

Meta-Learning to Detect Rare Objects

Yu-Xiong Wang Deva Ramanan Martial Hebert
 Robotics Institute, Carnegie Mellon University
 {yuxi ongw, deva, hebert}@cs. cmu. edu

Abstract

Few-shot learning, i.e., learning novel concepts from few examples, is fundamental to practical visual recognition systems. While most of existing work has focused on few-shot classification, we make a step towards few-shot object detection, a more challenging yet under-explored task. We develop a conceptually simple but powerful meta-learning based framework that simultaneously tackles few-shot classification and few-shot localization in a unified, coherent way. This framework leverages meta-level knowledge about “model parameter generation” from base classes with abundant data to facilitate the generation of a detector for novel classes. Our key insight is to disentangle the learning of category-agnostic and category-specific components in a CNN based detection model. In particular, we introduce a weight prediction meta-model that enables predicting the parameters of category-specific components from few examples. We systematically benchmark the performance of modern detectors in the small-sample size regime. Experiments in a variety of realistic scenarios, including within-domain, cross-domain, and long-tailed settings, demonstrate the effectiveness and generality of our approach under different notions of novel classes.

1. Introduction

Deep convolutional neural networks (CNNs) have revolutionized the landscape of large-scale visual recognition [31, 26]. A key driving factor is human supervision in the form of large amounts of annotated images. However, in many practical applications such as self-driving vehicles, recognition systems need to rapidly recognize some *never-before-seen* objects from a very limited number of examples [16, 49, 3]. Simply collecting more data does not scale and quickly becomes expensive [74, 66, 71, 21, 38].

This challenge of learning novel concepts from few labeled examples is commonly addressed in *few-shot* or *low-shot learning*. Much of its recent progress comes from framing few-shot learning as a meta-learning problem [62]. By simulating and solving a variety of few-shot learning

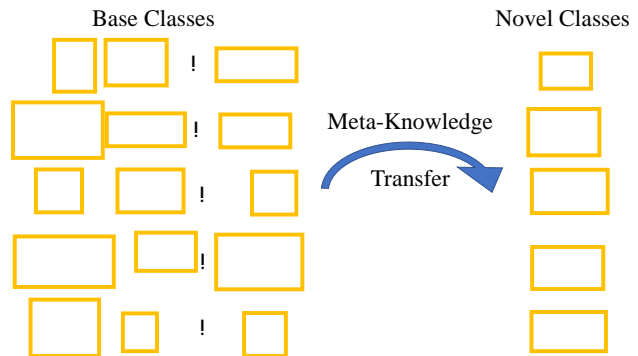


Figure 1: Few-shot object detection: learning an object detector for novel classes when only a few examples with annotated bounding boxes are available. Our meta-learning based framework leverages meta-level knowledge about “model parameter generation” from base classes with a large amount of annotated bounding box examples. Such knowledge is then transferred to guide the detector learning for novel classes in a sample-efficient way.

tasks from base categories with abundant labeled examples, meta-learning approaches acquire meta-level knowledge about *learning to learn* and then transfer such knowledge to tackle few-shot learning of novel categories. Despite notable successes, most of existing work has focused on simple image classification tasks with artificial settings, such as evaluating on small-scale datasets like mini-ImageNet and for contrived tasks like 5-way classification [67, 58].

In this paper, we make a step towards *few-shot object detection*, a more challenging task of practical importance that learns an object detector when only a few examples with annotated bounding boxes are available. Unlike image classification, detection requires not only classifying but also localizing (multiple) objects within an image, as well as dealing with distracting background regions [19]. Hence, prior few-shot learning approaches, developed with image classification in mind, are not readily applicable to our task. While there is some initial attempt in this direction as in [69, 6, 12, 29, 54], they simply introduce a few-shot classifier into a detection model *without* truly addressing few-shot localization, leading to suboptimal performance.

To address this limitation, we propose a meta-learning based framework that *simultaneously tackles few-shot clas-*

sification and few-shot localization in a unified, coherent way. This framework leverages meta-level knowledge about “model parameter generation” from base classes to facilitate producing a detector for novel classes from few examples (Fig. 1). *Our insight* is to disentangle the learning of category-agnostic and category-specific components in a modern CNN based detection model, and to exploit different meta-strategies for optimizing different components.

