Few-Shot End-to-End Object Detection via Constantly Concentrated Encoding across Heads

Jiawei Ma, Guangxing Han, Shiyuan Huang,
Yuncong Yang, and Shih-Fu Chang

Columbia University, New York NY 10027, USA
{jiawei.m, gh2561, shiyuan.h, yy3035, sc250}@columbia.edu

Abstract. Few-shot object detection (FSOD) aims to detect objects of new classes and learn effective models without exhaustive annotation. The end-to-end detection framework has been proposed to generate sparse proposals and set a stack of detection heads to improve the performance. For each proposal, the predictions at lower heads are fed into deeper heads. However, the deeper head may not concentrate on the detected objects and then degrades, resulting in inefficient training and further limiting the performance gain in few-shot scenario. In this paper, we propose a few-shot adaptation strategy, Constantly Concentrated Encoding across heads (CoCo-RCNN), for the end-to-end detectors. For each class, we gather the encodings which detect on its object instances and then train them to be discriminative to avoid degraded prediction. In addition, we embed the class-relevant encodings to the learnable proposals to facilitate the adaptation at lower heads. Extensive experimental results show that our model brought clear gain on benchmarks. Detailed ablation studies are provided to justify the selection of each component.

Keywords: End-to-end detector, constantly concentrated encoding.

1 Introduction

Deep convolution neural networks have achieved impressive successes in general object detection. Learning a deep detector typically requires sufficient annotated training instances, and the detection performance is far from satisfactory when the annotated samples are extremely limited. As such, few-shot object detection has been studied to mimic human vision system which has remarkable ability to learn the object visual appearance for new (*novel*) classes with a few instances.

Recently, the end-to-end framework [1,42,32] has been proposed for object detection. Different from the conventional methods [28,29,12,14,13] which generate dense proposals from the anchor boxes, the end-to-end framework sets a few learnable proposal vectors to generate sparse proposals for each image dynamically. Each vector is learned as part of the model parameters and serves as a proposal encoding to predict an object encoding in one detection head. Then, similar to the Faster-RCNN [29], a detection module is set for classification and bounding box regression while the prediction from different proposals are supposed to be diverse. To improve the performance, a small number of heads are

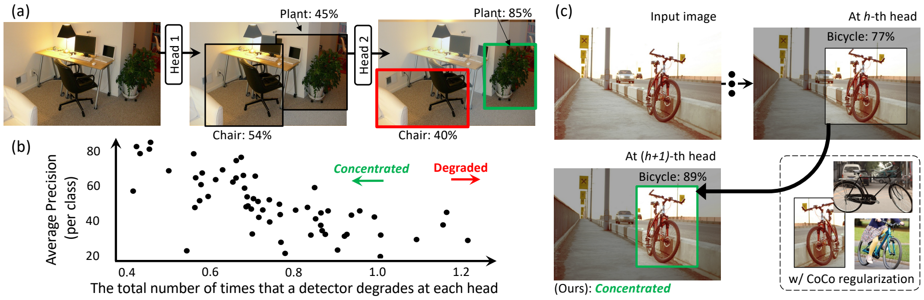

Fig. 1. (a) Within an end-to-end object detector, the prediction from a lower head can be fed into a deeper head. For each proposal, the prediction can be improved (green) when the detector concentrates on the object that has been detected or degraded (red) when it is distracted by other patterns . (b) Comparison between the detection precision and the total number of times that an object detector degrades at each head. For each class, the precision can be high when the detector can keep concentrating on the previously detected object instances and improve the performance (more details can be found in Supp.). (c) Comparing with conventional finetuning baseline, we add constantly concentrated encoding regularization to make the detector concentrated.

stacked. As shown in Fig. 1(a, green), the predictions from lower heads are fed into deeper heads for refinement. In this way, all heads can make predictions and the detection from deeper heads are closer to the groundtruth on average [1,32], *i.e.*, more accurate classification and higher intersection over union (IoU).

Though the detection precision of a deeper head is generally higher, for each proposal, as shown in Fig. 1(a, red), the prediction can still *degrade*. For example, given a proposal, the detection at lower heads may be close to one object in image, the prediction can then be *distracted* by other patterns in deeper heads. Meanwhile, by breaking down the detection scores for each class separately, as shown in Fig. 1(b), when the detector can keep *concentrating* on the objects in its input at each head, the detection can be refined constantly and the final detection precision is high. As such, even if the end-to-end object detector has achieved superior performance under large-scale training, adapting it for few-shot novel classes is still challenging, as it is hard to maintain the detector constantly concentrate on instances of novel classes in the data-hunger scenario.

As an end-to-end object detector makes sparse proposals to detect all object instances over the full image, at each head, the object encodings are supposed to be different from each other to avoid similar/overlapping predictions. Recent work has observed that an adapted object detector can properly localize novel instances [2,31], while a discriminative object encoding is important for strengthening the detection results [32,42]. Then, at each head, to make the model improve its input, *i.e.*, the detection at the previous head, it is important to make the detector concentrate on the class-relevant components and avoid being distracted. Thus, for each proposal, the object encodings from all heads are supposed to be discriminative such that the classification is accurate and consistent and the detection at each head can be refined continuously.

In this paper, as shown in Fig. 1(c), we propose CoCo-RCNN, a simple yet effective strategy for few-shot adaptation, for the end-to-end object detectors. We design the Constantly Concentrated Encoding (CoCo) regularization based on the supervised contrastive learning [20], aiming to make the object encodings discriminative and have high similarity with the groundtruth encodings of the same class. Different from the conventional supervised contrastive learning which performs augmentation through manipulation on the pixels, within the end-to-end object detector where multiple heads are stacked, the object encodings from different heads can be treated as the augmentation at feature-level. We use Sparse-RCNN as our baseline and first pre-train it on the classes with abundant annotated samples (*base*). Then, we adapt the base detector to novel classes by finetuning on only a few examples as well as minimizing the CoCo loss and detection losses. Meanwhile, as the proposal vectors are class-irrelevant and the model is difficult to concentrate on objects at lower heads, we also embed class-relevant information by adding each of the class encodings on a sub-group of proposal vectors. The contributions of this paper are as follows:

- We propose CoCo-RCNN, a few-shot adaptation strategy for end-to-end object detectors. At each head, for each proposal, the model is trained to concentrate on the object detected at previous head when the training data is limited.
- We design the constantly concentrated encoding loss, incorporating the supervised contrastive loss to make the object encodings discriminative. To encourage the detector to concentrate on object instances at lower heads, we additionally embed class-relevant information to the learnable proposals.
- We use Sparse-RCNN as a baseline model, and show that our CoCo-RCNN consistently achieves performance gain on PASCAL VOC and MSCOCO. We also provide comprehensive ablation studies to justify the design of each component and demonstrate its effect in large-scale training.

