Supervised Learning

The goal of a few-shot learner is to learn representations from the base classes, such that it can lead to good generalization of novel classes. As stated in [11] and [16], features learned from the base classes are likely to transfer

TABLE I
PERFORMANCE EVALUATION (NAP 50) OF OUR DOUBLE HEAD BASELINE ON PASCAL VOC SPLIT 1. **Bold** INDICATES SOTA

Method	Double head	unfreeze RPN + RoI head	base AP50		novel AP50	
			shot=3	shot=10	shot=3	shot=10
TFA w/cos [11]			**79.1**	78.4	44.7	56.0
baseline (Ours)	√		78.0	78.3	45.8	57.5
	√	√	78.8	**78.4**	**47.8**	**58.3**

to the novel classes without further parameter updates due to the class-agnostic feature extraction given a large number of labeled examples in base classes. However, the scarce fine-tuned data may make the model pay too much attention to the prominent but inessential information robust in the pre-trained feature, for example, "green grass" in Fig. 1. Therefore, ignoring the meaningless feature remains an ongoing challenge in FSOD.

This section introduces how to unify SSL into a few-shot detection field and utilize the image itself to strengthen the high-level structural information in feature maps and mitigate the effect of inessential but confusing features. SSL is usually constructed to learn feature representations, which are then used by the downstream tasks to do traditional supervised learning. Differently, in this article, we unify SSL with the traditional object detection framework and treat the losses of SSL tasks as regularizers during the FLP.

To better integrate with the detection task, we consider the following three different SSL tasks: the instance-level task, that is, *Rotation*; the patch-level task, that is, *Jigsaw puzzle* prediction; and the pixel-level task, that is, variant of SimCLR [17]. Fig. 3 shows these three tasks, and detailed settings are introduced below.

1) *Rotation:* Given an image $\bar{x} \in \bar{\mathcal{D}}$, we rotate $\bar{x}$ by an angle $\theta \in \{0°, 90°, 180°, 270°\}$ to obtain $\hat{x}$ as input data and the index of angle $\hat{y}$ as target label.
2) *Jigsaw puzzle:* Given an image $\bar{x} \in \bar{\mathcal{D}}$, we tile $\bar{x}$ into 3×3 regions to obtain $\hat{x}$ as input data, and the index of permutation $\hat{y}$ as target label. Note that the permutation number is reduced from 9! to 35 by following the setting in [18]. Hence, the labeled data $(\hat{x}, \hat{y})$ are derived automatically without any human labeling, and the loss is defined as follows:

$$\mathcal{L}_{\text{patch/instance}} = \sum_{\bar{x}_i \in \bar{\mathcal{D}}} L_{\text{CE}}(g \circ f(\hat{x}_i), \hat{y}_i) \tag{1}$$

where g is the function to predict labels of SSL tasks, by taking input as features extracted from the backbone f.
3) *Pixel-level task:* In Fig. 4, we first transform $\bar{x}$ randomly resulting in two correlated views as input data, denoted $\hat{x}_i$ and $\hat{x}_j$, which are then sent into backbone f and predictor g to extract feature maps. Unlike SimCLR [17], we regard each pixel vector in the two feature maps as a sample, and construct positive and negative pairs by computing the distances in the original image space between all pairs of pixels $\{(\hat{\mathbf{p}}_m^i, \hat{\mathbf{p}}_n^j)\}_{m,n=1}^S$ from the two feature maps, where S is the total number pixels in a feature map, $\hat{\mathbf{p}}_m^i \in \mathbb{R}^c$ and $\hat{\mathbf{p}}_n^j \in \mathbb{R}^c$ are the mth and nth pixel vectors in the feature map of $\hat{x}_i$ and $\hat{x}_j$, respectively, and c denotes the feature dimension. Concretely, if the distance between $\hat{\mathbf{p}}_m^i$ and $\hat{\mathbf{p}}_n^j$ is less than half of diagonal length of a feature map bin in the original image space, we consider $(\hat{\mathbf{p}}_m^i, \hat{\mathbf{p}}_n^j)$ as a positive pair and set $\hat{y}_{mn} = 1$, otherwise $\hat{y}_{mn} = 0$. Then for image $\bar{x}$, the loss function for all the positive pairs of $\{\hat{\mathbf{p}}_m^i\}_{m=1}^S$ in $\{\hat{\mathbf{p}}_n^j\}_{n=1}^S$ is

$$\ell(\hat{x}_i, \hat{x}_j) = -\frac{1}{S}\sum_{m=1}^{S} \log \frac{\sum_{n=1}^{S} \mathbb{1}_{[\hat{y}_{mn}=1]} \exp(\cos(\hat{\mathbf{p}}_m^i, \hat{\mathbf{p}}_n^j)/\tau)}{\sum_{n=1}^{S} \exp(\cos(\hat{\mathbf{p}}_m^i, \hat{\mathbf{p}}_n^j)/\tau)} \tag{2}$$

where $\cos(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ is the cosine similarity between $\hat{\mathbf{p}}_m^i$ and $\hat{\mathbf{p}}_n^j$, and τ denotes a temperature parameter. So, the pixel-level SSL loss can be formulated as follows:

$$\mathcal{L}_{\text{pixel}} = \sum_{\bar{x} \in \bar{\mathcal{D}}} (\ell(\hat{x}_i, \hat{x}_j) + \ell(\hat{x}_j, \hat{x}_i)). \tag{3}$$

Algorithm 1 summarizes pixel-level contrastive learning. ReSim [68] is the most related work that learns regional representations for Loc. Differently, ReSim [68] selects areas from the image, and the overlapping areas in different views are positive pairs. However, our pixel-level contrastive learning selects samples directly from the feature maps. It constructs positive and negative sample pairs based on a slack threshold of distance, which benefits semantic representation learning.

