Three-stage Training Pipeline with Patch Random Drop for Few-shot Object Detection

Shaobo Lin[1], Xingyu Zeng[1], Shilin Yan[2], and Rui Zhao[1]

[1] Sensetime Research
{linshaobo,zengxingyu,zhaorui}@sensetime.com
[2] Fudan University, Shanghai, China
tattoo.ysl@gmail.com

Abstract. Self-supervised learning (SSL) aims to design pretext tasks for exploiting the structural information of data without manual annotation, which has been widely used in few-shot image classification for improving the generalization of the model. However, few works explore the influence of SSL on Few-shot object detection (FSOD) which is a more challenging task. Besides, our experimental results demonstrate that using a weighted sum of different self-supervised losses causes performance degradation compared to using a single self-supervised task in FSOD. To solve these problems, firstly, we introduce SSL into FSOD by applying SSL tasks to the cropped positive samples. Secondly, we propose a novel self-supervised method: patch random drop, for predicting the location of the masked image patch. Finally, we design a three-stage training pipeline to associate two different self-supervised tasks. Extensive experiments on the few-shot object detection datasets, *i.e.*, Pascal VOC, MS COCO, validate the effectiveness of our method.

Keywords: Few-shot Object Detection · Self-supervised Learning.

1 Introduction

Deep Neural Networks (DNNs) have achieved great progress in many computer vision tasks [27, 26, 22]. However, the impressive performance of these models largely relies on a large amount of data as well as expensive human annotation. When the annotated data are scarce, DNNs cannot generalize well to testing data especially when the testing data belong to different classes of the training data. In contrast, humans can learn to recognize or detect a novel object quickly with only a few labeled examples. Few-shot learning, therefore, becomes an important research topic to learn from only a few examples. However, the generalization ability of the few-shot model is not satisfactory due to the lack of sufficient samples. Therefore, a novel strategy for improving the generalization power of a deep model is required.

Recently, self-supervised learning attracts many researchers' attention, because it can improve the generalization of the network without involving manual annotations. By designing pretext tasks to exploit the structural information

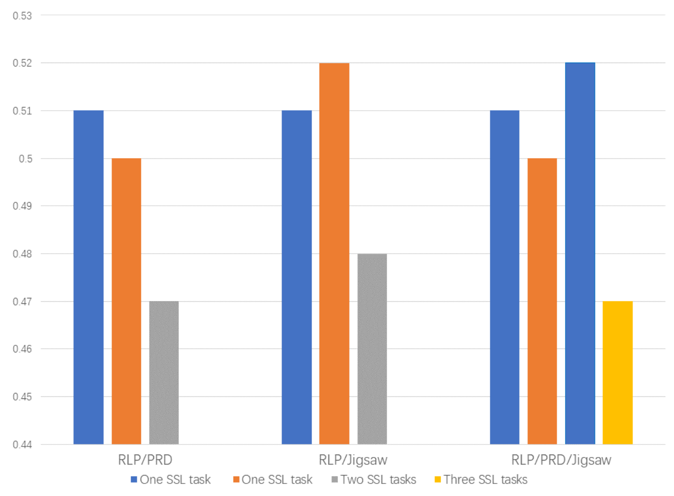


Fig. 1. Minimizing a weighted sum of multiple self-supervised losses can cause the degradation of performance compared to using only one self-supervised task. The adopted self-supervised learning (SSL) tasks include PRD, RLP, and Jigsaw. Y-axis means the novel AP50 on Pascal VOC.

of data itself, self-supervised learning aims to predict pseudo-labels only using the input images. Specifically, relative location prediction (RLP) [4] guides the model to learn the visual relative location in the image. MAE [14] re-constructs a high proportion of the masked patches. However, there is still a quality of the human vision, which is ignored by the current self-supervised learning. Specifically, if a region of an image is randomly cropped, human can easily identify the location of the significant change even the natural images have highly diversity. The reason is that the cropped patch can seriously effect the structural information of an image. To this end, we want to mask a random patch in an image, and use the model to predict the location of the masked patch. We call this method: patch random drop (PRD).

Some self-supervised tasks are gradually used to improve the performance of few-shot classification [1, 9, 20, 28, 29]. However, there are few works that introduce self-supervised learning into few-shot object detection which is a more challenging task. Self-supervised learning is to model the internal characteristics of a certain category or instance, which is applied to the whole image in few-shot classification. It is not suitable for the detection task, because object detection only cares about the positive samples. Using the self-supervised method like classification can make a model consider more about some changes of irrelevant background since there is a large region of background in an image, and thus cause the degradation of performance. In our work, by using the strategy of positive sample selection, we can introduce a self-supervised task into the few-shot object detection framework in such a simple way. Besides, we explore the integration of multiple self-supervised tasks in this framework.