2 Related Work

Object detection methods with dense proposals, have been widely used in many related tasks and the most representative method is Faster-RCNN [29]. Given the feature maps of a full image, a detector first uses the region proposal network (RPN) [29] to generate dense proposals ($\sim 10^5$). Each proposal is paired with an objectness score to indicate the possibility for the existence of objects. Then, the proposals ($\sim 1,000$) with high objectness scores are kept and used to extract object encodings from the original feature maps through RoI pooling [29]. Finally, the object encodings are used for detection, *i.e.*, classification and bounding box regression. As such, these methods are all termed two-stage detectors. In practice, each proposal is predicted w.r.t. an anchor box while each anchor box is determined by the spatial position, size, and aspect ratio. Thus, a large number of anchor boxes are manually defined to densely cover the full image, resulting in heavy computation. To improve the detection speed and training efficiency, methods such as YoLo [28] and SSD [26] have been proposed to directly predict the class and location of objects from the image feature maps in a single stage. However, all of the methods mentioned above need to generate dense candidates ($\geq 1,000$). Thus, the non-maximum suppression (NMS) [7] is required to fuse the detection results and obtain clean & sparse predictions.

End-to-end object detection methods, in contrast, set a few learnable proposals. Each proposal is represented as a vector and learned as part of the model parameters. The representative methods include Detr [1], Deformable Detr [42], and Sparse-RCNN [32]. Within a detection head, each proposal vector, serving as a proposal encoding, is used to make one prediction. Typically, a correlation module is set to connect each proposal encoding with the image feature maps and extract an object encoding. The correlation modules include cross-attention [34] and dynamic instance interaction [32]. The cross-attention module flattens the feature maps into a set of vision encodings and measures the affinity scores with the proposal encodings pair-wisely. For each proposal encoding, the visual encodings with high attention scores are kept in the corresponding object encoding. For Sparse-RCNN, instead, each proposal vector is paired with a learnable bounding box (part of model parameters). Then, the dynamic instance interaction will perform RoI pooling on the feature maps using the paired bounding box and connect the pooled feature with the proposal encoding to predict an object encoding. In this way, each proposal encoding is only compared with the feature maps of a sub-region. As a result, Sparse-RCNN is more efficient than other methods, *e.g.*, variants of Detr [1]. For the sake of training efficiency and low computational workload, we choose Sparse-RCNN [32] as our baseline.

The end-to-end object detectors are trained to generate sparse predictions such that manual intervention including NMS is no longer needed. The predictions by various proposals are supposed to be different such that all objects appearing in the image can still be detected. To improve the detection precision, multiple detection heads are stacked & cascaded where the predictions at lower heads are used as inputs at deeper heads. To balance the computational workload and performance, the number of heads is usually set as six for a fair comparison. For Sparse-RCNN, the object encodings and bounding boxes predicted by the current head are used as inputs for the next head. For the convenience of description, we omit description for bounding boxes but just mention that the object encodings are reused as proposal encodings in the stacked heads.

Few-shot object detection (FSOD) learns to detect objects of novel classes by only training on a few annotated instances (support). Different from the few-shot classification which can directly compare the global image features [30,33,27,41,18], FSOD is additionally supposed to localize the objects in images and distinguish the objects from the background. The methods for FSOD are mostly developed on the framework with dense proposals and can be roughly categorized into meta-learning-based and finetuning-based.

The *meta-learning-based* methods aim to learn a class-agnostic meta-learner and improve the detection performance by learning to align the support samples with the objects in testing images [9]. As the few support instances may be of various viewpoints or shapes, how to effectively extract the discriminative components and align the support with objects in test images is important. First, given the support samples, re-weighting the image feature maps is an effective strategy [19]. Then, FewX [6] obtains an attention-based meta-learner for RPN such that the class-relevant proposals are generated for further detection. The following works dig into this problem and propose attentive feature alignment module [9], query-adaptive heterogeneous graph convolution [8] and fully cross-transformer [10] to improve the performance. Meanwhile, *Han et al.* [11] propose to exploit class semantic information to assist in FSDO. The meta-learning-based method is a promising solution for transferring meta-knowledge from base classes to novel classes, and has shown its strength in extremely few-shot cases (*e.g.*, 1-shot) on challenging datasets (*e.g.*, MS COCO [25]).

The *finetuning-based* methods first obtain an initialization by pre-training the object detector with sufficient *base* samples and then finetune the the model a few support samples for *novel* classes. In this way, the finetuning-based methods aim to adapt a pre-trained model to novel classes efficiently, which has drawn increasing attention thanks to its simplicity. Recently, TFA [35] has shown that finetuning on a few data is a strong baseline. Then, by learning to detect objects from multiple scales, MPSR [39] has improved the performance further. In addition, FSCE [31] builds upon TFA and improves the detection performance by learning to obtain discriminative object encodings for FSOD. Different from the previous methods, the end-to-end object detector applies multiple heads to refine the detection progressively where our focus is to keep the detector concentrating on the detected objects during the refinement process to learn the adapted model efficiently.

3 Preliminary

3.1 Learning-Task Formulation

In FSOD, we are first given a *base* dataset $\mathcal{D}_{base}$, including abundant amount of annotated object instances from *base* classes $\mathcal{C}_{base}$. For each instance, the annotation consists of a class label c, and a bounding box (bbox) $u = (x, y, w, h)$ in the image. An image may contain multiple (N_T) instances from different classes, *i.e.*, $\mathcal{T} = \{(c_t, u_t)\}_{t=1}^{N_T}$. Then, we are given a *novel* set $\mathcal{D}_{novel}$ and the instances are from the novel classes $\mathcal{C}_{novel}$.