Therefore, the overall SSL loss is defined as follows:

$$\mathcal{L}_{\text{ssl}} = \lambda_1 \mathcal{L}_{\text{pixel}} + \lambda_2 \mathcal{L}_{\text{patch}} + \lambda_3 \mathcal{L}_{\text{instance}} \tag{4}$$

where λ_1–λ_3 are the scales to balance the three kinds of SSL tasks. In our article, the optimal values of these parameters are obtained via experimental studies, which are 0.4, 0.2, and 0.4, respectively.

B. Cooperative Classifier

Now, we introduce a CC to construct a regularization loss to improve the learned weights in the original classifier for novel classes, which is shown in Fig. 5(c). The original classifier in Faster R-CNN is only a one-layer fully-connected (FC) classifier, which is hard to meet the generalization requirement in FSL in Fig. 5(a). Thus, a cosine similarity metric is adopted to compute the similarity between the proposal feature and per-class weight in Fig. 5(b). Given an ith proposal with feature $\mathbf{p_i} \in \mathbb{R}^{\mathbf{c}}$, where c denotes the feature dimension, its classification score with the jth class weight $\mathbf{w_j} \in \mathbb{R}$ is computed as follows:

$$s_{ij} = \alpha \cos(\mathbf{p}_i, \mathbf{w}_j) \tag{5}$$

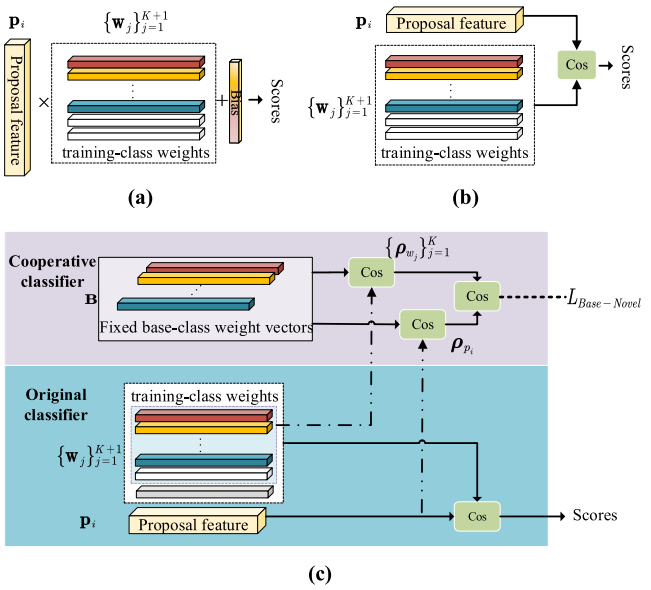


Fig. 5. **Illustration of the different classifiers in object detection.** (a) FC classifier consists of an FC layer. (b) Cosine similarity classifier computes the similarity between query proposal features $\mathbf{p}_i$ and the training weight vector $\{\mathbf{w}_j\}_{j=1}^{K}$. (c) Cosine similarity classifier with a CC. CC (top) first computes the correlation distributions of the ith proposal feature $\mathbf{p}_i$ and the training weights $\{\mathbf{w}_j\}_{j=1}^{K}$ of K classes from the primary cosine similarity classifier (bottom) on pretrained base-class weights $\mathbf{B}$, resulting in $\boldsymbol{\rho}_{p_i}$ and $\{\boldsymbol{\rho}_{w_j}\}_{j=1}^{K}$, respectively. Then, it calculates the cosine similarity between $\boldsymbol{\rho}_{p_i}$ and each vector in $\{\boldsymbol{\rho}_{w_j}\}_{j=1}^{K}$ to generate the classification score.

where s_{ij} is the classification score of the ith proposal on the jth class, $1 \le j \le K+1$, K is the number of categories, $\cos(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^{\top}\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$ is the cosine similarity function, and α is the scaling factor which is set to be 20 in our experiments. So, classification score of the ith proposal on all class weights is denoted as $\boldsymbol{S_i} = [s_{i1}, \ldots, s_{iK}, s_{i(K+1)}]$.

The cosine similarity classifier can reduce the variance of intraclass [11], and all the interclass variances share the same priority. By adopting the diagnosing tool [57] to analyze the false positives in novel classes, however, most of the false positives in novel categories are because of the misclassification between the base and novel objects, as shown in Fig. 5(b). Therefore, if a classifier can better distinguish base objects from novel objects by giving the relationship between them higher precedence, the detection of novel classes may increase.

To this end, we introduce a CC, paralleling the original classification branch, to exploit and constrain the relationship between the base and novel classes during FTP. The underlying assumption is that when proposals have identical labels [58], their correlation distributions on base classes should be similar. Therefore, the CC uses base classes as agents to calculate the transductive similarity between the proposal feature, $\mathbf{p}_i \in \mathbb{R}^c$, and class weights, $\{\mathbf{w}_j\}_{j=1}^{K}$. The classifier weights obtained in the first training stage are named as the reference base classes and denoted by $\mathbf{B} \in \mathbb{R}^{K_b \times c}$, where K_b is the number of base categories.