When using multiple self-supervised tasks [28, 29], the commonly used solution to associate the main task and auxiliary self-supervised tasks is optimizing the shared parameters by minimizing a weighted sum of the all losses. However, as the objectives of distinct tasks are different and the relationship between them is complicated, optimizing all losses can cause conflict. In few-shot object detection, we figure out that minimizing a weighted sum of the main tasks and several self-supervised tasks does not work, even worse than using a single self-supervised method as shown in Fig. 1. To solve this problem, we propose a novel three-stage training pipeline to associate two self-supervised tasks, including PRD and RLP. By using our proposed PRD, our method can outperform the baseline with a single self-supervised task, and achieves the state-of-the-art performance.

Our contributions can be summarized as three fold:

- We propose a novel self-supervised method: patch random drop (PRD) for few-shot object detection.
- To the best of our knowledge, we are the first to introduce multiple self-supervised tasks into few-shot object detection. We design a three-stage training pipeline for associating two different self-supervised methods, including PRD and RLP.
- Experiments evaluate the effectiveness of our approach on the few-shot object detection datasets, i.e., Pascal VOC, MS COCO.

2 Related Work

2.1 Few-shot Object Detection

There are two mainstream approaches in few-shot object detection, including meta-learning based and pre-train finetune-based methods.

Meta-learning based Methods Some works use meta-learning [7, 17, 36, 33, 19], where a meta-learner is introduced to acquire class agnostic meta-knowledge which is transferred to novel classes. These methods usually extract meta-knowledge from a set of auxiliary tasks via the episode-based strategy [31], where each episode contains C classes and K samples of each class, i.e., C-way K-shot. With the help of a meta learner that takes the support images as well as the bounding box annotations as inputs, the feature re-weighting modules are applied to a single-stage object detector (YOLOv2) [17] and a two-stage object detector (Faster R-CNN) [36]. A weight prediction meta-model is introduced to learn the category-agnostic components from base class examples while predicting parameters of category-specific components from the few examples [33]. CME [19] uses a class margin equilibrium (CME) approach, with the aim to optimize both feature space partition and novel class reconstruction in a systematic way. Transformation Invariant Principle (TIP) [18] is proposed for various meta-learning models by introducing consistency regularization on predictions from the transformed images.

Pre-train Finetune-based Methods Pre-train finetune-based approaches are the current one of the leading paradigms for few-shot object detection, which utilize a two-stage training pipeline to leverage the knowledge of base classes. TFA [32] is a simple two-stage fine-tuning approach, which significantly outperforms the earlier meta-learning methods. Following this framework, MPSR [35] adopts multi-scale positive sample refinement to handle scale variance problem. FSCE [30] proposes a simple yet effective approach to learning contrastive-aware object proposal encodings that facilitate the classification of detected objects.

2.2 Self-supervised Learning

Self-supervised methods have achieved great success in AI, including NLP and CV. In CV, self-supervised learning aims to construct some annotation-free pretext tasks to predict pseudo-labels only using the input images. The recent advances of self-supervised learning include two types: image generation and contrastive learning [1]. Image generation designs the pretext tasks to exploit semantic visual representation such as rotation prediction [10], relative location prediction [4], and jigsaw puzzle [25]. MAE [14] is a new image generation method via re-constructing a high proportion of the masked patches. Our proposed PRD belongs to this category. Contrastive learning [15, 3, 11] is training the feature representation of samples by bringing the features of positive pairs closer, and spreading the features of negative pairs apart. Momentum Contrast [15] trains a representation encoder by matching an encoded query to a dictionary of encoded keys via a contrastive loss. SimCLR [3] uses the normalized temperature-scaled cross-entropy loss as the contrast loss. BYOL [11] only relies on positive pairs to learn the feature representation.

When coming to few-shot learning with self-supervised tasks, [9] proposes a multi-task method combining the self-supervised auxiliary loss with the main few-shot classification loss. Conditional self-supervised learning (CSS) [1] is proposed to use prior knowledge to guide the representation learning of self-supervised tasks and introduces a three-stage training pipeline for few-shot image classification. There are some works [20, 28] aim to combine multiple self-supervised tasks with a few-shot classification model via the weighted summation loss. However, [28] shows that the improvement from the association of rotation prediction and BYOL is limited when compared to only using rotation prediction or BYOL. The results of [29] indicate that combining multiple self-supervised tasks via the weighted summation loss even hurts the performance of few-shot classification in some cases. Due to the limited samples, the summation of multiple self-supervised losses can hurt the learning of key semantic information.

In few-shot object detection, few works explore the influence of SSL. Besides, our experiments demonstrate that using a weighted sum of different self-supervised losses causes performance degradation compared to using a single self-supervised task. To solve these problems, firstly, we introduce SSL into FSOD by applying SSL tasks to the cropped positive samples. Secondly, we propose a self-supervised method: patch random drop, for predicting the location of the masked image patch. Finally, we design a three-stage training pipeline to associate two different self-supervised tasks, including our proposed PRD and commnly used RLP. Our method can benefit from this training paradigm and learn a better semantic representation than that of a weighted summation loss. However, our goal is to use SSL to improve FSOD and we do not validate our methods on general object detection tasks.