For an N_C-way K-shot FSOD task, there are N_C novel classes $|\mathcal{C}_{novel}| = N_C$ and each class has K annotated instances. The class sets for *base* and *novel* are disjoint, *i.e.*, $\mathcal{C}_{base} \cap \mathcal{C}_{novel} = \varnothing$. Following most finetuning-based methods [35,31], we first pre-train our object detector on $\mathcal{D}_{base}$ to obtain a base model and then finetune the model on $\mathcal{D}_{novel}$ for adaptation. Finally, we evaluate the adapted model on a test set.

3.2 End-to-End Object Detection

Conventional object detectors predict dense proposals w.r.t. each anchor box and filter out the proposals with low objectness scores. In contrast, the end-to-end framework sets a few learnable proposals to generate sparse predictions. As illustrated in Fig. 2, a proposal is represented as a proposal vector and learned as part of model parameters. Given the image feature maps, a proposal vector serves as a proposal encoding and is used to generate one object encoding for the image within one head, which is further used for detection. To improve the performance, multiple (N_H) heads are stacked and learned jointly.

In detail, at the h-th head where $h \in \{1...N_H\}$, a d-dim proposal encoding $\mathbf{p}_n^h \in \mathcal{R}^d$ is fed into a correlation module $f_a^h(\cdot,\cdot)$ to generate an object encoding $\mathbf{o}_n^h = f_a^h(\mathbf{p}_n^h, f_f(I)) \in \mathcal{R}^d$. The feature maps for image I is extracted by $f_f(\cdot)$ and the $n \in \{1...N_P\}$ indexes the encodings. The object encoding is then used for classification and bbox regression through a detection module $f_d^h(\cdot)$. When multiple heads are stacked, the object encoding $\mathbf{o}_n^h$ at the h-th head is directly used as the proposal encoding for the $(h+1)$-th head, *i.e.*, $\mathbf{p}_n^{h+1} = \mathbf{o}_n^h$. Then, only $\{\mathbf{p}_n^1\}_{n=1}^{N_P}$ in the first head are model parameters (learnable proposal vectors).

During training, at h-th head, we calculate the matching costs between the predictions $\{f_d^h(\mathbf{o}_n^h)\}_{n=1}^{N_P}$ and annotated instances $\mathcal{T}$ pair-wisely. The matching cost between t-th instance and n-th prediction $l^h(n,t)$ is a weighted sum of costs for classification and localization. Then, we find the bipartite matching such that the average matching cost is minimum and assign the labels to each prediction. We usually set $N_T < N_P$, and only N_T predictions at each head are assigned with the object instances (positive) while the rest (N_P-N_T) predictions are supposed to be background. For example, $t = m^h(n|I)$ means the t-th instance in image I is assigned to the prediction originating from the n-th proposal vector $\mathbf{p}_n^1$ while $t > N_T$ means the assigned label is background.

4 CoCo-RCNN for few-shot object detection

In this section, we present the proposed CoCo-RCNN to adapt the pre-trained base detector to novel classes efficiently and effectively. We first review the supervised contrastive learning in Section 4.1 and then explain the detailed strategy for constantly concentrated encoding regularization in Section 4.2. During adaptation, the CoCo loss is jointly minimized with the detection losses.

4.1 Supervised Contrastive Learning

Supervised contrastive learning (SupCT) is proposed to extract discriminative encodings for image classification. Given a batch $\mathcal{B}$ with N_B images, *i.e.*, $|\mathcal{B}| = N_B$, each image $\mathcal{B}(i)$ where $i \in \mathcal{I} \equiv \{1...N_B\}$ is used as an anchor. Then, a positive index set $\mathcal{I}_i' \subset \mathcal{I} \setminus \{i\}$ is selected, such that all images $\mathcal{B}(j)$ for $j \in \mathcal{I}_i'$ are of the same class as $\mathcal{B}(i)$. Then, the SupCT loss is defined as

$$\mathcal{L}_{SupCT}(\mathcal{B}) = \sum_{i \in I} \frac{-1}{|\mathcal{I}_i'|} \sum_{j \in \mathcal{I}_i'} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_j / \tau)}{\sum_{a \in \mathcal{I} \setminus \{i\}} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \quad (1)$$

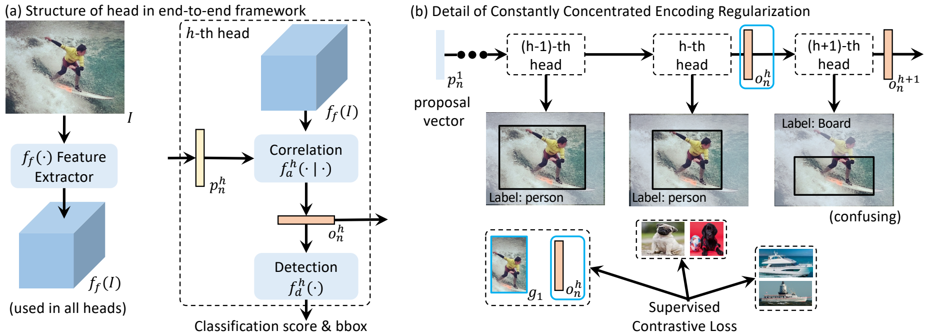
(a) Structure of head in end-to-end framework
I
$f_f(\cdot)$ Feature Extractor
$f_f(I)$
(used in all heads)
h-th head
$f_f(I)$
Correlation $f_a^h(\cdot\mid\cdot)$
p_n^h
o_n^h
Detection $f_d^h(\cdot)$
Classification score & bbox
(b) Detail of Constantly Concentrated Encoding Regularization
p_n^1
proposal vector
(h-1)-th head
h-th head
o_n^h
(h+1)-th head
o_n^{h+1}
Label: person
Label: person
Label: Board
(confusing)
g_1
o_n^h
Supervised Contrastive Loss

Fig. 2. (a): A correlation module is set to connect the proposal encoding with image feature maps and obtain an object encoding for detection. (b) When multiple heads are stacked/cascaded, the object encodings at each head are used as proposal encodings in the next head, our method will sample the object encodings whose prediction are similar as the prediction of input and then perform supervised contrastive learning. We use the assigned labels after bipartite matching as references and specifically highlight $\mathbf{o}_n^h$ and $\mathbf{o}_n^{h+1}$ for better illustration.