It is known that from bottom to top layers of a learned CNN, the model components make a transition from generic to specific and contribute differently to visual recognition [72]. For the low-level components (*e.g.*, bottom convolutional layers), their parameters are shared by many classes and thus category-agnostic. For the high-level components (*e.g.*, top fully-connected layers), their parameters tend to be category-specific. However, inspired by the recent meta-learning work [70, 1, 34, 57, 28], the way how these parameters change during the learning process (*i.e.*, their first order dynamics) might be shared by many classes. Such sharable properties with respect to the parameters or their dynamics can be identified when training a CNN detector on base classes, and then re-purposed to guide the detector learning for novel classes in a sample-efficient way.

Based on this insight, we design a few-shot detection model “MetaDet” built upon the widely-used detection approach Faster R-CNN [46]. Faster R-CNN relies on a region proposal network (RPN) to generate regions of interest (RoIs) on top of convolutional features, and finally uses two sibling branches to classify these RoIs into one of the object classes or background and regress to the refined bounding box positions. We treat the convolutional features and RPN as category-agnostic components, whose parameters are shared by base and novel classes. While the classifiers and bounding box regressors are category-specific, we view that the dynamics of their parameters are shared by base and novel classes. Concretely, inspired by [70], we introduce a parameterized *weight prediction meta-model* that is trained on the space of model parameters to predict a category’s large-sample bounding box detection parameters from its few-shot parameters. The meta-model is trained end-to-end with Faster R-CNN through a meta-learning procedure. While we focus on Faster R-CNN, our framework can be combined with other detectors like YOLO [43, 44] as well.

We explore our approach in a variety of realistic scenarios. We start by using the PASCAL VOC dataset [14, 13] to provide a means of establishing systematic quantitative evaluation of existing detection approaches in the small-sample size regime. We then compare with state-of-the-art approaches on the PASCAL VOC, MS-COCO [36], ImageNet [47], and iNaturalist [65] datasets, covering within-domain, cross-domain, and long-tailed settings. Importantly, these evaluations investigate *different notions* of “novel” classes that are useful in practice: (1) we have never

before seen novel classes, (2) we have seen only global image-level labels of novel classes, and (3) we might have seen novel classes as background without any labels.

Our contributions are three-fold. (1) We explore a challenging yet under-explored few-shot object detection problem and systematically benchmark modern detection models in the small-sample size regime. (2) We present a novel meta-learning based approach that disentangles the learning of category-agnostic and category-specific parameters in CNN based detectors. Our approach is simple, general, and jointly addresses few-shot classification and few-shot localization in a coherent way. (3) We show how our approach significantly facilitates the detection of novel classes from few examples in a variety of realistic scenarios, including within-domain, cross-domain, and long-tailed settings.

2. Related Work

Deep Learning based Object Detection. Modern object detection models focus on learning from abundant data to improve detection accuracy and speed. The R-CNN [19] and OverFeat [55] detectors initiate this success. Since then, flagship techniques are mainly represented by two types of detectors. A set of models are region-based, including R-CNN [19], Fast R-CNN [18], SPP-Net [25], Faster R-CNN [46], FPN [35], Mask R-CNN [24], and DCN [10]. Another family is proposal-free, including YOLO [43], YOLOv2 [44], YOLOv3 [45], and SSD [37]. We propose a general meta-learning based framework for few-shot detection and instantiate it with Faster R-CNN as well as YOLO.

Few-Shot Learning and Meta-Learning. Few-shot learning is a fundamental yet unsolved problem in machine learning and computer vision [61, 15, 30, 32, 67, 23, 20, 64]. Most of existing work is developed in the context of classification, which cannot directly apply to other tasks. Our approach falls under the umbrella of meta-learning [62, 63, 51, 52, 1, 50, 67, 70, 4, 42, 17, 58, 71, 22, 40, 34, 60, 28, 68, 53, 48, 8, 33], and is most related to the work on model parameter estimation [70]. We extend [70] from classification to detection scenarios by designing a meta-model that predicts the parameters of category-specific components for a detection model in the small-sample size regime.