3 Method

3.1 Preliminary

In few-shot detection, given a labeled base dataset $D_B = \{x_{Bi}, y_{Bi}\}$, there are C_B base classes with a large number of images in each class. Novel dataset $D_N = \{x_{Ni}, y_{Ni}\}$ with novel classes C_N consists of a few samples in each class, where xi and yi indicate training samples and labels, respectively. C_B and C_N do not have overlapping categories. The number of objects for each class in C_N is K for K-shot detection. The model is expected to detect objects in the test set with classes in $C_B \cup C_N$.

Pre-train finetune-based methods adopt a simple two-stage training pipeline. In the pre-training stage, the model is trained on base classes to obtain a robust feature representation. In the fine-tuning stage, the pre-trained model is then fine-tuned on a balanced few-shot set which is composed of both base and novel classes ($C_B \cup C_N$). Our proposed three-stage training pipeline is based on this framework.

3.2 Methodology

As illustrated in Fig. 2, the proposed training pipeline has three stages, in which the first stage is pre-training stage and the following two stages are fune-tuning stage. In the first stage, two models are pre-trained with two separate self-supervised tasks, including PRD and RLP, based on base classes, and then the backbone-P and backbone-R are got. In the second stage, the model pre-trained with PRD is finetuned on base and novel classes, obtaining a new backbone-P. In the third stage, the final model is re-initialized with the backbone-R and guided by the backbone-P based on base and novel classes. In all stages, we propose to use a positive selection module to crop the positive samples for the self-supervised tasks in our framework. When testing, the self-supervised heads and the positive selection module can be removed, thus our method is easy to deploy and use.

3.3 The Pre-Training Stage

There are two parallel steps in the pre-training stage. The difference between these two steps is the using auxiliary task which is RLP or PRD. Patch random drop (PRD) is our proposed self-supervised task, which is achieved by masking a random patch in an image and using the model to predict the location of the masked patch.

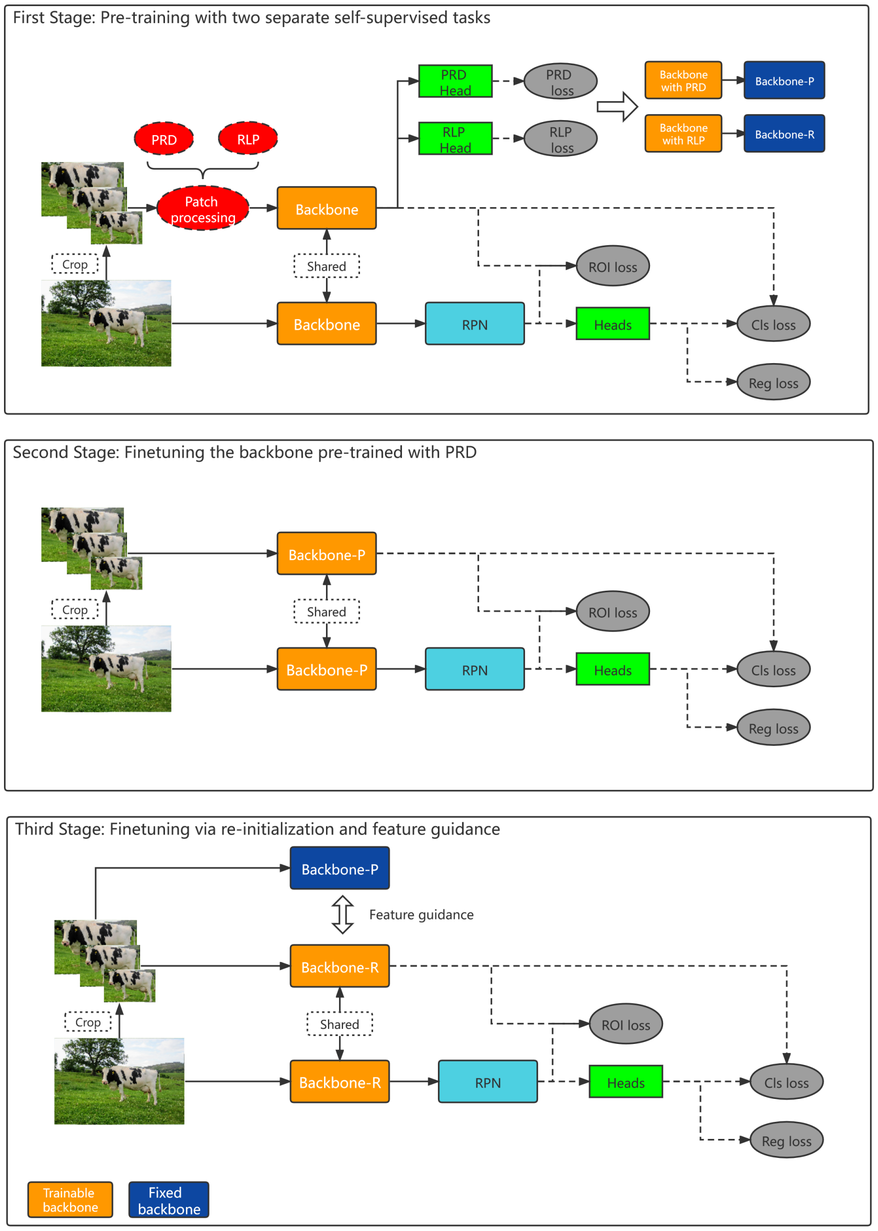


Fig. 2. Our three-stage training pipeline for few-shot object detection with two self-supervised methods, including PRD and RLP. In the first stage, two models are pre-trained with two separate self-supervised tasks based on base classes, and then backbone-P and backbone-R are got. In the second stage, the model pre-trained with PRD is finetuned on base and novel classes, obtaining a new backbone-P. In the third stage, the final model is re-initialized with the backbone-R and guided by the backbone-P based on base and novel classes.