Concretely, in Fig. 5(c) (top), the CC first calculates the similarity distributions $\boldsymbol{\rho}_{p_i} \in \mathbb{R}^{K_b}$ and $\{\boldsymbol{\rho}_{w_j}\}_{j=1}^{K} \in \mathbb{R}^{K \times K_b}$ of the ith proposal $\mathbf{p}_i \in \mathbb{R}^c$ and weights $\{\mathbf{w}_j\}_{j=1}^{K} \in \mathbb{R}^{K \times c}$ on base classes $\mathbf{B}$. After that, the similarity between $\boldsymbol{\rho}_{p_i}$ and $\{\boldsymbol{\rho}_{w_j}\}_{j=1}^{K}$ can be computed as follows:

$$\begin{aligned} \boldsymbol{\rho}_{p_i} &= \mathrm{sim}(\mathbf{p}_i, \mathbf{B}) \\ \boldsymbol{\rho}_{w_j} &= \mathrm{sim}(\mathbf{w}_j, \mathbf{B}) \\ s_{ij}^{\mathrm{coop}} &= \alpha \cos(\boldsymbol{\rho}_{p_i}, \boldsymbol{\rho}_{w_j}) \end{aligned} \tag{6}$$

where $\mathrm{sim}(\cdot)$ is the similarity function that measures the cosine similarity between a vector and each row of a matrix, and s_{ij}^{coop} is the classification score to predict the ith proposal as the jth class from the CC.

After computing the cooperative classification scores of N RoI proposals, $\{\boldsymbol{S}_i^{\mathrm{coop}}\}_{i=1}^{N} \in \mathbb{R}^{N \times K}$, to provide extra supervisory signal and constrain the relationship between base and novel classes during FTP, we introduce a novel regularization $\mathcal{L}_{\mathrm{base\text{–}novel}}$ that is formulated as follows:

$$\mathcal{L}_{\mathrm{base\text{–}novel}} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{[y_i \neq 0]} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{S}_i^{\mathrm{coop}}, y_i) \tag{7}$$

where y_i is the category label of the ith proposal, $\mathbb{1}_{[y_i \neq 0]}$ means that the CC only focus on the foreground proposals, and $\mathcal{L}_{\mathrm{CE}}$ refers to the cross-entropy loss. Note that $\{\boldsymbol{S}_i^{\mathrm{coop}}\}_{i=1}^{N} \in \mathbb{R}^{N \times K}$ is not part of the output, and the classifier finally outputs the scores $\{\boldsymbol{S}_i\}_{i=1}^{N} \in \mathbb{R}^{N \times K}$ of the original classifier. It is different from consistency loss in Retentive R-CNN [69] which takes the form of KL-Divergence to incorporate the base–class knowledge into the training of novel classes to detect all categories efficiently. However, our regularization $\mathcal{L}_{\mathrm{base\text{–}novel}}$ improves the classification of novel classes by using base classes, which utilizes the relationship between base and novel classes explicitly.

In this CC, we do not introduce additional network parameters nor change the training process. But it considers the correlation distributions of novel class weights and proposal features on base classes. The CC computes classification scores of novel classes according to their similar characteristics. More specifically, the corresponding regularization $\mathcal{L}_{\mathrm{base\text{–}novel}}$ can constrain these similarities indirectly to reduce performance damage. Meanwhile, similar properties in data-abundant base classes can be exploited to train neurons more broadly, reducing the network susceptibility to specific neurons and alleviating overfitting. The effect of CC and $\mathcal{L}_{\mathrm{base\text{–}novel}}$ on reducing the false positives in novel classes can be visualized in Fig. 2(c). By comparing with Fig. 2(b), we can observe that CC can reduce the misclassification rate of base classes in novel classes from 32% to 29%.

C. Overall Loss

The overall loss functions for the two training phases are

$$\begin{aligned} \mathcal{L}^1 &= \mathcal{L}_{\mathrm{rpn}}^1 + \mathcal{L}_{\mathrm{cls}}^1 + \mathcal{L}_{\mathrm{loc}}^1 + \lambda_{\mathrm{ssl}}\mathcal{L}_{\mathrm{ssl}} \\ \mathcal{L}^2 &= \mathcal{L}_{\mathrm{cls}}^2 + \mathcal{L}_{\mathrm{loc}}^2 + \mathcal{L}_{\mathrm{base\text{–}novel}} \end{aligned} \tag{8}$$

where $\mathcal{L}^1$ and $\mathcal{L}^2$ are the losses for the first and second training phases, respectively. $\mathcal{L}_{\mathrm{rpn}}^1$ is the loss for the output from RPN to distinguish foregrounds from BGs and refine the anchors. $\mathcal{L}_{\mathrm{cls}}^i$ is a cross-entropy for the object classification and $\mathcal{L}_{\mathrm{loc}}^i$ is a smoothed L_1 loss for box Loc regression, $i = 1, 2$. λ_{ssl} is the scale to balance the self-supervised regularization $\mathcal{L}_{\mathrm{ssl}}$, which is set to be 0.1 in our experiments.

TABLE II

FEW-SHOT DETECTION PERFORMANCE (AP50) ON VOC2007 TEST SET IN NOVEL CLASSES UNDER THREE BASE/NOVEL SPLITS. **Bold** AND *Underline* INDICATE SOTA AND THE SECOND BEST. FS^{3}C ACHIEVES COMPARABLE PERFORMANCE AND OUTPERFORMS ALL THE BASELINES IN MORE ROBUST SHOT SETUP UNDER SPLIT 3

