Meta-RCNN: Meta Learning for Few-Shot Object Detection

Xiongwei Wu
Singapore Management University
Singapore, Singapore
xwwu.2015@phdis.smu.edu.sg

Doyen Sahoo
Salesforce Research Asia
Singapore, Singapore
dsahoo@salesforce.com

Steven C.H. Hoi[1,2]
[1]Salesforce Research Asia
[2]Singapore Management University
Singapore, Singapore
chhoi@smu.edu.sg

ABSTRACT

Despite significant advances in deep learning based object detection in recent years, training effective detectors in a small data regime remains an open challenge. This is very important since labelling training data for object detection is often very expensive and time-consuming. In this paper, we investigate the problem of few-shot object detection, where a detector has access to only limited amounts of annotated data. Based on the meta-learning principle, we propose a new meta-learning framework for object detection named "Meta-RCNN", which learns the ability to perform few-shot detection via meta-learning. Specifically, Meta-RCNN learns an object detector in an episodic learning paradigm on the (meta) training data. This learning scheme helps acquire a prior which enables Meta-RCNN to do few-shot detection on novel tasks. Built on top of the popular Faster RCNN detector, in Meta-RCNN, both the Region Proposal Network (RPN) and the object classification branch are meta-learned. The meta-trained RPN learns to provide class-specific proposals, while the object classifier learns to do few-shot classification. The novel loss objectives and learning strategy of Meta-RCNN can be trained in an end-to-end manner. We demonstrate the effectiveness of Meta-RCNN in few-shot detection on three datasets (Pascal-VOC, ImageNet-LOC and MSCOCO) with promising results.

CCS CONCEPTS

• **Computing methodologies** → **Object detection**.

KEYWORDS

Object Detection, Deep Learning, Meta Learning

ACM Reference Format:
Xiongwei Wu, Doyen Sahoo, and Steven C.H. Hoi[1,2]. 2020. Meta-RCNN: Meta Learning for Few-Shot Object Detection. In *28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413832

1 INTRODUCTION

Object detection is the task of identifying various objects in a given image and localizing them with a bounding box, which is widely studied in computer vision. Following the success of deep learning for image classification [11, 12], recent years have witnessed remarkable progress in object detection with deep learning. A series of detection algorithms based on deep learning have been proposed which achieve state-of-the-art results on public detection benchmark datasets [7, 8, 14–16, 19, 21]. However, all these methods are data hungry, and require large amounts of annotated data to learn an immense number of parameters. For object detection, annotating the data is very expensive (far more than image classification), as it requires not only identifying the categorical labels for every object in the image, but also providing accurate localization information through bounding box coordinates. Moreover, in some applications, such as medical research, it's often impossible to even collect sufficient data to annotate. This warrants a need for effective detectors that can generalize well from small amounts of annotated data. We refer to the problem of learning detectors from limited labeled data as *few-shot detection*. For example, in *one-shot* detection, only one image is available with objects of interest annotated, and a detector needs to train on just this image and generalize. When presented with such small amounts of annotated data, traditional detectors tend to suffer from overfitting. Inspired by the fact that humans can learn a new concept from limited training data, we aim to develop a new few-shot detection method.

Recent years have seen active efforts for few-shot learning [6, 24, 26]. A promising direction among the emerging studies is to follow the principle of meta learning, where a set of tasks in a few-shot setting is simulated from a large corpus of annotated data, and the model is optimized to perform well over these few shot tasks. This trains the model to learn *how* to solve few-shot tasks. However, most existing efforts of meta learning are mainly focused on classification. Adapting few-shot classification algorithms directly for few-shot detection (e.g. by replacing the region classification branch of detector with a meta-learner) is non-trivial because of two major challenges. The first challenge is that the detection algorithms not only require classifying objects correctly, but also need to localize objects precisely in cluttered backgrounds using a Region Proposal Network (RPN) and bounding box (bbox) regressors. Thus, both RPN and bbox regressors should be capable enough to adapt to few-shot settings. The second challenge is that in a given few-shot task with one (or few) annotated image(s), the annotated image may contain objects from several classes, but only a few objects of interest are annotated. A few-shot detector should detect only the objects of interest. Unfortunately, a naively trained meta-detector's RPN would detect all objects (even objects from classes not of interest) and try to classify them as one of the classes of interest rather than background images (See Figure 1 for an example).

We aim to address these challenges in few-shot object detection by proposing a novel method using the meta-learning paradigm. In

MM '20, October 12–16, 2020, Seattle, WA, USA.