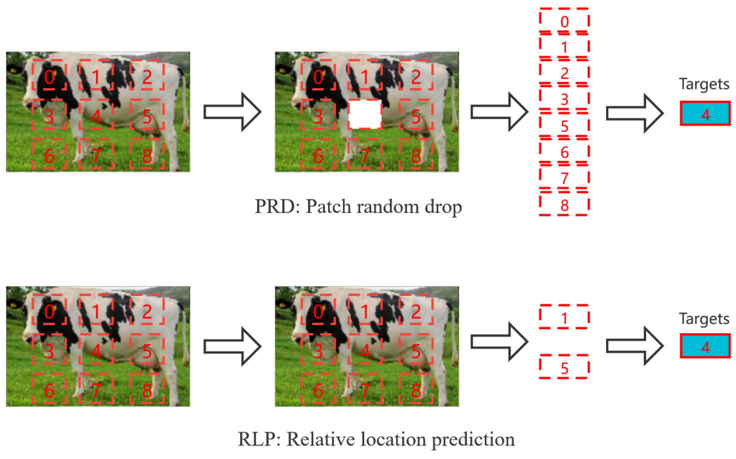


Fig. 3. The comparison between our PRD and RLP.

What is PRD and Why use PRD? The comparison between our PRD and RLP is shown in Fig. 3. RLP [4] is a self-supervised method for guiding model to learn the visual relative location in the image. It divides an image into 9 patches, and two patches are randomly selected as the inputs. The model is to predict the relative location of the two input patches. Similar to RLP, PRD divides an image into 9 patches. The masked patches are removed before model forwarding like MAE [14]. After removing the masked patch, the remaining patches are concatenated in order as the input of our model. The output of PRD head predicts the location of the dropped patch. The difference between PRD and MAE is the goal of model training. Specifically, PRD is predicting the location of the masked patch, while MAE is to re-construct the masked patches. The advantages of PRD to FSOD tasks include improving FSOD models and achieving better accuracy when associated with RLP. Moreover, PRD provides a new direction for designing more self-supervised methods by predicting the location of patches of our concern.

We consider the performance of applying RLP [4], Jigsaw Puzzle [25] and our PRD in our framework. In addition, we also do experiments on contrastive learning methods, such as simCLR and BYOL, and find them improvement is limited. MAE is not selected for implementation since it is designed for transformer. The combination of PRD and RLP can achieve better performance than others. Therefore, we build our framework with PRD and RLP.

It is worth noting that the proposed method is not a straightforward application of self-supervised learning in few-shot object detection. We propose to use the strategy of positive sample selection to obtain the foreground targets for self-supervised tasks. Besides, we explore the integration of multiple self-supervised tasks in our framework.

Pre-training with RLP We train the model with the main few-shot object detection task and the RLP self-supervised task. The learned feature representation is backbone-R in the first stage of Fig. 2.

$$L_1 = L_{main} + w_1 * L_{RLP} \tag{1}$$

Where L_{main} is the main few-shot object detection task defined as follows. L_{RLP} is the loss for relative location prediction. w_1 is the weight of L_{RLP}.

$$L_{main} = L_{roi} + L_{cls} + L_{reg} \tag{2}$$

Where L_{roi} is applied to the output of RPN and the feature from the cropped positive samples. L_{roi} is to distinguish foreground from backgrounds, L_{cls} is a cross-entropy loss for the box classifier, and L_{reg} is a smoothed L1 loss for the box regressor.

Given a set of N training images $D = \{X_i\}_{i=0}^{N}$, the self-supervised training objective that RLP learns to solve is:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L_{RLP}(X^i, \theta, \eta) \tag{3}$$

θ and η are the learnable parameters of backbone and RLP head. The loss function L_{RLP} is defined as:

$$-|y_2 - y_1| * log(H_R(P_R|\eta)) \tag{4}$$

Where $|y_2 - y_1|$ is the label for L_{RLP} and P_R is the predicted probability for the input. H_R is RLP head which is a fully-connected layer. P_R is computed as:

$$P_R = cat(F(g(X_i|y_1)|\theta), F(g(X_i|y_2)|\theta)) \tag{5}$$

Where g is patch processing for RLP by randomly selecting two image patches according to the label y_1 and y_2 that $y_1, y_2 \in \{0,1,2,3,4,5,6,7,8\}$. $F(*)$ obtains the predicted probability for the input and θ is the learnable parameters of model F. Then the features from the two selected image patches are concatenated.