Method	w/G	Novel Set1					Novel Set2					Novel Set3				
		shot=1	2	3	5	10	shot=1	2	3	5	10	shot=1	2	3	5	10
LSTD [14]	✗	8.2	1.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3
FSRW [9]	✗	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
RepMet [10]	✗	26.1	32.9	34.4	38.6	41.3	17.2	22.1	23.4	28.3	35.8	27.5	231.1	31.5	34.4	37.2
Meta R-CNN [13]	✗	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
MPSR [12]	✗	41.7	42.5	51.4	55.2	*61.8*	24.4	29.3	39.2	39.9	47.8	35.6	41.8	42.3	48.0	49.7
FSCE [2]	✗	44.2	43.8	51.4	61.9	63.4	27.3	29.5	43.5	44.2	50.2	37.2	41.9	47.5	54.6	*58.5*
TFA w/fc [11]	✗	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 [11]	✗	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
DeFRCN [66]	✗	*53.6*	*57.5*	*61.5*	*64.1*	60.8	*30.1*	*38.1*	*47.0*	**53.3**	**47.9**	*48.4*	*50.9*	*52.3*	*54.9*	57.4
FS^{3}C (TFA w/cos)	✗	44.1	46.5	51.6	57.8	61.5	27.6	29.5	40.9	40.0	46.8	40.4	44.5	46.0	52.9	55.6
FS^{3}C (DeFRCN)	✗	**56.0**	**57.8**	**62.0**	**64.7**	**61.9**	**32.5**	**40.9**	**48.3**	*53.2*	*47.6*	**49.4**	**52.7**	**52.7**	**56.2**	**59.7**
FSDetView [39]	✓	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
FSCE [2]	✓	32.9	44.0	46.8	52.9	59.7	23.7	30.6	38.4	43.0	48.5	22.6	33.4	39.5	47.3	54.0
Retentive R-CNN [69]	✓	*42.4*	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
TIP [70]	✓	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
TFA w/fc [11]	✓	22.9	34.5	40.4	46.7	52.0	16.9	26.4	30.5	34.6	39.7	15.7	27.2	34.7	40.8	44.6
TFA w/cos [11]	✓	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
DeFRCN [66]	✓	40.2	*53.6*	*58.2*	*63.6*	*66.5*	*29.5*	*39.7*	*43.4*	*48.1*	*52.8*	*35.0*	*38.3*	*52.9*	*57.7*	*60.8*
FS^{3}C (TFA w/cos)	✓	28.5	38.4	43.4	49.2	54.6	17.9	28.1	31.5	37.0	43.9	18.5	28.1	37.8	41.7	46.9
FS^{3}C (DeFRCN)	✓	**44.5**	**57.2**	**61.6**	**66.1**	**67.5**	**31.9**	**43.1**	**47.4**	**50.9**	**54.0**	**38.2**	**50.6**	**55.2**	**59.4**	**61.9**

TABLE III

FEW-SHOT DETECTION PERFORMANCE ON COCO FOR NOVEL CLASSES. **Bold** AND *Underline* INDICATE SOTA AND THE SECOND-BEST. *MODEL IS REEVALUATED USING THE STANDARD PROCEDURE FOR A FAIR COMPARISON

Method	w/G	Backbone	novel AP		novel AP50		novel AP75	
			shot=10	30	shot=10	30	shot=10	30
LSTD [14]	✗	VGG-16	-	-	3.2	6.7	-	-
FSRW [9]	✗	YOLO V2	5.6	9.1	12.3	19.0	4.6	7.6
FSOD-Att [15]	✗	FRCN-R50	14.3	-	*28.3*	-	13.1	-
FSDetView* [39]	✗	FRCN-R50	10.7	14.7	24.9	*30.6*	6.7	12.2
DAnA-FasterRCNN [67]	✗	FRCN-R50	**18.6**	21.6	-	-	*17.2*	*20.3*
Meta R-CNN [13]	✗	FRCN-R101	8.7	12.4	19.1	25.3	6.6	10.8
MPSR [12]	✗	FRCN-R101	9.8	14.1	17.9	25.4	9.7	14.2
FSCE [2]	✗	FRCN-R101	11.9	16.4	-	-	10.5	16.2
TFA w/fc [11]	✗	FRCN-R101	10.0	13.4	19.2	24.7	9.2	13.2
TFA w/cos [11]	✗	FRCN-R101	10.0	13.7	19.1	24.9	9.3	13.4
DeFRCN [66]	✗	FRCN-R101	*18.5*	*22.6*		-	-	-
FS^{3}C (TFA w/cos)	✗	FRCN-R101	11.0	15.1	23.6	28.9	10.0	14.9
FS^{3}C (DeFRCN)	✗	FRCN-R101	*18.5*	**22.7**	**33.9**	**39.8**	**17.5**	**23.0**
FSDetView [39]	✓	FRCN-R50	12.5	14.7	27.3	*30.6*	9.8	12.2
Retentive R-CNN [69]	✓	FRCN-R101	10.5	13.8	-	-	-	-
TIP [70]	✓	FRCN-R101	16.3	18.3	**33.2**	35.9	*14.1*	*16.*
FSCE [2]	✓	FRCN-R101	11.1	15.3	-	-	9.8	14.2
TFA w/fc [11]	✓	FRCN-R101	9.1	12.0	17.3	22.2	8.5	11.8
TFA w/cos [11]	✓	FRCN-R101	9.1	12.1	17.1	22.0	8.8	12.0
DeFRCN [66]	✓	FRCN-R101	*16.8*	*21.2*	-	-	-	-
FS^{3}C (TFA w/cos)	✓	FRCN-R101	10.2	13.7	19.2	24.9	9.9	13.6
FS^{3}C (DeFRCN)	✓	FRCN-R101	**17.4**	**22.0**	*30.3*	**36.7**	**17.6**	**21.4**

V. EXPERIMENTS AND ANALYSIS

A. Datasets and Settings

1) Datasets: To be consistent with previous works [9], [11], [12], [13], [15], [39], [60], we evaluate our framework on PASCAL VOC 2007+2012 and MS-COCO, respectively. For a fair comparison, we use the same train/test splits and evaluation protocol provided in [9]. For the PASCAL VOC dataset, we randomly choose five out of 20 categories as novel classes with $K \in \{1, 2, 3, 5, 10\}$, and the remaining 15 categories are taken as the base classes. COCO is a more challenging object detection dataset, which contains 80 categories, including those 20 categories in Pascal VOC. Similarly, on the MS-COCO dataset, we set 60 out of 80 object categories disjoint with PASCAL VOC as base classes, and the remaining 20 categories are as novel ones with $K \in \{10, 30\}$. To verify the generalization ability of our FS^{3}C in cross-domain situations, we also consider training our model on