Object Detection with Limited Supervision. Our few-shot detection task is an under-explored problem and recently, there has been some initial attempt in this direction [6, 54, 29]. A regularized fine-tuning method is proposed [6] to transfer a pre-trained detector to the few-shot task. In [54], distance metric learning is exploited to model the multi-modal distribution of each class for object detection. A meta-model is introduced to adjust pre-trained features to detect novel classes [29]. Our approach is different from the prior work in three important ways. (1) Most of the work [54, 29] simply transforms a few-shot classifier into a detector, while we simultaneously address few-shot classi-

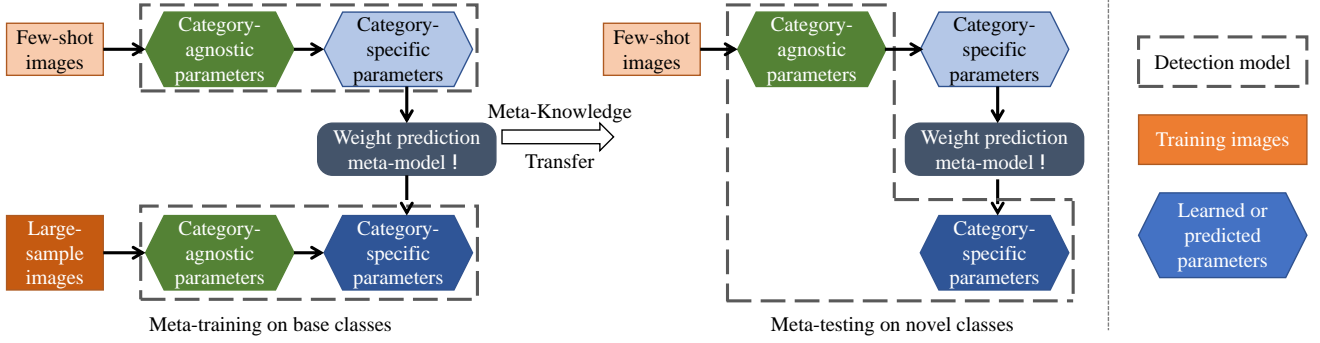


Figure 2: Our meta-learning based few-shot detection approach “MetaDet”. We use different meta-strategies to disentangle the learning of category-agnostic and category-specific parameters in a CNN based detector (*e.g.*, Faster R-CNN). Learning base detector and meta-model during meta-training (left): we train the large-sample base detector and obtain the category-agnostic parameters; we sample few-shot base detection tasks, and learn a *weight prediction meta-model* T that transforms from few-shot to large-sample category-specific parameters. Learning novel detector during meta-testing (right): we initialize the category-agnostic parameters from the base detector and use T to predict category-specific parameters. Light color denotes few-shot parameters; dark color denotes large-sample or predicted parameters.

fication and localization. (2) The existing work [6, 54, 29] either transfers knowledge from large-sample base set or meta-knowledge from few-shot base detection tasks, *but not both*. By contrast, we leverage both of them and show they are all useful for few-shot detection. (3) We systematically benchmark modern approaches for few-shot detection and we significantly outperform the prior work by large margins. There are other settings of detection with limited supervision, such as weakly-supervised detection [59, 5, 11], semi-supervised detection [69, 39, 12], and zero-shot detection [2, 41, 73], which are different from ours.

3. Meta-Learning based Object Detection

Figure 2 illustrates the framework of our meta-learning based few-shot object detection approach “MetaDet”. Through learning to learn from a large set of few-shot detection tasks, which are simulated on base classes with abundant annotated data, MetaDet allows us to rapidly generate a detector for novel classes using just a few labeled examples.

3.1. Meta-Learning Setup for Few-Shot Detection

We extend the widely-used setup for few-shot classification [67, 17] to establish few-shot detection. Specifically, we have a base category set C_{base} and a novel category set C_{novel} , in which $C_{\text{base}} \cap C_{\text{novel}} = \emptyset$. Correspondingly, we have a large-sample base dataset $S_{\text{base}} = \{(I_i, Y_i)\}$, where $\{I_i\}$ are input images, and $\{Y_i\}$ are the corresponding annotations indicating labels and bounding boxes for objects of base classes. In k -shot detection, we have a novel dataset $S_{\text{novel}} = \{(I_i, Y_i)\}$, in which each novel class has k bounding box annotations. We aim to learn a detection algorithm on S_{base} that is able to generalize to unseen categories C_{novel} . Here, we focus on the detection performance on C_{novel} , which is evaluated on a held-out test set.