Pre-training with PRD We train the model with the main few-shot task and the PRD self-supervised task. The learned feature representation is backbone-P in the first stage of Fig. 2. The loss L_2 in this step is computed as:

$$L_2 = L_{main} + w_2 * L_{PRD} \tag{6}$$

L_{PRD} is the loss of PRD task. w_2 is the weight of L_{PRD}.

Given a set of N training images $D = \{X_i\}_{i=0}^{N}$, the self-supervised training objective that PRD aims to solve is:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L_{PRD}(X^i, \theta, \eta) \tag{7}$$

θ and η are the learnable parameters of backbone and PRD head. The loss function of L_{PRD} is defined as:

$$-y * log(H_P(O|\eta)) \tag{8}$$

Where y represents the location of the dropped patch which is the label for L_{PRD} and O is the predicted probability for the input. H_P is the PRD head with the learnable parameters η. H_P is a fully-connected layer. O is defined as:

$$O = F(g(X_i|y)|\theta) \tag{9}$$

Where g is patch processing for PRD by dropping a random patch according to the label y that $\{y \in 0, 1, 2, 3, 4, 5, 6, 7, 8\}$, and remaining image patches are concatenated as the input. θ is the learnable parameters of backbone F.

3.4 The Fine-tuning Stage

In the second stage, we finetune the backbone pre-trained with PRD. The loss used in this stage is L_{main}.

In the third stage, we use the feature representation trained by our proposed PRD to guide the learning of the representation trained from RLP. The parameters of backbone-P are fixed and the parameters of backbone-R are trainable for prediction. The right order of the feature guidance is important which is proved in our experimental section. The loss of the third stage is L_f defined as:

$$L_f = L_{main} + w_3 * L_{FG} \tag{10}$$

L_{FG} is the loss for feature guidance, which is computed as following. Feature guidance is different from knowledge distillation which uses a large weight to ensure the consistency of features or probability value between the teacher model and the student model. When increasing w_3, the performance of the final model declines in our setting. By setting w_3 properly to guide the feature learning, our model can achieve better performance, thus our strategy is named by feature guidance.

$$L_{FG} = \sum_{i=1}^{K}(|F_p^i - F_r^i|^2) \tag{11}$$

Where F_p and F_r are the feature representations from the backbone-P and backbone-R. K means the number of FPN levels.

The work which is most similar to our pipeline is CSS [1]. CCS also introduces self-supervised learning into a few-shot model and proposes a muti-stage training pipeline. However, the differences between our training pipeline and CSS are as follows. First, we are the first work to introduce the image-level self-supervised methods into few-shot object detection, in which the self-supervised tasks are applied to the cropped positive samples. Other methods use self-supervised learning to improve few-shot classification. Second, our work associates two different

		split 1			split 2			split 3		
Method/Shot		1	3	10	1	3	10	1	3	10
FRCN+ft [17]	ICCV2019	11.9	29	36.9	5.9	23.4	28.8	5.0	18.1	43.4
FRCN+ft-full [36]	ICCV2019	13.8	32.8	45.6	7.9	26.2	39.1	9.8	19.1	45.1
FR [17]	ICCV2019	14.8	26.7	47.2	15.7	22.7	40.5	21.3	28.4	45.9
MetaDet [33]	ICCV2019	18.9	30.2	49.6	21.8	27.8	43	20.6	29.4	44.1
Meta R-CNN [36]	ICCV2019	19.9	35	51.5	10.4	29.6	45.	14.3	27.5	48.1
TFA [32]	ICML2020	25.3	42.1	52.8	18.3	30.9	39.5	17.9	34.3	45.6
MPSR [35]	ECCV2020	41.7	51.4	61.8	24.4	39.2	47.8	35.6	42.3	49.7
CME [19]	CVPR2021	41.5	50.4	60.9	27.2	41.4	46.8	34.3	45.1	51.5
FSCN [21]	CVPR2021	40.7	46.5	62.4	27.3	40.8	46.3	31.2	43.7	55.6
Retentive R-CNN [8]	CVPR2021	42.4	45.9	56.1	21.7	35.2	40.3	30.2	43	50.1
HallucFsDet [37]	CVPR2021	47	46.5	54.7	26.3	37.4	41.2	40.4	43.3	49.6
FSCE [30]	CVPR2021	44.2	51.4	63.4	27.3	43.5	50.2	37.2	47.5	58.5
UPE [34]	ICCV2021	43.8	50.3	61.7	31.2	41.2	48.3	35.5	43.9	53.5
QA-FewDet [12]	ICCV2021	42.4	55.7	63.4	25.9	**46.6**	51.1	35.2	47.8	53.5
Meta faster-rcnn [13]	AAAI2021	43	**60.6**	**65.4**	27.7	46.1	**51.4**	40.6	**53.4**	58.6
FADI [2]	NIPS2021	50.3	54.2	63.2	30.6	40.3	48	**45.7**	49.1	**59.6**
Ours		**54.6**	56.5	61.4	**37.9**	44.4	47.6	42.8	46.6	51.2

Table 1. Comparison with state-of-the-art few-shot object detection methods on VOC2007 test set for novel classes of the three splits. **black** indicate state-of-the-art (SOTA).

self-supervised methods and uses feature guidance to guide the learning of feature representation, while CSS only consider one self-supervised method and use a graph convolution network to generate a weight matrix for the final feature.