where $\mathbf{z}_i \in \mathcal{R}^d$ is the encoding for image $\mathcal{B}(i)$ after l_2-normalization and τ is a temperature hyperparameter used to rescale the affinity score. Minimizing $\mathcal{L}_{SupCT}(\mathcal{B})$ trains the feature extractor to maximize the similarity between features of the same class (positive pairs) while pushing away the features from different classes (negative pairs). Usually, to ensure at least one positive pair can be built for each anchor image in the batch, we set B as large as possible or perform data augmentation to each sample in the batch. As noted in [20], then, the SupCT is in effect performing pair-wise comparison where the disagreement between the two encodings in a positive pair is induced by the variation between image instances and difference resulting from augmentation.

4.2 Constantly Concentrated Encoding

At the h-th head, the f_a^h models the correlation between a proposal encoding $\mathbf{p}_n^h$ and the image feature maps $f_f(I)$. The features with high co-attention is kept in the object encoding $\mathbf{o}_n^h$. When the prediction $f_d^{h-1}(\mathbf{o}_n^{h-1})$ has been close to an object $\mathcal{T}(t)$ in the image, *i.e.*, $t = m^{h-1}(n|I)$ and $t < N_T$, as $\mathbf{p}_n^h = \mathbf{o}_n^{h-1}$, under the Constantly Concentrated encoding (CoCo) regularization, the detector is trained to still concentrate on the discriminative component of object $\mathcal{T}(t)$, such that the prediction $f_d^h(\mathbf{o}_n^h)$ can be improved w.r.t. $f_d^{h-1}(\mathbf{o}_n^{h-1})$. Different from the classification task, being close to an object means both the confidence score for classification and the IoU with annotated boxes for localization are high. As a discriminative object encoding is important to improve detection result [32,42] while no spatial prior is available for learnable proposals, CoCo regularization applies supervised contrastive learning and designs an encoding selection strategy correspondingly to build positive and negative pairs.

Given an image I, we first use the annotated bboxes $\mathcal{T}$ to extract groundtruth encodings $\{\mathbf{g}_t\}_{t=1}^{N_T}$ from the feature maps $f_f(I)$ through RoI pooling [7,15]. Then, for each $\mathbf{g}_t$, we gather the positive object encodings that detect the object $\mathcal{T}(t)$ for calculating $\mathcal{L}_{SupCT}$. As the costs in bipartite matching considers *both classification and localization*, we use the matching results as a reference to sample object encodings. In addition, we need to avoid confusing cases when two objects are of high overlapping or the detector is distracted by other patterns (detailed discussion is provided in Section 5.3). Thus, we jointly consider the matching results from two neighboring heads.

For $\mathbf{g}_t$ where $t \leq N_T$, at the h-th head where $h > 1$, we first check the prediction from the proposal encodings, *i.e.*, the labels assigned at the $(h-1)$-th head, and find the $\mathbf{p}_n^h$ where $t = m^{h-1}(n|I)$. Then, the object encoding $\mathbf{o}_n^h$ will be treated as positive if $\mathcal{T}(t)$ is also assigned to its prediction $f_d^h(\mathbf{o}_n^h)$, *i.e.*,

$$m^h(n|I) = m^{h-1}(n|I) \tag{2}$$

or matching cost between $\mathcal{T}(t)$ and $f_d^h(\mathbf{o}_n^h)$ is smaller, *i.e.*,

$$l^h(n,t) < l^{h-1}(n,t) \tag{3}$$

After checking all heads, we have the object encodings $\mathcal{P}_t$ for $\mathbf{g}_t$ and each encoding in $\mathcal{P}_t$ is of class c_t. In practice, the condition in Eq. (2) has been enough for FSOD. However, as the bipartite matching is obtained when the global matching cost is minimum, it is still possible that $l^h(n,t) < l^{h-1}(n,t)$ though $m^h(n|I) \neq t$. As such, we avoid false negative cases by considering Eq. (3) and it is useful in large-scale training. As mentioned above, we ignore the object encoding $\mathbf{o}_n^h$ that $m^{h-1}(n|I) \neq m^h(n|I)$ and $(m^{h-1}(n|I) - t)(m^h(n|I) - t) = 0$ as it is confusing. Thus, it is possible that none of Eqs. (2) and (3) is met at some heads, and we have $|\mathcal{P}_t| \leq N_T$ (including $\mathcal{P}_t = \varnothing$, *i.e.*, no object encodings selected for $\mathbf{g}_t$).

Though the matching results can help the selection at h-th heads where $h > 1$, the proposal vectors $\{\mathbf{p}_n^1\}_{n=1}^{N_P}$ are not trained to be class-specific. Then, the $\{\mathbf{o}_n^1\}_{n=1}^{N_P}$ cannot be directly determined and the model may not be capable to concentrate on relevant objects at lower heads. Thus, we add class encodings to the learnable proposal vectors to embed class-specific information.

For the convenience of implementation, we directly crop the annotated instances out of the few images used in finetuning and use a frozen ResNet-101 [16] (pretrained on ImageNet [3]) to extract a visual feature for each instance. Then, we average the visual features for each class c as class encodings. During adaptation, we also learn an MLP to post-process the class encodings such that the dimension is the same as the proposal vectors, *i.e.*, $\mathbf{s}_c \in \mathcal{R}^d$. Then, for each class c, we randomly select a subset of $\{\mathbf{p}_n^1\}_{n=1}^{N_P}$ and add the encoding $\mathbf{s}_c$ to each of the selected proposal vectors. Thus, the object encoding $\mathbf{o}_n^1$ will be selected in $\mathcal{P}_t$ with $\mathbf{g}_t$ when both $m^1(n|I) = t$ and the encoding added to $\mathbf{p}_n^1$ is of class c_t.