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413832

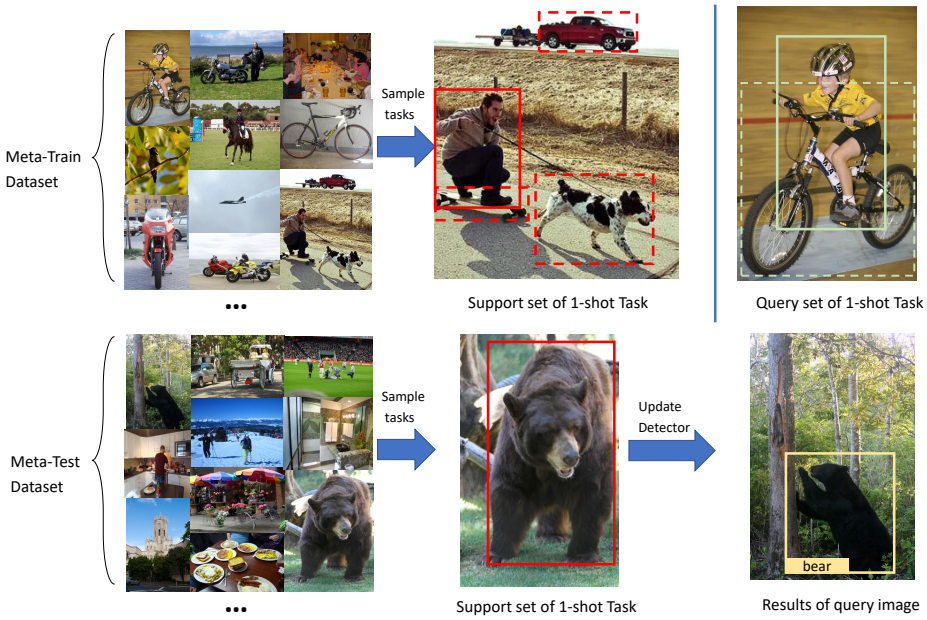

Figure 1: Few-shot object detection in the meta-learning setting. From the meta-train dataset, a Kway-Nshot support set and a query set are sampled to create a task. The meta detector makes predictions on the query set by using the prior knowledge learned from the support set, and updates the detector based on the training loss on the query set. In this example, despite many objects ("person", "dog", "truck", etc), the meta-train sample task aims to just detect "person". At test time, a single annotated image from a novel class (e.g., "bear") is available for the detector to learn a model that can generalize.

particular, we propose Meta-RCNN, an end to end trainable meta object detector, which follows the episodic learning paradigm of meta-learning [26], where multiple few-shot tasks are simulated based on a given meta-train dataset. Specifically, for a given task, we first construct a class prototype for each of the annotated object categories in the support set, where the prototype is a feature map representing the class. Using these prototypes, a class-specific feature map of the entire image is constructed, i.e., we obtain a class-specific feature map of the entire image for each of the class prototypes. This means that each feature map is tailored to detect only objects of the class of the prototype, by giving higher attention to appropriate regions in the image containing that object. This is followed by a Region-Proposal Network (RPN), weight-shared by all classes, to generate proposals. We aim to generate class-specific proposals by applying the RPN to class-specific feature maps. Each region proposal is then fed to a binary classifier (to determine if it is an object of the specific class or not) and a bounding box regressor to predict the location of the object.

Meta-RCNN learns a few-shot detector where the whole framework can be trained via meta-learning in an end-to-end manner. In contrast to the naive adaptation of meta-learning for classification into an object detection framework, Meta-RCNN meta-learns the few-shot classifier, the RPN, and the bbox regressor, thus making all three components suitable for handling few-shot scenarios. Moreover, Meta-RCNN learns a class-specific feature map for a given class prototype enabling easier distinction between classes of interest and backgrounds (where other objects in the image from classes not of interest are considered as backgrounds). We demonstrate the effectiveness of Meta-RCNN on two few-shot detection benchmarks: Pascal-VOC and ImageNet-LOC, and achieve promising results under different few-shot settings. Our key contributions in this work include:

- We propose a novel meta-learning based framework for few-shot object detection tasks, which is able to learn the ability to perform few-shot detection (both proposal generation and object classification) in an episodic learning paradigm.
- Based on our meta-learning framework, we propose a new few-shot object detection algorithm "Meta-RCNN" that extends the popular Faster-RCNN under the few-shot setting.
- We evaluate the performance of Meta-RCNN on two few-shot object detection benchmarks and the promising results validate the effectiveness of the proposed method.

The rest is organized as follow. Section 2 introduces some preliminaries and overviews object detection tasks under few-shot detection settings. Section 3 presents the proposed Meta-RCNN method. Section 4 discusses our experimental results. Section 5 reviews related work in object detection, meta learning and few-shot object detection. Finally Section 6 concludes this work.

2 PRELIMINARIES

2.1 Problem Setting

We present the formal problem setting of a few-shot object detection task in this paper. Consider two object detection datasets $\mathcal{D}$ and $\mathcal{T}$. $\mathcal{D}$ is a large-scale image dataset with annotated objects from $|C_D|$ categories, and $\mathcal{T}$ is a small target dataset with images that contains annotated objects from $|C_T|$ target categories. There is no category overlap between the two datasets: $C_D \cap C_T = \phi$. Our goal is to learn an object detector from the annotated data from $\mathcal{D}$ and $\mathcal{T}$

to detect the target categories of objects for any unlabeled/unseen image in $\mathcal{T}$. When the number of annotated objects in $\mathcal{T}$ is very small (e.g. 1 object is annotated per category), it becomes a few-shot detection task. In this paper, we investigate how to address this problem setting, by training a model that acquires the ability to quickly adapt to a novel few-shot detection task

We adopt the meta learning principle for few-shot object detection tasks. Meta-learning, also known as "learning to learn", aims to devise models that can learn new abilities or adapt to new environments rapidly with a few training examples. We follow the standard paradigm of meta learning with two stages: meta-training and meta-test. During the meta-training stage, the model is optimized through sampling mini-batches called "episodes" for training, where each episode is created to mimic the few-shot learning task by subsampling both categories and samples.