4 Experiments

4.1 Datasets and Evaluation Protocols

We evaluate our methods on Pascal VOC [6, 5] and MS COCO [23]. In PASCAL VOC, we adopt the common strategy [27, 26] that using VOC 2007 test set for evaluating while VOC 2007 and 2012 train/val sets are used for training. Following [36], 5 out of its 20 object categories are selected as the novel classes, while keeping the remaining 15 ones as the base classes. We evaluate with three different novel/base splits from [36], named as split 1, split 2, and split 3. Each split contains 15 base categories with abundant data and 5 novel categories with K annotated instances for K = 1, 3, 10. Following [36, 32, 30], we use the mean average precision (mAP) at 0.5 IoU threshold as the evaluation metric and report the results on the official test set of VOC 2007. When using MS COCO, 20 out of 80 categories are reserved as novel classes, the rest 60 categories are used as base classes. The detection performance with COCO-style AP, AP50, and AP75 for K = 10 and 30 shots of novel categories are reported.

4.2 Implementation Details

Our baseline is TFA [32] combined with the positive sample selection strategy. TFA is the most basic and representative pretrain-finetune based method with

	novel AP		novel AP50		novel AP75	
Method/Shot	10	30	10	30	10	30
FR [17]	5.6	9.1	12.3	19	4.6	7.6
Meta R-CNN [36]	8.7	12.4	19.1	25.3	6.6	10.8
TFA [32]	10	13.7	-	-	9.3	13.4
MSPR [35]	9.8	14.1	17.9	25.4	9.7	14.2
CME [19]	15.1	16.9	24.6	28	16.4	17.8
Retentive R-CNN [8]	10.5	13.8	-	-	-	-
FSCN [21]	11.3	15.1	20.3	29.4	-	-
FSCE [30]	11.1	15.3	-	-	9.8	14.2
UPE [34]	11	15.6	-	-	10.7	15.7
QA-FewDet [12]	10.2	11.5	20.4	23.4	9.0	10.3
Meta faster-rcnn [13]	12.7	16.6	**25.7**	**31.8**	10.8	15.8
FADI [2]	12.2	16.1	22.7	29.1	11.9	15.8
Ours	**15.3**	**17.1**	24.9	28.4	**16.5**	**18.1**

Table 2. Few-shot object detection performance on MS COCO. **Black** indicate the state-of-the-art (SOTA).

a two-stage training pipeline. TFA can represent most of the pretrain-finetune based few-shot methods because they have the same training pipeline and the same model structure. In detail, we use Faster R-CNN [27] as our base detector and ResNet-101 [16] with a Feature Pyramid Network [22] as the backbone. All models are trained using SGD with a batch size of 4 and weight decay of 0.0001. A learning rate of 0.005 is used during all stages. For the three splits of PASCAL VOC, w_1 is 0.01, 0.05, and 0.01, w_2 is 0.005, 0.001, and 0.0001, and w_3 is 0.001, 0.00005, and 0.00005.

4.3 Comparison with State-of-the-art Methods

To verify the effectiveness of our method, several competitive few-shot object detection methods are compared. The results are shown in Table. 1 and Table. 2. Following [13, 19, 35, 2], we use a single run with the same training images to get the results of different shots.

Results on PASCAL VOC. Following [36, 32, 30], we provide the average AP50 of the novel classes on PASCAL VOC with three splits in Table. 1. Our method can outperform previous methods by a large margin in extremely low-shot settings (i.e. 1-shot). The effectiveness of our method is fully demonstrated.

Results on MS-COCO. We report the COCO-style AP, AP50, and AP75 of the 20 novel classes on MS-COCO in Table. 2. Our method sets new state-of-the-art for 10 and 30 shots, under the same testing protocol and metrics.

Components			split 1			split 2			split 3		
RLP	PRD	TTP	1	3	10	1	3	10	1	3	10
-	-	-	48.3	52.8	59.6	35.6	42.1	47	40.6	45.2	50.3
✓	-	-	50.6	53.6	**62.9**	36.6	45.3	47.4	41.8	46	51.1
-	✓	-	50.3	54.4	60.7	36	43.6	46.9	41.4	44.2	49.6
✓	✓	-	46.6	53.5	57.9	32.4	43.5	47	41.5	44.6	49.9
-	✓	✓	51.7	55.6	59.3	36.9	43.2	46.5	40.1	42	49.1
✓	-	✓	54.1	56.4	61.3	37.3	**45.5**	47.2	39.8	46.4	50.2
✓	✓	✓	**54.6**	**56.5**	61.4	**37.9**	44.4	**47.6**	**42.8**	**46.6**	**51.2**

Table 3. Components of our proposed training pipeline. RLP is training the model with relative location prediction. PRD is training the model with patch random drop. TTP is our three-stage training pipeline. When RLP and PRD are used together, a weighted summation loss is applied to associate these two losses. Removing PRD is using the baseline model for feature gudiance. Removing RLP means adopting the baseline model as the initialization in the third stage.

	split 1			split 2			split 3		
Method/Shot	1	3	10	1	3	10	1	3	10
Baseline	48.3	52.8	59.6	35.6	42.1	47	40.6	45.2	50.3
sum(PRD,RLP)	46.6	53.5	57.9	32.4	43.5	47	41.5	**46.8**	**52.4**
FG(PRD,RLP)	**54.6**	**56.5**	**61.4**	**37.9**	**44.4**	**47.6**	**42.8**	46.6	51.2

Table 4. The way of integrating self-supervised tasks: summation vs feature guidance. Sum is using a weighted summation loss. FG means feature guidance.