After gathering $\mathcal{P}_t$ for each $\mathbf{g}_t$ from all heads, we calculate SupCT loss on all selected encodings $\cup_{t=1}^{N_T}\{\{\mathbf{g}_t\} \cup \mathcal{P}_t\}$. In this way, comparing with the conventional SupCT learning which directly performs augmentation on the low-level image pixels, we perform augmentation at the feature-level for each $\mathbf{g}_t$. As the

Table 1. Performance comparison on the PASCAL VOC dataset (nAP$_{50}$).

Method	Venue	Split 1					Split 2					Split 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
YOLOv2-ft [36]	ICCV'19	6.6	10.7	12.5	24.8	38.6	12.5	4.2	11.6	16.1	33.9	13.0	15.9	15.0	32.2	38.4
MetaYOLO [19]	ICCV'19	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
Meta R-CNN [40]	ICCV'19	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
TFA-w/ fc [35]	ICML'20	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 [35]	ICML'20	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
MPSR [39]	ECCV'20	41.7	-	51.4	55.2	61.8	24.4	-	39.2	39.9	47.8	35.6	-	42.3	48.0	49.7
CGDP+FSCN [23]	CVPR'21	40.7	45.1	46.5	57.4	62.4	27.3	31.4	40.8	42.7	46.3	31.2	36.4	43.7	50.1	55.6
CME (MPSR) [22]	CVPR'21	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
FSCE [31]	CVPR'21	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
SVD (FSCE) [38]	NeurIPS'21	46.1	43.5	48.9	60.0	61.7	25.6	29.9	44.8	47.5	48.2	39.5	45.4	48.9	53.9	56.9
FSOD-Up [37]	ICCV'21	43.8	47.8	50.3	55.4	61.7	31.2	30.5	41.2	42.2	48.3	35.5	39.7	43.9	50.6	53.5
CoCo-RCNN		43.9	44.5	53.1	64.6	65.5	29.4	31.3	43.8	44.3	51.8	39.1	43.9	47.2	54.7	60.3
TFA w/ cos † [35]	ICML'20	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
FSCE † [31]	CVPR'21	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
Sparse-RCNN †		28.2	39.5	45.1	51.1	56.3	21.1	30.5	34.1	37.6	43.2	21.4	30.8	37.5	43.7	49.6
CoCo-RCNN †		**33.5**	**44.2**	**50.2**	**57.5**	**63.3**	**25.3**	**31.0**	**39.6**	**43.8**	**50.1**	**24.8**	**36.9**	**42.8**	**50.8**	**57.7**

More comparison can be found in Supp.. †: The performance averaged from multiple runs.

groundtruth encodings $\{\mathbf{g}_t\}_{t=1}^{N_T}$ are obtained through RoI pooling instead of correlation module, we set a linear layer to process $\{\mathbf{g}_t\}_{t=1}^{N_T}$. Also, as the object encodings are obtained from different heads, we also set a linear layer (projector) to process the object encodings for each head.

5 Experiment

5.1 Benchmark Datasets and Implementation Detail

PASCAL VOC consists of 20 classes where the class split for $\mathcal{C}_{base}$ and $\mathcal{C}_{novel}$ are 15 and 5 separately. The base training data $\mathcal{D}_{base}$ are from PASCAL VOC 07+12 trainval sets [4,5]. The novel set $\mathcal{D}_{novel}$ are randomly sampled where $K = \{1, 2, 3, 5, 10\}$. Following [35], we conduct experiments on three standard base-novel class partitions which are marked as $\{1, 2, 3\}$. In each partition, for fair comparison, we use the same sampled novel instances and report the AP$_{50}$ for novel detections (nAP$_{50}$) on PASCAL VOC 2007 test set [4].

MS COCO is derived from COCO14 [25] consisting of 80 classes where the split for $\mathcal{C}_{base}$ and $\mathcal{C}_{novel}$ are 60 and 20. The 20 classes is in common with PASCAL VOC. The train set $\mathcal{D}_{base}$ and $\mathcal{D}_{novel}$ are from COCO14 train set. We set $K = \{1, 10, 30\}$ and report scores of novel detection on COCO 14 val dataset.

Implementation Details. We build CoCo-RCNN based on Sparse-RCNN and use ResNet-101 with FPN [24] as backbone to extract feature maps. For fair comparison, we set $N_H = 6$ and all heads are stacked/cascaded. (*Class encodings*) Following the standard few-shot finetuning pipeline, we also include a few instances of base classes during finetuning. Thus, we gather class encodings for each class from $\mathcal{C}_{base} \cup \mathcal{C}_{novel}$. (*Background encodings*) The GPU memory usage

Table 2. Performance comparison of *novel* detection on the MS COCO dataset.

Method	Venue	1-shot	10-shot						30-shot					
		nAP	nAP	nAP_{50}	nAP_{75}	nAPs	nAPm	nAPl	nAP	nAP_{50}	nAP_{75}	nAPs	nAPm	nAPl
MetaYOLO [19]	ICCV'19		5.6	12.3	4.6	0.9	3.5	10.5	9.1	19.0	7.6	0.8	4.9	16.8
MetaDet [36]†	ICCV'19		7.1	14.6	6.1	1.0	4.1	12.2	11.3	21.7	8.1	1.1	6.2	17.3
Meta R-CNN [40]	ICCV'19		8.7	19.1	6.6	2.3	7.7	14	12.4	25.3	10.8	2.8	11.6	19.0
TFA w/ fc [35]†	ICML'20	2.9	9.1	17.3	8.5	-	-	-	12.2	22.2	11.8	-	-	-
TFA w/ cos [35]†	ICML'20	3.4	9.1	17.1	8.8	-	-	-	12.1	22.0	12.0	-	-	-
MPSR [39]	ECCV'20	2.3	9.8	17.9	9.7	3.3	9.2	16.1	14.1	25.4	14.2	4.0	12.9	23.0
FSCE [31]†	CVPR'21		11.1	-	9.8	-	-	-	15.3	-	14.2	-	-	-
CME [22]	CVPR'21		15.1	24.6	16.4	4.6	16.6	26.0	16.9	28.0	17.8	4.6	18.0	29.2
TIP [21]†	CVPR'21		16.3	**33.2**	14.1	5.4	**17.5**	25.8	18.3	**35.9**	16.9	6.0	**19.3**	29.2
DCNet [17]†	CVPR'21		12.8	23.4	11.2	4.3	13.8	21	18.6	32.6	17.5	**6.9**	16.5	27.4
FSOD-UP [37]	ICCV'21		11.0	-	10.7	4.5	11.2	17.3	15.6	-	15.7	4.7	15.1	25.1
SVD (FSCE) [38]	NeurIPS'21		12.0	-	10.4	4.2	12.1	18.9	16.0	-	15.3	6.0	16.8	24.9
SVD (MPSR) [38]	NeurIPS'21		11.0	-	10.6	4.4	11.5	17.1	16.2	-	15.9	4.6	14.6	26.6
CoCo-RCNN†		**5.2**	**16.4**	26.5	**16.5**	**5.4**	13.4	**27.8**	**19.2**	32.9	**21.0**	5.8	18.1	**32.8**