Through meta-learning, we are interested in training a *learning procedure* (*i.e.*, meta-learner) that guides the gen-

eration of a detector (*i.e.*, learner) for a k -shot detection task. Meta-learning algorithms achieves this by explicitly mimicking the few-shot learning scenario and learning from a collection of k -shot detection tasks sampled from S_{base} . Each of these sampled tasks is termed as *an episode*. Meta-learning algorithms thus have two stages: meta-training on S_{base} and meta-testing on S_{novel} . During meta-training, we randomly sample k bounding box annotations per class on S_{base} , and train the corresponding detector. The meta-level knowledge regarding learning detectors across various detection tasks is at the same time aggregated into the meta-learner. During meta-testing, the base detector is adapted on S_{novel} for novel classes via the meta-learner.

3.2. Basic Detector and Meta-Strategies

The goal here is to estimate the parameters of a detector desired for C_{novel} based on both S_{base} and S_{novel} . For modern deep CNN based detectors, are composed of class-agnostic and class-specific components [72, 46, 44]. Our framework thus exploits meta-level knowledge about parameter generation of a detection model and leverages specialized meta-strategies for these different types of components. This parameter-oriented perspective also allows us to jointly address few-shot classification and few-shot localization in a unified, consistent manner.

Our meta-learning framework applies to a variety of CNN based detectors. Here we instantiate the framework with Faster R-CNN [46], because it is a simple detection model which achieves impressive performance and upon which the most recent detectors are built. Faster R-CNN consists of a region proposal network (RPN) for generating region proposals and a detection network (Fast R-CNN) [18] which uses these proposals to detect objects. A backbone convolutional network is shared by these two networks and provides convolutional feature maps. RPN identifies region proposals on the feature maps by predicting

the probability of an anchor (*i.e.*, reference box) being foreground or background and refining the anchor. These region proposals are then reshaped using a RoI pooling layer, fed into the detection network for predicting the object class via a softmax classifier and producing the bounding box offsets via per-class bounding box regressors.

Meta-strategy for category-agnostic components. We treat the convolutional network, RPN, and the bottom layers of the detection network as category-agnostic components, whose parameters are shared by both base and novel classes. In fact, in the design of Faster R-CNN [46], RPN is category-agnostic: its box-classification layer only assigns a binary class label (of being an object or not) to each anchor without differentiating the specific object classes, and its box-regression layer regresses from an anchor to a nearby ground-truth box without considering the class label of the ground-truth. Such a shared property enables us to transfer the category-agnostic parameters from the base to the novel detector or use them as initialization for fine-tuning.

Meta-strategy for category-specific components. However, we are still faced with the difficulty of learning the parameters of the category-specific components from few examples. In Faster R-CNN, the top layer of the detection network contains category-specific parameters that are used to perform bounding box classification and regression for each class. While these parameters are not directly transferable between base and novel classes, inspired by [70], the *dynamics pattern* of how they change from the parameters trained on a small dataset to those trained on a large dataset can be characterized by a *generic, category-agnostic transformation*. We introduce a parametrized weight prediction meta-model T to learn such a transformation through meta-training process.

3.3. Weight Prediction Meta-Model

For a given category c , let w_{det}^c denote the class-specific object detection weights in the last layer of the detection network learned from the large-sample base dataset S_{base} . Let w_{det}^c denote the corresponding weights learned from the k -shot episode dataset sampled from S_{base} . The weight prediction meta-model $T(\cdot)$ regresses from w_{det}^c to w_{det}^c in the model parameter space: $w_{det}^c = T(w_{det}^c; \cdot)$, where \cdot are category-agnostic, learned parameters.

Loss function. The *same* $T(\cdot)$ is applied to any class c . The meta-objective function for each class in an episode is:

$$\|T(w_{det}^c; \cdot) - w_{det}^c\|^2 + \sum_{(x,y) \in \text{RoI}^c} \text{loss}(D(x; T(w_{det}^c; \cdot)), y). \quad (1)$$

The final loss is minimized with respect to \cdot , which is averaged over all $c \in C_{base}$ and over w_{det}^c generated in all the episodes. ‘loss’ refers to the standard performance loss used to train the detection network D (*e.g.*, multi-task loss consisting of bounding box classification and regression losses), and RoI^c denotes the training RoIs labeled with

class c . $\lambda > 0$ is the regularization hyper-parameter used to control the trade-off between the two terms.

Note that the bounding box detection branch contains two types of detection weights: the RoI classification weights w_{cls}^c and the bounding box regression weights w_{loc}^c . Here we use the concatenation of the two types of weights, *i.e.*, $w_{dec}^c = [w_{cls}^c, w_{loc}^c]$. We thus simultaneously address few-shot classification and few-shot localization in a unified way, extending the sole classification in [70]. $T(\cdot)$ can be implemented as a small fully-connected neural network and jointly trained with the detector, as shown in Fig. 2.