Specifically, during meta-training, few-shot detection tasks are sampled from the annotated dataset $\mathcal{D}$, in which each task consists of a support set S and a query set Q, where the support set mimics the few-shot annotated training data, and the query set mimics the test data for this task. More formally, consider a few-shot simulation with an K-Way and N-shot setting, for the i-th task, a support set S_i^D is created by randomly sampling a subset of images from $\mathcal{D}$ with K ways (namely K categories from C_D) and N shots (namely N images per category). During the same episode, a query set Q_i^D is created by randomly sampling a subset of images from $\mathcal{D}$ with the same K ways (namely the same K categories from C_D) but different N_Q shots (different N_Q images per category). The pair of S_i^D and Q_i^D is used as a few-shot task sample for training the model:

$$\mathrm{D}_i = \left\{S_i^D, Q_i^D\right\} \subset \mathcal{D} \tag{1}$$

During the meta-test stage, following the similar K-way and N-shot sampling approach, a pair of query and support subsets can be sampled from $\mathcal{T}$ for performance evaluation:

$$\mathrm{T}_i = \left\{S_i^T, Q_i^T\right\} \subset \mathcal{T} \tag{2}$$

where S_i^T is a support set and Q_i^T is a query set that serves as the ground-truth test set for evaluation. After adapting the model learned from the meta-training stage rapidly on the support set S_i^T, we can test the performance of the resulting model by evaluating the prediction results on the query set Q_i^T. These performance evaluation results are averaged across multiple few-shot tasks to evaluate the expected performance of the meta-trained few-shot detector over a variety of novel few-shot detection tasks.

2.2 Overview of Faster RCNN

In this paper, we extend the popular Faster RCNN algorithm [21] as our base model for few-shot detection tasks.

Faster RCNN consists of two components, an RPN (Region Proposal Network) for proposal generation and Fast RCNN for region classification as well as bounding box regression. RPN generates a sparse set of proposals from an input image. In particular, RPN extracts a feature vector from each region by scanning the whole image using sliding windows, followed by a binary classifier (objects vs backgrounds) and a bounding box regressor, where easy negatives are filtered. For each proposal, a fixed-length feature vector is extracted using ROI Pooling layers, which is then fed into a sequence of dense connected layers branching into two outputs: 1) classification: softmax probability over $K+1$ classes (K target classes plus a background class), and 2) regression: four real-values for refining bounding box position.

Specifically, we denote by u the category label, v the ground truth bounding box, p the predicted probability distribution over C classes, t_u the predicted bounding box prediction of class u, and λ as the trade-off parameter. L_{cls} represents softmax loss and L_{loc} represents SmoothL1 loss function for localization. The entire network can be optimized end-to-end by minimizing loss $L(p, u, t^u, v)$:

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v), \tag{3}$$

However, this framework requires a lot of training samples to obtain a good performance. In the next section, we present the proposed Meta-RCNN which builds over Faster RCNN and is specifically designed to address few-shot detection.

3 META-RCNN

3.1 Overview

We now present our proposed method Meta-RCNN for few-shot detection (See Figure 2 for an overview). Meta-RCNN is trained with multiple few-shot tasks simulated from the meta-train dataset. For each episode, a few object categories of interest are assumed to be annotated (Support set). During meta-training, a prototype is computed for each object category. The prototype is simply the mean vectors of features of all the annotated objects from that category. For a given query image, for each of these category prototypes, a class-specific feature map is generated by using a class-attention module which combines the prototype information with the feature map of the entire image. This feature map only highlights the signals of the class of interest, and suppresses information from other classes. This is followed by an RPN, which aims to generate proposals from the class-specific feature maps. This is followed by a binary classifier to predict if the object belongs to the specific category or not, and a bounding box prediction. Based on these predictions and the ground truth of the query images, a loss is computed and is used to update the model.

Meta-RCNN is a general paradigm to train few-shot detector via meta-learning. For each task, irrelevant categories and background can be filtered by an attention module, and the final generated feature map learns a general representation for the given few-shot detection task. Compared with [22] and [28], our Meta-RCNN is more general as the whole framework can be optimized end-to-end under the meta-learning paradigm, including RPN, the classifier and the bounding box regressor. Since all the components are meta-trained, each of them is suitable for few-shot learning. Next, we present the details of the model.

3.2 Meta-Training

Following the Meta-Training paradigm introduced previously, multiple Kway-Nshot tasks are simulated from the large annotated dataset $\mathcal{D}$. To fit the memory size during the Meta-Training stage, we meta-train the model by adopting the setting of 2way-5shot tasks, in which 10 support images are sampled for S_i^D (1 support image per class) and 1 image sampled to form the query set Q_i^D, which results in only a total of 11 images for each task D_i.

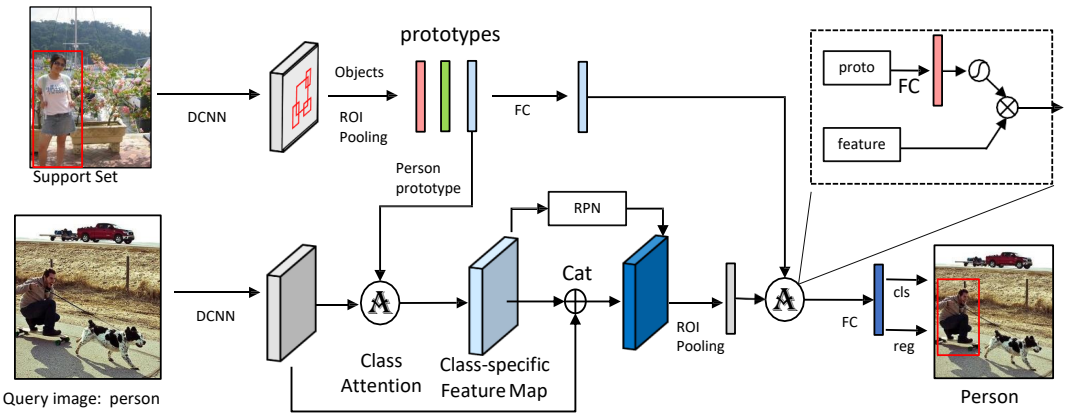

Figure 2: The Meta-RCNN workflow. We extract a set of prototypes ("prior") of different categories from the support set. For each class, conditioned on these prototypes, a class-specific feature map from the query set is generated by applying the class-aware attention module to the feature map of the entire image. The new class-specific feature map is tailored to detect objects of that specific class (e.g., class 'person' in the example). A weight-shared RPN is applied on the class-specific feature map, followed with a binary region classification layer and a bounding box regressor. The whole network is trained end-to-end.