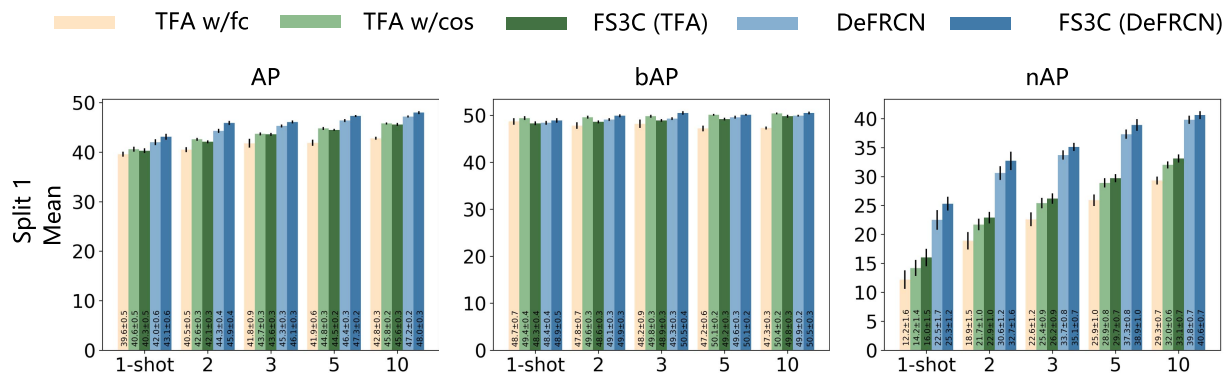


Fig. 6. Generalized object detection benchmarks on VOC.

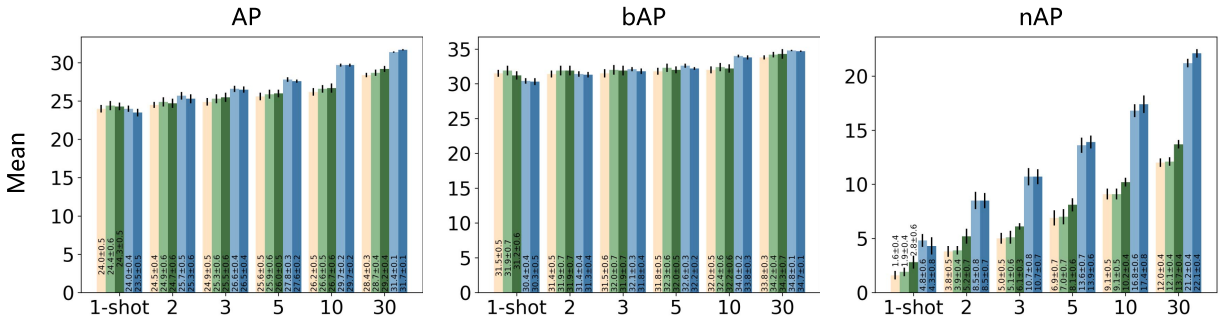


Fig. 7. Generalized object detection benchmarks on COCO.

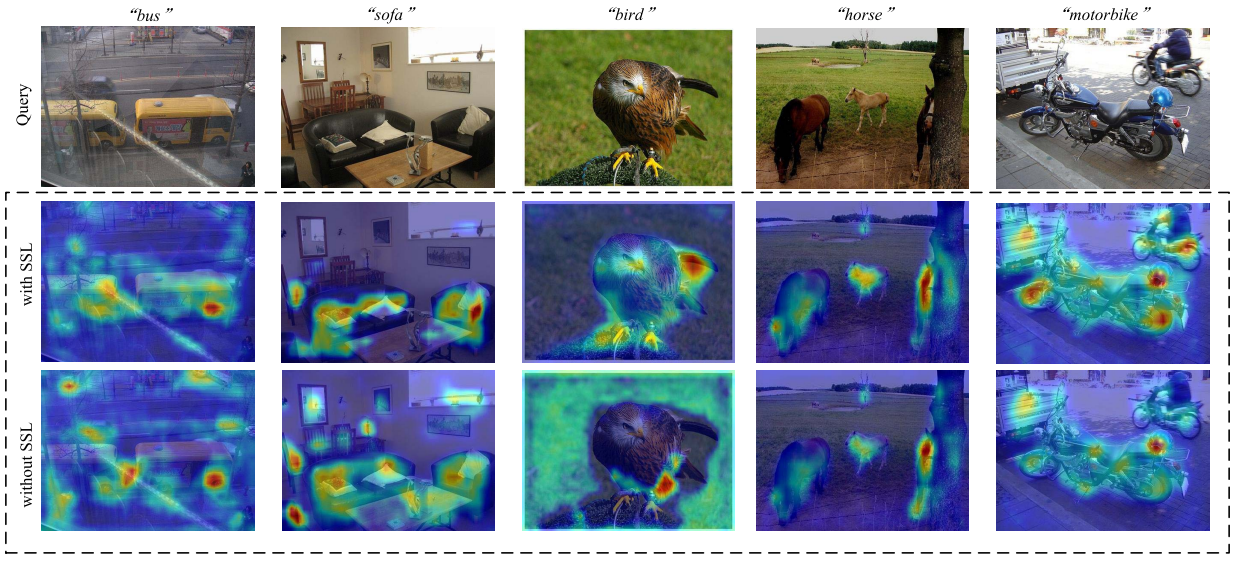


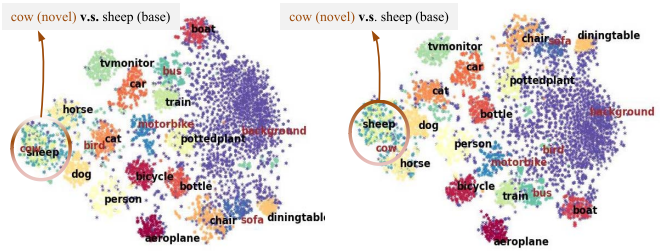
Fig. 8. Some visualizations of Grad-Cam++ activation of query images on the VOC2007 test set in novel classes of the first base/novel split. Compared with TFA, FS^{3}C pays more attention to the objects.