The full table can be found in Supp. †: The performance averaged from multiple runs.

for object detection is huge, *i.e.*, each GPU can hold at most four images, and the end-to-end object detector is characterized by generating sparse proposals. Thus, the encoding pairs built in each batch is limited, which is different from the related literature [31,20] (*e.g.*, 1024^2 pairs per batch) and results in less efficient training. To mitigate this issue, we include the object encodings of background, *i.e.*, low classification score for all classes and low IoU with all objects in the image, into the CoCo regularization. These object encodings are only used to build negative pairs in SupCT loss and none of them is used as an anchor. (*Multiple runs*) Finally, for each base-novel class split, we average the performance over 10 runs and report the average detection score. More details can be found in Supp.

5.2 Comparison with State-of-the-Arts

As shown in Table 1, we compare CoCo-RCNN with the finetuning-based adaptation methods. For fair comparison, we first show the baseline performance by directly finetuning Sparse-RCNN on the novel instances without any regularization. Benefiting from the multi-head structure, the Sparse-RCNN baseline outperforms the Faster-RCNN baseline TFA [35].

FSCE [31] improves detection precision by learning discriminative encodings (obtained through RoI pooling) and also use the IoU between proposals and annotated bbox to modify the SupCT loss. Instead, we perform comparison on object encodings and each encoding is output by the correlation module without explicit spatial prior. With our CoCo regularization, for each proposal, the object encoding in deeper head is trained to still concentrate on the object detected at lower heads. In this way, we can improve the adaptation performance clearly and keep achieving clear gain upon a stronger baseline. Meanwhile, as the end-to-end object detector can predict high-quality bboxes and the correlation module can generalize to new classes, in Table 2, CoCo-RCNN achieves high score in nAP_{75} and the challenging 1-shot scenario.

Table 3. Ablation study on the constantly concentrated encoding regularization.

Method	VOC 10-shot			MS COCO 10-shot			MS COCO 30-shot		
	nAP	nAP$_{50}$	nAP$_{75}$	nAP	nAP$_{50}$	nAP$_{75}$	nAP	nAP$_{50}$	nAP$_{75}$
hard-deepest	13.0	16.8	14.1	2.7	7.5	1.4	3.1	5.2	3.0
hard-lowest	18.5	25.0	20.3	3.4	9.2	1.9	6.7	10.0	6.5
distillation	43.1	59.4	46.5	15.9	28.1	15.9	16.8	28.0	17.4
contrastive	44.5	62.1	48.2	17.0	29.2	16.8	18.7	30.4	19.4
iou-supct	46.0	61.8	51.0	17.7	29.6	17.8	20.0	31.0	19.8
input-supct	43.3	60.2	47.7	17.3	30.0	17.2	19.1	30.7	19.9
CoCo-RCNN	47.2	65.5	51.5	18.1	30.4	18.2	20.6	33.8	21.4

5.3 Ablation on Constantly Concentrated Encoding

Different from the conventional object detectors which directly predict the class and location for each object, the end-to-end object detectors in effect improve the detection for each proposal across heads. Thus, we set constantly concentrated encoding regularization to prevent the object encodings from being distracted.

At each head, an object encoding will be selected as positive when the assigned labels before and after the head is consistent. In this way, for each proposal, our loss serves as a soft regularization where we do not force the model to make consistent prediction at all heads. Then, we discuss relevant alternatives for the regularization and summarize the results on Table 3.

- (*hard-deepest*) We assign the same label for predictions originating from the same proposal and use the matching results at the last head to assign labels.
- (*hard-lowest*) Similar to *hard-deepest*, we use the bipartite matching result at the first head as reference to assign labels for all heads.
- (*distillation*) At each head, the predictions are used as soft-labels to supervise the previous head. Thus, we minimize the kl-divergence loss between the probability distribution of h-th head and $(h-1)$-th head (for classification) and the $l1$ loss between the predicted bboxes (for localization).
- (*contrastive*) We perform contrastive learning among the sampled object encodings where only encodings in $\mathcal{P}_t \cup \{\mathbf{g}_t\}$, *i.e.*, corresponding to the same object, will be treated as positive to each other.
- (*iou-supct*) Use contrastive proposal encoding (CPE) proposed in FSCE [31] for the constantly concentrated encoding regularization.
- (*input-supct*) At each head, use the label assigned to the prediction at the previous head to select the object encoding in $\mathcal{P}_t$ for $\mathbf{g}_t$.

As the learnable proposals are class-irrelevant, it is hard for the object encodings at lower heads to detect discriminative components of objects. Thus, the predictions by the lowest head and the deepest head vary, resulting in disparate label assignments. Then, when we naively use the same label-prediction assignment across all heads, the precision drops clearly (*hard-deepest*, *hard-lowest*).

Instead of doing the hard-label assignment, *distillation* adds soft labels to the original detection losses. At each head, we observe the $l1$ loss for localization is small and the main contribution is thus from the classification. In particular, the classification logits can indicate the relationships between classes. Then, combining one-hot labels and soft labels will help with the classifier training.

Thus, the model could be confused and the object encodings is less discriminative. However, since we have limited positive pairs within each batch, it is rare to have different object instances of the same class within one batch. As such, the features can still be trained to be discriminative, and the precision drop w.r.t. the full method is not huge.