3.2.1 Learning Prior from Support Set via Prototypical Networks. For meta-learning, a critical step is to learn prior from the support set, which can facilitate the subsequent training on the query set. In our approach, we adapt the idea of Prototypical Networks (PN) [24] to learn prior (namely "prototypes") from the support set.

Specifically, consider an episodic task D_i, the images from the support set S_i^D are fed into the Faster RCNN to generate the region features of objects belonging to the K selected categories. For each class c_k from the K object categories of interest, a prototype P_k is computed as the mean vector of the corresponding region features of objects belonging to the class c_k, namely

$$P_k = \frac{1}{N_k} \sum_{j=1}^{N_k} r_c(x_j) \tag{4}$$

where P_k denotes prototype of class c_k, $r_c(x_j)$ denotes the j-th region features of an object x_j from class c_k, and N_k denotes the total number of objects belonging to c_k in current support set.

3.2.2 Class-aware Attention Map. Based on the above prototypes, the images from the query set Q_i^D are fed into the same Faster RCNN model to obtain the image feature map before RPN and ROI Pooling. For each category, a class-specific feature map is learned based on the input query image and its corresponding prototype. We employ a learnable class-aware attention module, with the aim to highlight the signals of target class and suppress the signals of other categories. The class-aware attention module is based on basic channel-wise multiplication. The prototype P_k for a class c_k is encoded by a Fully-Connected (FC) layer ϕ, which is later combined with feature map f by element-wise multiplication:

$$F_k = f \odot \phi(P_k) \tag{5}$$

Thus, one new feature map F_k is generated for each category c_k to highlight the objects belonging to the class c_k. Based on the new feature map F_k, a weight-shared RPN is followed to produce region proposals. In order to recover the information lost in the class attention module, we generate a new feature map by concatenating original map with class-specific map and finally learn the region features on the new feature map by cropping the region features. To further enhance the representation of region proposals, we attach the prototype with the region feature r by element-wise multiplication:

$$R_c = r \odot \phi(P_k) \tag{6}$$

Furthermore, we apply margin loss between positive and negative samples to enhance the feature representation.

3.2.3 Overall Training Loss on the Query Set. Finally, we need to train the detector on the query set based on the class-aware feature maps learned from the support set. Specifically, a binary region classifier and a bbox regressor together with the RPN are jointly optimized w.r.t the query set Q_i^D as follows:

$$L(Q_i^D; S_i^D, \theta) = \sum_{(x,y,b) \in Q_i^D} L_{\text{loc}}\Big(\hat{b}(x), b\Big) + L_{\text{cls}}\Big(\hat{y}(x), y\Big) + L_{\text{RPN}} \tag{7}$$

where (x, y, b) denotes the object features, class label and bounding box respectively, and θ represents the parameters of Meta-RCNN. Notably, the number of negative training samples increases significantly in our framework, and thus we use hard negative mining strategy to train the model.

3.3 Meta-Test

During the meta-test stage, we have two settings: (i) multiple episodic prediction; (ii) single episodic prediction.

For multiple episodic prediction, we sample multiple few-shot detection tasks T_i from $\mathcal{T}$ for performance evaluation. In particular, each meta-test task T_i consists of a support set S_i^T for fast adaption/training, and a query set Q_i^T for test/evaluation. About the fast adaption for each meta-test task T_i, we first compute the set of prototypes from the support set S_i^T, and use these prototypes

Dataset	Train	#Img	#cls	Test	#Img	#cls
FSOD-VOC	VOC2007trainval	~ 3.8k	10	VOC2007test	~ 2.5k	10
FSOD-ImageNet	ImageNet-LOC	~ 53k	100	ImageNet-LOC	~ 117k	214
FSOD-COCO	COCO2017train-60	~ 115k	60	COCO2017val-20	~ 5k	20

Table 1: Summary of two Few-Shot Object Detection (FSOD) benchmark testbeds in our experiments

to generate class-specific feature maps. We then perform a light finetuning of the detection model based on the labeled images on the support set. Finally, we can evaluate the meta-detector's prediction outputs on the query set Q_i^T as a traditional object detection problem:

$$(p, u) = \text{Meta-RCNN}(Q_i^T; S_i^T, \theta) \tag{8}$$

where p is class probability vector and u is the location set of bounding boxes predicted by Meta-RCNN on the query set.

For single episodic prediction, we only evaluate the model on the single, fixed test set to directly compare with other benchmarks.