60 base-class images from MS COCO and 20 novel-class images from PASCAL VOC, which is denoted as *COCO* to *PASCAL*.

2) Evaluation: There are two popular evaluation protocols in FSOD: FSOD and generalized FSOD (G-FSOD). The former only focuses on performance in novel classes. In contrast, the latter reports bAP and the overall AP and nAP, which allows us to observe the network's overall performance.

3) Implementation Details: To verify our framework on FSOD, we employ Faster R-CNN [6] as our base object detector, where the feature extractor is set to be ResNet-101 [5] with a Feature Pyramid Network [61], which is pretrained on ImageNet [62]. Of course, other object detectors and feature extractors can replace these modules easily.

Next, our model is trained end-to-end on a DGX1 server with eight Tesla V100 GPUs. During the training, we adopt an



(a) (b)

Fig. 9. t-SNE visualization of class vectors and proposal features with respect to FS^{3}C trained with/without cooperating loss. The dot means proposal feature, and the word indicates the category and the corresponding weight vector (**brown** and **black** fonts indicate novel and base classes, respectively). (a) Without and (b) with the base–novel loss.

TABLE IV

mAP COMPARISON OF NOVEL/BASE CLASSES ON VOC SPLIT 1. **Bold** AND *Underline* INDICATE SOTA AND THE SECOND-BEST. WE MAINTAIN STRONG PERFORMANCE ON BASE CLASSES WHILE ACHIEVING COMPETITIVE PERFORMANCE ON NOVEL CLASSES

Method	base AP50		novel AP50	
	shot=3	shot=10	shot=3	shot=10
Meta R-CNN [13]	64.8	67.9	35.0	51.5
MPSR [12]	67.8	71.8	*51.4*	**61.8**
TFA w/cos [11]	**79.1**	**78.4**	44.7	56.0
FS^{3}C (TFA w/cos)	*78.9*	*78.1*	**51.6**	*61.5*

SGD optimizer with a batch size of 16, momentum of 0.9, and weight decay of 0.0001. The learning rate is 0.02 for the first training stage. Then, we scale the training steps when training the number of shots for the fine-tuning stage. We train models over ten random samples of training shots to obtain averages and confidence intervals, which are the same as TFA [11].

4) Baselines: We compare our model with 13 competitive FSOD methods, which are LSTD [14], RepMet [10], Meta R-CNN [13], FSRW [9], TFA [11], FSOD-Att [15], MPSR [12], FSDetView [39], DeFRCN [66], FSCE [2], Retentive R-CNN [69], DAnA-FasterRCNN [67], and TIP [70]. Since TFA [11] can integrate different classifiers, we denote them as TFA w/fc and TFA w/cos, respectively. LSTD [14], RepMet [10], and FSRW [11] which are built upon VGG [63], Inception V3 [64], and YOLOv2 [38], respectively. All other methods are built upon Faster R-CNN [6]. Note that these methods made modifications to address different facets of FSOD, but most of them can easily be integrated into our framework to further improve performance. We build FS^{3}C (TFA w/cos) and FS^{3}C (DeFRCN) on the classic fine-tuning method TFA [11] and SOTA fine-tuning method DeFRCN [66], respectively, to prove the effectiveness.

B. Quantitative Analysis

1) Comparison With the SOTA: We compare our approach with the existing competitive FSOD methods, where we provide the average AP50 of the novel classes on PASCAL VOC with three splits in Table II, as well as the average AP/AP50/AP75 of the novel classes on COCO in Table III.

In Table II, we can observe that our approach FS^{3}C (DeFRCN) achieves most of the best results compared to other SOTA methods in both FSOD and G-FSOD settings. Moreover, even built on a weaker TFA model, our approach FS^{3}C (TFA w/cos) can achieve competitive performance, especially in Novel Set3. We can observe that our approach FS^{3}C (TFA w/cos) can achieve comprehensive and distinct improvements over TFA in more few-shot settings. Specifically, the improvements over the two-shot setting in splits 1 and 3 are +10.4% and +10.3%, respectively. In Table III, we re-evaluate FSDetView with the same train/test splits as us for a fair comparison. We can observe that our model produces a substantial improvement in AP and AP75 over SOTA methods on novel classes. Furthermore, we also notice that FS^{3}C consistently outperforms TFA and DeFRCN across different shots in all novel AP settings.

Moreover, we conduct cross-dataset experiments on the VOC 2007 test set using ten-shot images per class. In this setting, all models are trained on the base classes from the COCO dataset and fine-tuned using a ten-shot image per class from PASCAL. The performance of PASCAL novel classes is worse than that of base classes in the PASCAL dataset due to the large domain shift. The mAP of FSRW [9], Meta R-CNN [13], MPSR [12], and DeFRCN [66] are 32.29%, 37.4%, 42.3%, and 55.9%, respectively, while our method FS^{3}C (DeFRCN) achieves 56.3%, which highlights the cross-domain generalization ability of our model. Meanwhile, FS^{3}C (TFA w/cos) also achieves a competitive mAP of 44.6%.