For *iou-supct*, the CPE differs from SupCT by using the IoU to reweight the loss for each anchor feature. Then, in CPE, the loss from an anchor encoding will contribute less to the detector update if the IoU is low. In FSCE, as the proposal encodings are extracted by RoI pooling the image feature maps, the IoU between the proposal and groundtruth bbox can thus be directly calculated. However, for the end-to-end object detectors, each object encoding is obtained through a correlation module and no spatial prior is available, we thus use the predicted bbox as a reference to calculate the IoU as weights. Then, replacing SupCT with CPE does not result in significant difference. After all, as the object encodings are not directly pooled from the feature maps, the referred bbox may not be precise. Meanwhile, for the selected positive object encodings, we observed that the IoU of the predicted bbox is high. Thus, CPE is similar to that of SupCT.

Lastly, *input-supct* adjusts the sampling strategy by only using the label assignment at the current head for selection. Then, the object encodings at lower heads will always be selected. However, the encodings at lower heads may not be discriminative in nature, and enforcing CoCo loss on those object encodings may confuse the model, *e.g.*, the two objects with high overlapping (a child is playing with a dog) can only be distinguished at deeper heads. Besides, when the object encoding is distracted by objects of different classes, applying CoCo loss will also be risky, *e.g.*, for the same proposal, the ‘chair’ is initially detected at lower heads but the encodings at deeper heads are distracted by ‘couch’. In addition, as the object encoding at the h-th head could succeed in detecting large objects but may fail in finding tiny instances, our sampling strategy in effect dynamically determines the object encodings used for comparison.

5.4 Discussion

Ablation study of our full method is summarized in Table 4. Compared with baseline (Row_1), adding class encodings to proposal vectors (Row_2) or performing SupCT among object encodings from the 2nd to 6th heads (Row_3) can clearly facilitate the final detection, while combining them can further improve the detection precision (Row_4). As the pairs to be sampled from each batch are limited, we thus include negative object encodings of background in the regularization, which mitigates the training inefficiency. By comparing the anchor feature and the background encodings during the network training, the object encodings can be more discriminative and we are thus capable to improve the performance. To note, even though each proposal vector $\mathbf{p}_n^1$ is added with an encoding of class c, $\mathbf{p}_n^1$ is not trained to predict the objects of class c specifically.

Table 4. Ablation study of the full method.

2nd-6th heads	Class Encoding	Negative	VOC 10-shot nAP	VOC 10-shot nAP_{50}	VOC 10-shot nAP_{75}	MS COCO 10-shot nAP	MS COCO 10-shot nAP_{50}	MS COCO 10-shot nAP_{75}	MS COCO 30-shot nAP	MS COCO 30-shot nAP_{50}	MS COCO 30-shot nAP_{75}
			45.0	58.1	48.9	13.5	22.7	13.6	16.3	26.3	16.8
	✓		42.3	61.2	45.8	14.9	24.9	15.3	17.3	28.3	17.9
✓			45.2	63.5	49.2	16.8	28.4	16.8	19.2	31.1	19.7
✓	✓		47.0	65.3	51.0	17.4	29.4	17.4	20.1	32.6	20.6
✓		✓	46.2	64.5	50.8	17.6	30.2	17.4	19.9	32.7	20.7
✓	✓	✓	47.2	65.5	51.5	18.1	30.4	18.2	20.6	33.8	21.4

Table 5. Multiple runs for class encodings

Run	1	2	3	4	5
VOC 10-shot	65.5	65.3	65.4	65.5	65.5
MS COCO 30-shot	20.6	20.6	20.9	20.4	20.5

Table 6. Ablation study of projector

Project	Separate	Shared
VOC 10-shot	65.5	64.0
MS COCO 30-shot	20.6	19.8

After all, through self-attention [34,32], the discriminate components shared between two classes can benefit each other, *e.g.*, cats and dogs have four legs.

During testing, we randomly assign the proposal vectors for each class. As we have 300 learnable proposal vectors, each class encoding is added to at least 3 (10) proposal vectors for MS COCO (Pascal VOC). However, as compared in Table 5, the overall performance is stable.

Large-scale object detection. Besides the FSOD task, our constantly concentrated encoding can also benefit large-scale object detection. As the parameters are completely trained from scratch, we do not add class encodings on the proposal vectors and the object encodings are only sampled from the 2nd to 6th heads. As shown in Fig. 3(a), the detection AP at early checkpoints grows faster and the final detection performance is also improved from 46.3 AP to 47.5 AP. Meanwhile, as the bipartite matching is performed for each head separately, for a proposal whose prediction at h-th head is assigned to t-th instance, *i.e.*, $m^h(n|I) = t$ and $t \leq N_T$, its prediction at $(h+1)$-th head can be assigned to the background though the prediction is closer to that instance, *i.e.*, $m^h(n|I) > N_T$ and $l^h(n,t) < l^{h-1}(n,t)$. Thus, including Eq. (3) can contribute 0.8 AP gain.

Deep supervision, *i.e.*, supervising the prediction at each head separately, is important for end-to-end detectors [1]. As shown in Fig. 3(b), it is also necessary for few-shot finetuning. With deep supervision, the lower detection heads can also be tuned to adapt to the novel classes such that the object encodings are lower heads can learn to concentrate on the novel object instances. However, when deep supervision is removed, performance by the deepest head drops significantly. Furthermore, we vary the number of heads in the end-to-end object detector and retrain the model on VOC and MS COCO. As summarized in Fig. 3(c), when fewer heads are set in the framework, the adaptation precision is even worse. It might because the gradient from deeper heads can also benefit the adaptation of lower heads. As such, when fewer heads are set, the precision is compromised.

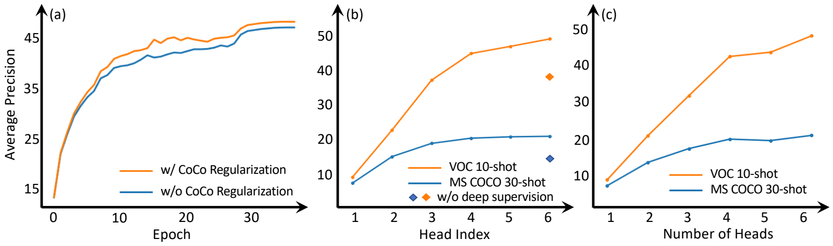

Fig. 3. (a) Testing curve on COCO17 (large-scale). (b) Detection precision at each head of an adapted six-head model. (c) Detection precision by detectors with different number of stacked heads.