4 EXPERIMENTS

4.1 Datasets

We construct three benchmark testbeds to facilitate the performance evaluation of few-shot object detection (FSOD) in meta-learning settings: (i) FSOD-VOC based on Pascal VOC2007; (ii) FSOD-ImageNet based on the animal subset of ImageNet-LOC dataset; and (iii) FSOD-COCO based on MSCOCO. Table 1 gives a summary of the datasets. Pascal VOC2007 has 20 categories with 5k images in trainval set and 5k images in test set. A subset of 10 categories are randomly selected from VOC2007 trainval set for Meta-Training and the remaining 10-category subset of VOC2007 test set is used for Meta-Test. Images without target object categories are excluded in Meta-Test. For FSOD-ImageNet benchmark, we use the subset of first 100 animal classes of ImageNet in Meta-Training stage and the subset of remaining 214 animal species in ImageNet-LOC in Meta-Test stage. For FSOD-COCO benchmark, we use the 20 categories set in Pascal VOC in Meta-Test stage and the subset of remaining 60 categories in MSCOCO in Meta-Test stage. The backbone used in FSOD-VOC and FSOD-COCO benchmark is pre-trained on ImageNet. In FSOD-ImageNet benchmark, following the same setting as RepMet [22], we adopt the weights of Faster R-CNN [21] pretrained on MSCOCO dataset as backbone. Notably, there is no class overlap between MSCOCO set and FSOD-ImageNet test set. In our experiments, we conduct two types of evaluation: multiple episodic task and single episode task. For multiple episodic task, we evaluated our model on multiple tasks which are randomly sampled from test set, while for single episodic task, we finetune and evaluate the model on a fixed single few-shot dataset in Meta-Test stage (denoted as traditional settings).

4.2 Experimental Setups

4.2.1 **Task Generation.** For each benchmark, Meta-RCNN is evaluated on multiple tasks with different Kway-Nshot few-shot settings (N annotated images per category). For FSOD-VOC benchmark, we have 3 few-shot settings to evaluate Meta-RCNN: 5way-1shot, 5way-3shot and 5way-5shot. In detection, a single image has more than one object, so here we define the meaning of shot as instance number, not image number. On FSOD-ImageNet benchmark, we mainly follow [2] and [22] with two settings: 50way-1shot and 50way-5shot.

4.2.2 **Meta-model Parameter Setting.** In Meta-Training stage, we totally finetune the model for 8k and 60k iteration in FSOD-VOC benchmark and FSOD-ImageNet benchmark respectively. There is 1 image per class in query set to update the model weights. The initial learning rate is set to 1e-3 and is reduced to 1e-4 every 4k/30k iterations. We set the batch size of query as 4 during update.

4.2.3 **Basic Detection Parameter Settings.** The parameter settings for Meta-RCNN are identical to vanilla Faster RCNN. Proposals overlap with objects higher than 0.5 are considered positive and less than 0.3 are negative. During Meta-Training the top 128 most confident proposals are selected for training, and 300 proposals with highest confidence score are selected during evaluation. We build our Meta-RCNN based on Faster RCNN with VGG16 [23] and ResNet [9] model which are pretrained on ImageNet.

4.2.4 **Model Evaluation.** We evaluate Meta-RCNN based on multiple tasks of few-shot settings, which follows the evaluation metric of standard meta learning settings. Specifically, during the meta-test evaluation stage, a set of 200 Kway-Nshot tasks are sampled from the meta-test dataset, and only the images from the query set in each task will be evaluated. The mean Average Precision (mAP) over the selected K categories is used as the performance evaluation score.

4.3 Results on FSOD-VOC Benchmark

We evaluate our Meta-RCNN on FSOD-VOC benchmark where a subset of 10 Pascal VOC categories are selected for Meta-Training and another 10 categories are used for Meta-Test. For a fair comparison, these two subsets are split as similar as possible. For example, we keep animal categories on both sides since they share similar semantics information.

We compare with the proposed Meta-RCNN method on FSOD-VOC by implementing the following three baselines:

- **vanilla FRCN** [21]: the vanilla Faster RCNN which is the most popular object detection algorithm with competitive performance on many benchmarks. The vanilla FRCN is not designed for few-shot detection problem, but we try to include this baseline by fine-tuning the detector on the few-shot training data.
- **LSTD** [2] is a few-shot detection algorithm based on Faster RCNN. LSTD uses categorical regularization to transfer knowledge from L to S.
- **FRCN-PN** is a simple baseline for few-shot object detection using meta learning, which combines Faster RCNN and Prototype Networks [24]. Specifically, it replaces the final FC

Method	Backbone	5way-1shot	5way-3shot	5way-5shot
vanilla FRCN [21]	VGG16	14.78%	20.34%	26.89%
LSTD [2]	VGG16	17.66%	22.37%	29.00%
FRCN-PN	VGG16	12.71%	13.91%	14.33%
FRCN-PN (Finetuned.)	VGG16	16.48%	21.51%	26.01%
Meta-RCNN (ours)	VGG16	**19.03**%	**24.51** %	**31.23** %

Table 2: mAP Performance Evaluation on the FSOD-VOC benchmark

layer of Faster RCNN by the non-parametric prototypical networks.

All the above baselines including the proposed Meta-RCNN are based on VGG16 [23]. For regular FRCN and LSTD, we first train a global Faster RCNN during Meta-Training, and then the pretrained detectors are adapted to different tasks during Meta-Test. During Meta-Test, Meta-RCNN and vanilla FRCN are finetuned over 4 epochs while LSTD requires longer finetuning period (10 epochs). For FRCN-PN, prototypes of different categories are extracted as Meta-RCNN, and metric distances are learned to assign correct labels to each proposal. For fair comparison, we also add one baseline of finetuning FRCN-PN in Meta-Test stage, where the images of support set are also used as query images. Table 2 shows the results on three settings.