2) Performances on Base Classes: To analyze the different performances on base and novel classes in the FSOD setting, we report the results of three/ten-shot detection on the first split of the VOC dataset in Table IV. We can observe that our approach has a much higher average AP50 on the base classes than MPSR [12] with a gap of about ten points while achieving competitive performance on novel classes. In the G-FSOD setting, following [11], we further adopt 30 and ten repeated runs with different randomly sampled training shots for PASCAL VOC and COCO, respectively, and show averages and confidence intervals in Figs. 6 and 7. Interestingly, our method improves the average novel-class accuracies of TFA and DeFRCN in all the settings, with small confidence intervals, and keeps the good performance over base classes.

3) Ablation Studies: To validate the impact of three kinds of SSL and CC in FS^{3}C, we provide an ablation study on three/ten-shot object detection with PASCAL VOC in the first novel split setup, as shown in Table V. We can observe that SSL and the CC can boost our double head baseline of TFA by roughly 1–3 points on novel classes. Since DeFRCN has a robust classifier from the pretrained model, our CC plays a small role in improving its performance. For SSL, pixel-level contrastive learning performs the best on both three- and ten-shot settings. Although the patch-level task simulates the more complex structural information, its performance is not as good as others. Because the complex *Jigsaw puzzle* task may lead to features that are suitable to solve the pretext task, while not the detection task. By combining all SSL tasks, the detector can achieve better performance. Finally, no matter whether built on TFA or DeFRCN, the best performance on novel classes is achieved by combining SSL and CC (FS^{3}C).

Fig. 10. Visualizations of the prediction by the proposed FS^{3}C. (a) and (c), respectively, refer to TFA and DeFRCN, correspondingly, (b) and (d) represent FS^{3}C (TFA w/cos) and FS^{3}C (DeFRCN). Compared with FS^{3}C, TFA and DeFRCN are inferior: mislocalizing the motorbike in the first column; misclassifying or missing detection horse in the middle column; missing detection of bus, sofa, and bird or the lower confidence in the last three columns.

TABLE V
ABLATION STUDY ON THE VOC2007 TEST SET FOR NOVEL CLASSES OF THE FIRST SPLIT. **Bold** INDICATES SOTA

SSL			Base-Novel loss (Cooperative Classifier)	nAP50 (TFA)		nAP50 (DeFRCN)	
pixel-level	patch-level	instance-level		shot=3	shot=10	shot=3	shot=10
				47.8	58.3	61.5	60.8
√				49.0 (+1.2%)	59.9 (+1.6%)	61.8 (+0.3%)	61.4 (+0.6%)
	√			48.0 (+0.2%)	58.7 (+0.4%)	61.7 (+0.2%)	61.3 (+0.5%)
		√		48.5 (+0.7%)	59.2 (+0.9%)	61.8 (+0.3%)	61.5 (+0.7%)
√	√	√		49.5 (+1.7%)	60.4 (+2.1%)	62.0 (+0.5%)	61.7 (+0.9%)
			√	49.3 (+1.5%)	59.5 (+1.2%)	61.6 (+0.1%)	61.0 (+0.2%)
√	√	√	√	**51.6 (+3.8%)**	**61.5 (+3.2%)**	**62.0 (+0.5%)**	**61.9 (+1.1%)**

C. Qualitative Analysis

1) Self-Supervised Regularization: To gain deep insights into SSL, we compare our model (with SSL only) with TFA on novel-class objects with the aid of Grad-CAM++ [65], which is shown in Fig. 8. We can observe that SSL not only diverts the module's attention to object rather than noise BG, for example, the first three columns, but also boosts the feature response on the correct class of objects, for example, the last two columns. This observation verifies our conjecture that SSL can strengthen semantic information, reducing the impact of inessential features by introducing structural information.

2) Base–Novel Regularization: In Table V, we already notice that with the base–novel regularization, that is, with cooperative loss, the detection performance on the novel class does improve. Here, we would like to investigate whether the performance improvement is due to the better separation of base objects from novel objects or not. For better comparison, we visualize the proposal features with high similarity scores computed with class weight vectors. Fig. 9 shows the t-SNE visualization of these proposal features that are trained with or without cooperative loss in a three-shot setting of VOC split 1. We can observe that proposal features from "sheep" (base) and "cow" (novel), "cat" and "bird" are almost overlapping in Fig. 9(a), which is trained without base–novel regularization. This observation validates the false-positive analysis in Fig. 9(b). Once we add the base–novel regularization, we can observe that these overlaps are reduced by a big margin. It demonstrates that our CC and base–novel regularization can improve the original classifier by pushing novel classes away from base classes in the semantic space.

3) Detection Results: We provide a qualitative visualization of the detected objects in a ten-shot setting in Fig. 10, which are the results from TFA w/cos, DeFRCN, and our model (FS^{3}C), respectively. We observe that FS^{3}C can detect novel-class objects more sensitively and predict high-quality boxes, especially in complex BGs. On the other hand, we also

notice some failure cases, including misclassifying novel objects as similar base objects, and missing detection, for example, column 4 in Fig. 10(b). The possible reason behind the failures can be the tiny size of the object and the ambiguous structure.

VI. Conclusion

This article studies the problem of missing structural and semantic information and misclassification between base and novel classes in fine-tuning-based FSOD methods. To deal with these two issues, we first unify SSL into an object detection framework as an auxiliary task to strengthen the structural information. Then, we branch the primary classification head with a CC to introduce a new classification objective to further exploit the relationship between base and novel classes, thus improving the classification performance. Finally, extensive experiments on PASCAL VOC and COCO datasets demonstrate the effectiveness of our model.

References

[1] B. Li, B. Yang, C. Liu, F. Liu, R. Ji, and Q. Ye, "Beyond max-margin: Class margin equilibrium for few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7363–7372.