Projectors are the linear layers set to map the object encodings to a common feature space such that the SupCT loss can be calculated. As the object encodings are sampled from all heads, we set a separate projector at each head. Then, as shown in Table 6, we observed that the detection score will drop a bit when we share the parameters across all projectors. Even though a few more parameters are introduced, sharing the parameters will assume that the object encodings will exactly be in the same space. Instead, setting separate sets of parameters will be more flexible. However, we can still see that the object encodings are similar to each other as the performance drop is not significant.

6 Conclusion

In this paper, we have proposed CoCo-RCNN, an adaptation strategy of end-to-end object detectors for FSOD. As a degraded prediction at each head may result in inefficient adaptation when the training data is limited, we design the constantly concentrated encoding regularization. We use the label assignments at neighboring heads as references to gather object encodings, and then perform supervised contrastive learning to make them discriminative. In this way, the detector is trained to keep concentrating on the objects that have been detected and constantly improve the detection precision. Experiments on two datasets demonstrate the effectiveness of CoCo-RCNN. Detailed ablation study is provided to compare the potential variances of CoCo regularization and ours also benefits the large-scale training. In addition to make the encodings at each head discriminative, the relationship between encodings of different heads will be studied in the future to further explore the strength of end-to-end detectors.

Acknowledgement This material is based on research sponsored by Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-1000. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation therein. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Air Force Laboratory, DARPA or the U.S. Government.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
2. Dai, Z., Cai, B., Lin, Y., Chen, J.: Up-detr: Unsupervised pre-training for object detection with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1601–1610 (2021)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
5. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html
6. Fan, Q., Zhuo, W., Tang, C.K., Tai, Y.W.: Few-shot object detection with attention-rpn and multi-relation detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4013–4022 (2020)
7. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 1440–1448 (2015)
8. Han, G., He, Y., Huang, S., Ma, J., Chang, S.F.: Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3263–3272 (2021)
9. Han, G., Huang, S., Ma, J., He, Y., Chang, S.F.: Meta faster r-cnn: Towards accurate few-shot object detection with attentive feature alignment. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 780–789 (2022)
10. Han, G., Ma, J., Huang, S., Chen, L., Chang, S.F.: Few-shot object detection with fully cross-transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5321–5330 (2022)
11. Han, G., Ma, J., Huang, S., Chen, L., Chellappa, R., Chang, S.F.: Multimodal few-shot object detection with meta-learning based cross-modal prompting. arXiv preprint arXiv:2204.07841 (2022)
12. Han, G., Zhang, X., Li, C.: Revisiting faster r-cnn: A deeper look at region proposal network. In: International Conference on Neural Information Processing. pp. 14–24 (2017)
13. Han, G., Zhang, X., Li, C.: Single shot object detection with top-down refinement. In: 2017 IEEE International Conference on Image Processing (ICIP). pp. 3360–3364. IEEE (2017)
14. Han, G., Zhang, X., Li, C.: Semi-supervised dff: Decoupling detection and feature flow for video object detectors. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1811–1819 (2018)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
17. Hu, H., Bai, S., Li, A., Cui, J., Wang, L.: Dense relation distillation with context-aware aggregation for few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10185–10194 (2021)

18. Huang, S., Ma, J., Han, G., Chang, S.F.: Task-adaptive negative envision for few-shot open-set recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7171–7180 (2022)
19. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8420–8429 (2019)
20. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in Neural Information Processing Systems **33**, 18661–18673 (2020)
21. Li, A., Li, Z.: Transformation invariant few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3094–3102 (2021)
22. Li, B., Yang, B., Liu, C., Liu, F., Ji, R., Ye, Q.: Beyond max-margin: Class margin equilibrium for few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7363–7372 (2021)
23. Li, Y., Zhu, H., Cheng, Y., Wang, W., Teo, C.S., Xiang, C., Vadakkepat, P., Lee, T.H.: Few-shot object detection via classification refinement and distractor retreatment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15395–15403 (2021)
24. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
25. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
27. Ma, J., Xie, H., Han, G., Chang, S.F., Galstyan, A., Abd-Almageed, W.: Partner-assisted learning for few-shot image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10573–10582 (2021)
28. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems **28** (2015)
30. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
31. Sun, B., Li, B., Cai, S., Yuan, Y., Zhang, C.: Fsce: Few-shot object detection via contrastive proposal encoding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7352–7362 (2021)
32. Sun, P., Zhang, R., Jiang, Y., Kong, T., Xu, C., Zhan, W., Tomizuka, M., Li, L., Yuan, Z., Wang, C., et al.: Sparse r-cnn: End-to-end object detection with learnable proposals. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14454–14463 (2021)
33. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: European Conference on Computer Vision. pp. 266–282. Springer (2020)

34. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
35. Wang, X., Huang, T., Gonzalez, J., Darrell, T., Yu, F.: Frustratingly simple few-shot object detection. In: International Conference on Machine Learning. pp. 9919–9928. PMLR (2020)
36. Wang, Y.X., Ramanan, D., Hebert, M.: Meta-learning to detect rare objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9925–9934 (2019)
37. Wu, A., Han, Y., Zhu, L., Yang, Y.: Universal-prototype enhancing for few-shot object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9567–9576 (2021)
38. Wu, A., Zhao, S., Deng, C., Liu, W.: Generalized and discriminative few-shot object detection via svd-dictionary enhancement. Advances in Neural Information Processing Systems **34** (2021)
39. Wu, J., Liu, S., Huang, D., Wang, Y.: Multi-scale positive sample refinement for few-shot object detection. In: European conference on computer vision. pp. 456–472. Springer (2020)
40. Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., Lin, L.: Meta r-cnn: Towards general solver for instance-level low-shot learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9577–9586 (2019)
41. Ypsilantis, N.A., Garcia, N., Han, G., Ibrahimi, S., Van Noord, N., Tolias, G.: The met dataset: Instance-level recognition for artworks. In: Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2) (2021)
42. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: International Conference on Learning Representations (2020)