From Table 2, the performances of all four methods improve when the number of training shots increases. Notably, FRCN-PN obtains less improvement when the number of shots increases, primarily because the non-parametric classifier of PN limits its learning capacity from the increased training samples. By contrast, benefit from the finetuning operation as well as the FC layer in final classification and regression, our Meta-RCNN can still maintain consistent improvement when more training samples are available. Furthermore, it is interesting to see that the vanilla FRCN outperforms FRCN-PN even in very few-shot cases (5way-1shot) if FRCN-PN is not finetuned, where non-parametric property does not help PN obtain better performance. We argue this is because few-shot detection is generally more challenging than few-shot classification, as we discussed in introduction section. FRCN-PN cannot learn a representative prototype of background classes and the whole framework cannot be optimized by meta learning style (e.g., RPN and bbox regressors). The failure of FRCN-PN indicates naively attach components from few-shot classification framework cannot solve few-shot detection problem. Finally, our Meta-RCNN achieves better results than all the baselines, which validates the effectiveness of our method.

4.3.1 **Ablation study of RPN**. Here, we analyze the performance of RPN to validate our concerns of negative impact of irrelevant categories. We use vanilla FRCN and FRCN-PN as our baselines. The models are optimized in the same manner as before, but during Meta-Test phase, we evaluate the average recall on each task instead of mAP performance.

As observed from the results in Table 3, the vanilla FRCN significantly outperforms FRCN-PN that combines Faster RCNN and Prototypical Network. This is because the objects of irrelevant categories in the same image hurt the training process of RPN. By contrast, our Meta-RCNN outperforms these two baselines significantly. This is because our Meta-RCNN learns a general feature map for all *K*way-*N*shot detection tasks and optimizes RPN by meta learning, which proves to be more effective in few-shot settings. Notably, the results are surprising since the recall of RPN in few-shot scenario is significantly lower (> 90% with enough training data on VOC dataset).

Model	Backbone	5w-1s	5-3s	5w-5s
vanilla FRCN	VGG16	24.9%	26.5%	28.4%
FRCN-PN	VGG16	24.7%	24.9%	26.1%
Meta-RCNN (ours)	VGG16	**26.8**%	**29.1**%	**34.9**%

Table 3: Recall evaluation of Meta-RCNN on FSOD-VOC benchmark test set. For brevity, "5way-1shot" is abbreviated as "5w-1s".

4.4 Results on FSOD-ImageNet Benchmark

On FSOD-ImageNet benchmark, we adapt the weights of a network pretrained on MSCOCO trainval set, and then optimize Meta-RCNN based on this initialized model (which is similar to the other baselines). The Meta-RCNN is evaluated on the animal subset of ImageNet-LOC, which only contains single animal category per image and thus there are no irrelevant classes during training. This is simpler than the situation we discussed. In addition to FRCN and LSTD, we also include another recent baseline **RepMet** [22], which replaces FC classification layers in FRCN with more careful design of PN layers (learning multiple prototypes per class etc.), as well as much stronger backbone (DCN [3] and FPN [14]). Table 4 shows the results, in which our Meta-RCNN outperforms the other methods.

Model	Backbone	50w-1s	50w-5s
vanilla FRCN [21]	VGG16	16.5%	34.3%
LSTD [2]	VGG16	19.2%	37.4%
RepMet [22]	DCN+FPN	24.1%	39.6%
Meta-RCNN (ours)	ResNet101	**25.3**%	**40.6**%

Table 4: mAP performance evaluation on FSOD-ImageNet benchmark. Here "50way-1shot" is abbreviated as "50w-1s".

4.5 Results on Traditional VOC Bencmark

We compare our model on VOC dataset in the same manner (1 task) as several baselines in literature. We follow the experiment settings as [10]. We first train on a large annotated dataset, and then finetune on a single few-shot dataset. We use VOC2007 and VOC2012 trainval for training, and VOC2007 test set for testing. During training, we use 15 categories for large annotated dataset and 5 categories for few-shot dataset. We report the results in Table 5, in which our Meta-RCNN surpasses all the competitors on the same benchmark, including the existing SOTA method FSOD [5].

Model	Backbone	1-shot	2-shot	3-shot	5-shot	10-shot
YOLO-joint	ResNet-101	0.0	0.0	1.8	1.8	1.8
YOLO-ft	ResNet-101	3.2	6.5	6.4	7.5	12.3
YOLO-ft-full	ResNet-101	6.6	10.7	12.5	24.8	38.6
FRCN+joint	ResNet-101	2.7	3.1	4.3	11.8	29.0
FRCN+ft	ResNet-101	11.9	16.4	29.0	36.9	36.9
FRCN+ft-full	ResNet-101	13.8	19.6	32.8	41.5	45.6
LSTD [2]	ResNet-101	8.2	11.0	12.4	29.1	38.5
MetaYolo [10]	ResNet-101	14.8	15.5	26.7	33.9	47.2
MetaDet-YOLO [27]	VGG16	17.1	19.1	28.9	35.0	48.8
MetaDet-FRCN [27]	VGG16	18.9	20.6	30.2	36.8	49.6
MetaR-CNN [28]	ResNet-101	19.9	25.5	35.0	45.7	51.5
FSOD [5]	ResNet-101	31.7	32.0	33.4	41.6	50.0
Meta-RCNN (ours)	ResNet-101	**31.9**	**33.7**	**35.9**	**46.3**	**53.1**

Table 5: mAP performance on Pascal VOC benchmark. All the models are evaluated with 5 ways on VOC2007 test set. For MetaDet only VGG16 results are available. Notably, for 1-shot and 2-shot settings, we do not finetune our Meta-RCNN model to avoid overfitting.