[2] B. Sun, B. Li, S. Cai, Y. Yuan, and C. Zhang, "FSCE: Few-shot object detection via contrastive proposal encoding," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7352–7362.

[3] C. Zhu, F. Chen, U. Ahmed, Z. Shen, and M. Savvides, "Semantic relation reasoning for shot-stable few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8782–8791.

[4] Y. Wu *et al.*, "Rethinking classification and localization for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10186–10195.

[5] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[7] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. ICRL*, Toulon, France, 2017.

[8] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.

[9] B. Kang, Z. Liu, X. Wang, F. Yu, J. Feng, and T. Darrell, "Few-shot object detection via feature reweighting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2020, pp. 8420–8429.

[10] L. Karlinsky *et al.*, "RepMet: Representative-based metric learning for classification and few-shot object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 5197–5206.

[11] X. Wang, T. E. Huang, T. Darrell, J. E. Gonzalez, and F. Yu, "Frustratingly simple few-shot object detection," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 9919–9928.

[12] J. Wu, S. Liu, D. Huang, and Y. Wang, "Multi-scale positive sample refinement for few-shot object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Glasgow, U.K.: Springer, Aug. 2020, pp. 456–472.

[13] 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 *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9577–9586.

[14] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A low-shot transfer detector for object detection," 2018, *arXiv:1803.01529*.

[15] Q. Fan, W. Zhuo, C.-K. Tang, and Y.-W. Tai, "Few-shot object detection with attention-RPN and multi-relation detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 4013–4022.

[16] Z. Yue, H. Zhang, Q. Sun, and X.-S. Hua, "Interventional few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1–13.

[17] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn. (ICML)*, 2020, pp. 1597–1607.

[18] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. 14th Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 69–84.

[19] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018.

[20] M. Everingham et al., "The PASCAL visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[21] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zürich, Switzerland: Springer, Sep. 2014, pp. 740–755.

[22] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum, "One-shot learning by inverting a compositional causal process," in *Proc. 26th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Lake Tahoe, NV, USA, 2013, pp. 2526–2534.

[23] H. Wang *et al.*, "CosFace: Large margin cosine loss for deep face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5265–5274.

[24] Y. Sun, *Deep Learning Face Representation by Joint Identification-Verification*. Hong Kong: The Chinese University of Hong Kong (Hong Kong), 2015.

[25] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. Frank Wang, and J.-B. Huang, "A closer look at few-shot classification," 2019, *arXiv:1904.04232*.

[26] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4690–4699.

[27] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," 2017, *arXiv:1703.05175*.

[28] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," 2017, *arXiv:1711.04043*.

[29] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," 2016, *arXiv:1606.04080*.

[30] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[31] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, and L. Wang, "Boosting few-shot learning with adaptive margin loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 12576–12584.

[32] B. Liu *et al.*, "Negative margin matters: Understanding margin in few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 438–455.

[33] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7278–7286.

[34] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Int. Conf. Learn. Represent. (ICLR)*, 2016, pp. 1–11.

[35] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[36] R. Hou, H. Chang, B. Ma, S. Shan, and X. Chen, "Cross attention network for few-shot classification," 2019, *arXiv:1910.07677*.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1097–1105.

[38] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7263–7271.

[39] Y. Xiao, V. Lepetit, and R. Marlet, "Few-shot object detection and viewpoint estimation for objects in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3090–3106, Mar. 2023, doi: 10.1109/TPAMI.2022.3174072.

[40] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 1422–1430.

[41] G. Larsson, M. Maire, and G. Shakhnarovich, "Learning representations for automatic colorization," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 577–593.

[42] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 649–666.

[43] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15535–15545.
[44] P. Bojanowski and A. Joulin, "Unsupervised learning by predicting noise," 2017, *arXiv:1704.05310*.
[45] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*.
[46] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 766–774.
[47] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[48] O. J. Hénaff *et al.*, "Data-efficient image recognition with contrastive predictive coding," 2019, *arXiv:1905.09272*.
[49] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," 2018, *arXiv:1808.06670*.
[50] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6707–6717.
[51] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.
[52] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.
[53] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2536–2544.
[54] C. Doersch and A. Zisserman, "Multi-task self-supervised visual learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2051–2060.
[55] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, "Scaling and benchmarking self-supervised visual representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6391–6400.
[56] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1920–1929.
[57] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 340–353.
[58] Z. Wang, Y. Zhao, J. Li, and Y. Tian, "Cooperative bi-path metric for few-shot learning," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1524–1532.
[59] Anonymous, "Cooperating rpn's improve few-shot object detection," in *Proc. Int. Conf. Learn. Represent.*, 2021. [Online]. Available: https://openreview.net/forum?id=in2qzBZ-Vwr
[60] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, and T. Xiang, "Incremental few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 13846–13855.
[61] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.
[62] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[63] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[64] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2016, pp. 2818–2826.
[65] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.
[66] L. Qiao, Y. Zhao, Z. Li, X. Qiu, J. Wu, and C. Zhang, "DeFRCN: Decoupled faster R-CNN for few-shot object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 8681–8690.
[67] T.-I. Chen *et al.*, "Dual-awareness attention for few-shot object detection," *IEEE Trans. Multimedia*, early access, Nov. 4, 2021, doi: 10.1109/TMM.2021.3125195.
[68] T. Xiao, C. J. Reed, X. Wang, K. Keutzer, and T. Darrell, "Region similarity representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 10539–10548.
[69] Z. Fan, Y. Ma, Z. Li, and J. Sun, "Generalized few-shot object detection without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4527–4536.
[70] A. Li and Z. Li, "Transformation invariant few-shot object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3094–3102.