4.4 Ablation Study

Component Analysis To show the effectiveness of our method, we first make a detailed comparison with the baseline by adding the components of our method. As shown in Table. 3, each component can improve the performance of our baseline in most of settings on PASCAL VOC benchmark. To be specific, RLP improves baseline by 2.3, 0.8, 3.3 and 1, 3.2, 0.4 and 1.2, 0.8, 0.8 for K=1, 3, 10 on novel split1, split2, and split3. PRD improves baseline by 2, 1.6 1.1 and 0.4, 1.5, -0.1 and 0.8, -1, -0.6 for K=1, 3, 10 on novel split1, split2 and split3. If using PRD and RLP via a weighted summation loss, the performance become even worse than the baseline. By using our three-stage training pipeline, our method outperforms the baseline up to 6% and achieves better performance than the baseline with a single self-supervised task (RLP or PRD).

The Way of Integrating Self-supervised Tasks We compare our feature guidance to integrating self-supervised tasks by a weighted summation loss. The results are shown in Table. 4, in which using feature guidance is better than the weighted summation loss in almost all settings. In few-shot scenarios, due to the lack of training samples, optimizing the model via a weighted summation loss may cause conflict, since the objectives of distinct tasks are different and the relationship between them is complicated.

	split 1			split 2			split 3		
Method/Shot	1	3	10	1	3	10	1	3	10
Baseline	48.3	52.8	59.6	35.6	42.1	47	40.6	45.2	50.3
FG(RLP,PRD)	48.4	54.7	59.3	35.5	42.2	47.5	41	44.3	49.2
FG(PRD,RLP)	**54.6**	**56.5**	**61.4**	**37.9**	**44.4**	**47.6**	**42.8**	**46.6**	**51.2**

Table 5. The order of the two models pre-trained with PRD and RLP. FG means feature guidance. FG(RLP, PRD) is using backbone-R to guide the learning of backbone-P. FG(PRD, RLP) is using backbone-P to guide the learning of backbone-R.

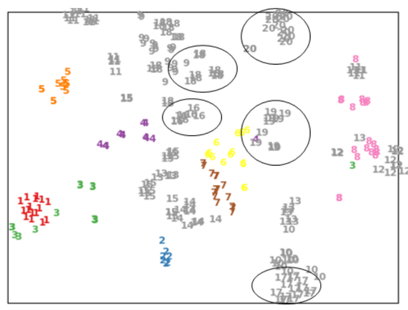
	split 1			split 2			split 3		
Method/Shot	1	3	10	1	3	10	1	3	10
Baseline	48.3	52.8	59.6	35.6	42.1	47	40.6	45.2	50.3
+Jigsaw	52.2	56.4	60.9	35.3	41.9	46.2	39.3	45.6	50.2
+PRD	50.3	54.4	60.7	36	43.6	46.9	41.4	44.2	49.6
FG(Jigsaw,RLP)	52.4	53.6	60.6	36.4	**44.6**	47	40.5	**47.1**	**51.5**
FG(PRD,RLP)	**54.6**	**56.5**	**61.4**	**37.9**	44.4	**47.6**	**42.8**	46.6	51.2

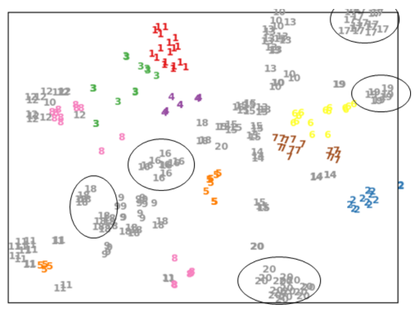
Table 6. PRD vs Jigsaw. FG means feature guidance. FG(Jigsaw, RLP) is using the backbone pre-trained with Jigsaw Puzzle to guide the learning of backbone-R.

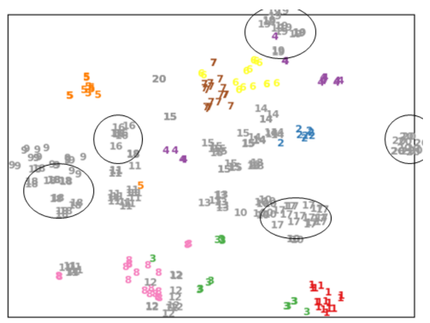
The Order of Feature Guidance In Fig. 2, we use backbone-P to guide the learning of backbone-R. If we use backbone-R to guide the learning of backbone-P, the results are shown in Table. 5, indicating the position of these two models is important. The results show that the model pre-trained with PRD can provide better guidance, and the model pre-trained with RLP is suitable for initialization.