4.6 Results on Traditional COCO benchmarks

In this section, we report the few-shot detector benchmark evaluation results on MSCOCO datasets. The model is trained with ResNet-50, and we resize the shorter size of the image into 800 pixels, with longer size no more than 1333 pixels. The support image is resized into 320x320. We use 20 categories in VOC dataset for testing and the left 60 categories for training. We finetune the model for 120k iterations in meta-training stage and 3k iterations in Meta-Test stage. The initial learning rate is set to 0.001 and will decays 10 times by every 3 epochs. Finally we evaluate our model in MSCOCOval set (5k images) with two settings: 10 shots and 30 shots. Table 6 shows the evaluation results. As observed from Table 6, our Meta-RCNN method outperforms all the existing methods including some state-of-the-art meta-learning approaches with substantial margins.

4.7 Visualization

Here we visualize some results from Pascal VOC in Figure 3. Notably, the model is adopted with only 1-object per class **without** any further finetuning on novel classes.

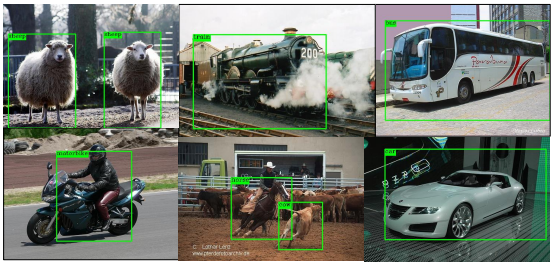

Figure 3: Our model can correctly detect novel objects even with 1-shot object. However, finetuning is required to deal with more complicated scenarios.

4.8 Discussions

Extension to other Meta-Learning Methods: Beyond prototypical networks, other meta-learning methods such as MAML [6] in principle can also be applied, e.g., we can apply MAML for vanilla FRCN framework, which updates the base model with the average gradient step of multiple tasks. However, in our experiments, the training process of MAML was unstable. This may be because few-shot detection is generally more difficult than few-shot classification, due to multiple dependent loss objectives (FRCN relies on RPN and regression loss etc.) and more complicated noisy contexts. In future, we plan to explore extensions to other meta-learning methods.

5 RELATED WORK

5.1 Generic Object Detection

Object detection based on deep learning can be broadly divided into two families: two-stage detectors and one-stage detectors. Two-stage detectors such as RCNN [8], Fast RCNN [7] and Faster RCNN [21], first generate a sparse set of proposal candidates, and a fixed-length feature vector is extracted from each of these candidates, followed by a categorical classifier and a bounding box regressor. Two-stage detectors have achieved state-of-the-art results on many public benchmarks [9, 14], but are often slower than one-stage detectors. One-stage detectors such as SSD [16], YOLO [18, 19] and RefineDet [29] directly generate categorical proposals from the feature map and thus avoid cascaded region classifiers. One-stage detectors can achieve real-time inference speed but the detection accuracy is often inferior to two-stage detection algorithms. Both detection families assume access to a large set of annotated data, and are not suitable for scenarios where the model has access to small amounts of annotated training data. In contrast, our proposed Meta-RCNN method addresses detection in the few-shot setting, and achieves promising results.

5.2 Meta Learning for few-shot classification

Few-shot learning has been widely explored in image classification particularly through meta-learning. [17] optimized a base-model via an LSTM-based meta-learner which simulated traditional SGD optimization. [6] proposed MAML which learned a good model initialization which could adapt to a new task in few gradient step updates. Based on MAML, [13] proposed Meta-SGD which learned a set of learnable parameters to control gradient step of different tasks. Learning initialization is potentially a very general

Shot	Baselines	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
10	Meta-Yolo [10]	DarkNet-19	5.6	12.3	4.6	0.9	3.5	10.5
	FRCN+ft [28]	ResNet-50	1.3	4.2	0.4	0.4	0.9	2.1
	FRCN+ft-full [28]	ResNet-50	6.5	13.4	5.9	1.8	5.3	11.3
	MetaDet [27]	VGG16	7.1	14.6	6.1	1.0	4.1	12.2
	Meta R-CNN [28]	ResNet-50	8.7	19.1	6.6	2.3	7.7	14.0
	Meta-RCNN (Ours)	ResNet-50	**9.4**	**17.1**	**9.4**	**1.7**	**11.2**	**18.1**
30	Meta-Yolo [10]	DarkNet-19	9.1	19.0	7.6	0.8	4.9	16.8
	FRCN+ft [28]	ResNet-50	1.5	4.8	0.5	0.3	1.8	2.0
	FRCN+ft-full [28]	ResNet-50	11.1	21.6	10.3	2.9	8.8	18.9
	MetaDet [27]	VGG16	11.3	21.7	8.1	1.1	6.2	17.3
	Meta R-CNN [28]	ResNet-50	12.4	25.3	10.8	2.8	11.6	19.0
	Meta-RCNN (Ours)	ResNet-50	**12.8**	**25.5**	**12.2**	**2.3**	**12.3**	**19.3**

Table 6: Low-shot detection performance on COCO val set with 20-way novel classes. Two 20-way settings are evaluated: 10 shots and 30 shots per each novel class using the ResNet-50 backbone. For MetaDet only VGG16 results are available

idea for few-shot learning however, the training process can be unstable [1] especially in challenging tasks such as detection. [26] proposed Matching Networks based on a non-parametric principle by learning a differentiable K-Nearest Neighbour model, and [24] proposed Prototypical Networks using the similar principle. These have been extended to learning the distance metric for K-NN [25] and semi-supervised few-shot learning[20]. Directly adapting these techniques for object detection is not trivial, especially for the RPN, which generates proposals for all objects rather than proposals only for the objects of interest for the few-shot task.