Visualization Fig. 4 shows the visual embeddings of using one self-supervised method and combining multiple self-supervised methods via a weighted sum of the losses or our FG (feature guidance). In Fig. 4(a) and b, the distributions of category "18" and "9" are easy to be confused, while the boundaries of the categories pairs, including "18"|"9","16"|"18" and "17"|"10", are not clear in Fig. 4(c) demonstrates that simple combination of multiple self-supervised tasks is not a good choice for few-shot object detection. In Fig. 4(d), t-SNE [24] visualization of our method shows that our training pipeline with feature guidance can model the within-class similarity and build better classification boundaries, which demonstrate the effectiveness of our method.

Comparison with Other Self-supervised Methods We use Jigsaw Puzzle [25] to guide the learning of the third stage of our training pipeline. The results of using PRD is similar to Jigsaw Puzzle, which is shown in Table. 6. By using feature guidance, our PRD can achieve better performance than Jigsaw Puzzle in most of settings.


(a) t-SNE visualization from the model with PRD.


(b) t-SNE visualization from the model with RLP.


(c) t-SNE visualization from the model with a weighted sum of PRD and RLP loss.

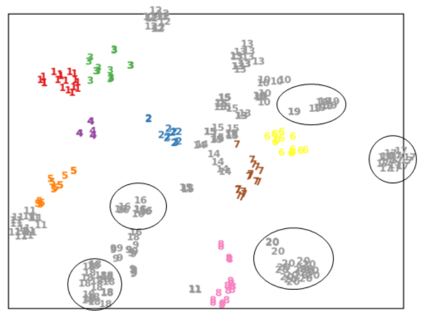

(d) t-SNE visualization from the model with our method.

Fig. 4. t-SNE visualization of the object proposal embeddings of cropped instances during the finetuning stage. The solid circles (16 to 20) denote novel categories during the fine-tuning stage. Others (1 to 15) are the base categories.

5 Conclusions

In this paper, we study the influence of self-supervised learning on Few-shot object detection. Our experimental results demonstrate that using a weighted sum of the self-supervised losses can not achieve better accuracy than using a single task in FSOD. To solve these problems, firstly, we introduce self-supervised learning into FSOD by applying SSL tasks to the cropped positive samples. Secondly, we propose a novel self-supervised method: patch random drop (PRD), for predicting the location of the masked image patch. Finally, we design a three-stage training pipeline to associate two self-supervised methods, including PRD and RLP. Experiments evaluate the effectiveness of our approach on the few-shot object detection datasets, i.e., Pascal VOC, MS COCO.

References

1. An, Y., Xue, H., Zhao, X., Zhang, L.: Conditional self-supervised learning for few-shot classification
2. Cao, Y., Wang, J., Jin, Y., Wu, T., Chen, K., Liu, Z., Lin, D.: Few-shot object detection via association and discrimination. Advances in Neural Information Processing Systems **34** (2021)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
4. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision. pp. 1422–1430 (2015)
5. Everingham, M., Eslami, S.A., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International journal of computer vision **111**(1), 98–136 (2015)
6. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International journal of computer vision **88**(2), 303–338 (2010)
7. 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)
8. Fan, Z., Ma, Y., Li, Z., Sun, J.: Generalized few-shot object detection without forgetting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4527–4536 (2021)
9. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8059–8068 (2019)
10. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
11. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., et al.: Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems **33**, 21271–21284 (2020)
12. 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)
13. 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. arXiv preprint arXiv:2104.07719 (2021)
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021)
15. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9729–9738 (2020)
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. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8420–8429 (2019)

18. 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)
19. 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)
20. Li, H., Tao, R., Li, J., Qin, H., Ding, Y., Wang, S., Liu, X.: Multi-pretext attention network for few-shot learning with self-supervision. In: 2021 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2021)
21. 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)
22. 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)
23. 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)
24. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
25. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
26. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7263–7271 (2017)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1137–1149 (2016)
28. Simard, N., Lagrange, G.: Improving few-shot learning with auxiliary self-supervised pretext tasks. arXiv preprint arXiv:2101.09825 (2021)
29. Su, J.C., Maji, S., Hariharan, B.: When does self-supervision improve few-shot learning? In: European Conference on Computer Vision. pp. 645–666. Springer (2020)
30. 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)
31. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29**, 3630–3638 (2016)
32. Wang, X., Huang, T.E., Darrell, T., Gonzalez, J.E., Yu, F.: Frustratingly simple few-shot object detection. arXiv preprint arXiv:2003.06957 (2020)
33. Wang, Y.X., Ramanan, D., Hebert, M.: Meta-learning to detect rare objects. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9925–9934 (2019)
34. 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)
35. 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)

36. 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 International Conference on Computer Vision. pp. 9577–9586 (2019)
37. Zhang, W., Wang, Y.X.: Hallucination improves few-shot object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13008–13017 (2021)