5.3 Few-shot Object Detection

In contrast to classification, few-shot detection has received less attention. [4] addressed few-shot detection using large-scale unlabeled data. Their model is based on a semi-supervised method which extracts knowledge from unlabeled dataset to enrich training dataset by self-paced learning and multi-modal learning. However, their method may be misled by the incorrect predictions from initial model and also requires expensive re-training the model for every new task. [2] proposed a Low-shot Transfer Detector (LSTD) using regularization to transfer knowledge from source domain to target domain by minimizing the gap between the two domains.

Recently, in parallel with our work [1], there have been some concurrent efforts in applying meta-learning for object detection similar to our work. [22] proposed RepMet as a meta-learning based few-shot detector which replaces the fully connected classification layer of a standard detector with a prototypical network. However, it suffers from two critical limitations in that RPN and bounding box regression are not tailored for few-shot challenges (as they are not meta-trained), and it often fails in distinguishing object classes of interest from background (including object classes not of interest). [10] proposed Meta-YOLO by applying meta-learning with YOLO. They optimize the few-shot detector by re-weighting the channels of global features with support images. [27] proposed MetaDet as a meta-learning framework for few-shot detection. In MetaDet, they disentangle the learning process of class-agnostic parts and class-specific parts, and learn a meta model to predict class-specific parameters from few-shot data. Different from MetaDet which learns a meta model for parameter prediction, our model focuses on feature learning. We learn a meta model to calibrate the feature representation from few-shot data. [28] shares the same name as our model but has a very restrictive definition of the components that are meta-learned (only the region of interest features). Moreover, there are limitations with regard to this approaches ability to be easily extended to quickly adapt to novel few-shot tasks (e.g. it assumes that a given few-shot detection task will have objects from the annotated train dataset, and thus can wrongly predict novel objects as belonging to one of the categories from training data. In contrast to these approaches, in our model, all the components can be optimized in an end-to-end manner under the meta-learning paradigm, making each component few-shot capable, and this model can quickly adapt to any novel few-shot detection task.

Finally, we would like to compare with another Meta R-CNN [28] which was proposed in parallel with our work. Despite the similar names, there are a number of key differences. First of all, we re-formulate the original multi-class classification task into a set of binary classification tasks, which simplify the whole few-shot detection task. Second, the RPN in our Meta-RCNN is meta-trained jointly together with the classifiers and bbox regression, while the approach in [28] only applies meta-learning for classifiers and regressors, and their RPN is trained by a conventional way. Besides, the training procedures of the two methods are very different, including the model design and training data sampling, etc.

6 CONCLUSION

We investigated the problem of few-shot object detection in a meta-learning setting, and proposed a new few-shot object detection method named Meta-RCNN. Specifically, we propose to learn prior from the support set by borrowing the idea of prototypical networks, and then extend the popular Faster RCNN method by jointly training the RPN, the object classifier and the bounding box regressor together in a meta-learning framework. We propose a novel class-aware attention module to facilitate the meta-training. We conduct extensive experiments on multiple few-shot object detection benchmarks and obtain promising results. In future work, we plan to extend our framework by exploring more recent meta-learning techniques and evaluating diverse detectors.

[1] An early version of this work was completed and submitted to a conference in 2019

REFERENCES

[1] Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2018. How to train your MAML. *arXiv preprint arXiv:1810.09502* (2018).

[2] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. 2018. LSTD: A Low-Shot Transfer Detector for Object Detection. In *AAAI*.

[3] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable convolutional networks. In *ICCV*.

[4] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. 2018. Few-Example Object Detection with Model Communication. In *TPAMI*.

[5] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. 2020. Few-Shot Object Detection with Attention-RPN and Multi-Relation Detector. In *CVPR*.

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning, ICML*.

[7] Spyros Gidaris and Nikos Komodakis. 2015. Object detection via a multi-region and semantic segmentation-aware cnn model. In *ICCV*.

[8] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.

[10] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. 2019. Few-shot Object Detection via Feature Reweighting. In *ICCV*.

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *CVPR*.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NeurIPS*.

[13] Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835* (2017).

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *ICCV*.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. SSD: Single Shot MultiBox Detector. In *ECCV*.

[17] Sachin Ravi and Hugo Larochelle. 2016. Optimization as a model for few-shot learning. *ICLR* (2016).

[18] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-Time Object Detection. In *CVPR*.

[19] Joseph Redmon and Ali Farhadi. 2016. YOLO9000: Better, Faster, Stronger. In *arXiv preprint arXiv:1612.08242*.

[20] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676* (2018).

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*.

[22] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Sharathchandra Pankanti, Rogerio Feris, Abhishek Kumar, Raja Giries, and Alex M Bronstein. 2019. RepMet: Representative-based metric learning for classification and one-shot object detection. In *CVPR*.

[23] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *arXiv preprint arXiv:1409.1556*.

[24] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*.

[25] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in neural information processing systems*.

[27] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2019. Meta-Learning to Detect Rare Objects. In *ICCV*.

[28] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. 2019. Meta R-CNN : Towards General Solver for Instance-level Low-shot Learning. In *ICCV*.

[29] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z Li. 2018. Single-Shot Refinement Neural Network for Object Detection. In *CVPR*.