3.4. Meta-Learning Procedure

The meta-learning procedure consists of two phases: meta-training on S_{train} and meta-testing on S_{novel} .

Stage-wise meta-training. We split the meta-training procedure into two stages for category-agnostic and category-specific components, respectively. During the first stage, we train a large-sample base detector $D(\cdot; \cdot)$ on the entire dataset S_{train} in the normal way [46]. This provides us with the basic detector which will be used for novel classes and the large-sample parameters w_{det}^c of the class-specific components.

During the second stage, we perform few-shot episode detection. In each episode, we randomly sample k bounding box annotations per class on S_{train} . We leverage the large-sample base detector $D(\cdot; \cdot)$ trained in the first stage to generate the k -shot detector as $D(\cdot; \cdot, \setminus w_{det}^c, w_{det}^c)$. That is, we freeze the category-agnostic parameters as those learned in the large-sample setting $\setminus w_{det}^c$ and retrain the category-specific parameters w_{det}^c from scratch. We use w_{det}^c, w_{det}^c together with the k -shot examples to train the meta-model $T(\cdot; \cdot)$ based on the meta-objective in Eqn. (1). Everything is trained end-to-end.

Meta-testing. To train the k -shot novel detector on S_{novel} , we initialize its category-agnostic parameters from the base detector as $\setminus w_{det}^c$, and we randomly initialize its category-specific parameters w_{det}^c . Following [70], we use the meta-model $T(\cdot; \cdot)$ to predict the desired w_{det}^c as biased regularization while fine-tuning the detector. During inference time, the meta-model is detached, and our detector is the standard (Faster R-CNN) detector.

4. Experimental Evaluation

We explore the use of our meta-learning based approach “MetaDet” for few-shot detection tasks. We begin with extensive evaluation on the PASCAL VOC dataset [14, 13], benchmarking the performance of CNN based detectors in the small-sample size regime and addressing variants and different design choices of MetaDet. We then evaluate on challenging, large-scale MS-COCO [36], ImageNet [47], and iNaturalist [65] datasets and compare with state-of-the-art approaches. We use Faster R-CNN [46] as the detec-

tor in the most of the experiments. We evaluate in a variety of scenarios, including within-domain, cross-domain, and long-tailed settings, and investigate different notions of novel classes, demonstrating the generality of our approach.

4.1. Within-Domain Few-Shot Detection

Following the setting commonly used in few-shot classification [67, 58], we first consider few-shot detection in the within-domain scenario. Here we treat each dataset as a domain. In this setting, a dataset is randomly split into two *non-overlapping* portions: one set of base classes with hundreds or thousands of examples per class and an additional set of novel classes with a small number of examples (k) per class. Here, we focus on the performance of novel classes. We construct similar settings for few-shot detection on the heavily benchmarked PASCAL VOC [14, 13] and MS-COCO [36] object detection datasets.

4.1.1 Evaluation and Analysis on PASCAL VOC

Dataset and task. PASCAL VOC [14, 13] contains 20 object categories. Following the standard protocol in [18, 46], training is performed on the union of VOC 2007 and 2012 trainval with a total of 16.5k images and evaluation is on 5k VOC 2007 test images. Consistent with [29], we divide the 20 categories randomly into 15 base classes and 5 novel classes. Our k-shot detection task is to learn a detector for the 5 novel classes from k annotated bounding box examples per class, where k is set to be 1, 2, 3, 5, and 10. During meta-training, we learn the base detector and the weight prediction meta-model using the trainval set of only the base classes. To do so, we sample randomly a collection of k-shot detection tasks on the base classes. During meta-testing, we adapt our base detector guided by the meta-model to generate the novel detector. We produce randomly 3 different sets of class splits. For each split, we run 5 trials and report the average detection performance (mAP).

Implementation details. Our detector is the standard Faster R-CNN [46], with VGG16 [56] as the backbone architecture pre-trained on the ImageNet-1k classification dataset [47]. The final bounding box detection parameters consist of a 4,096-d RoI classification parameter vector w_{cls}^c and a 4,096-d bounding box regression parameter vector w_{loc}^c for each class. We use a 3-layer fully-connected network with Leaky ReLU nonlinearity as our weight prediction meta-model T , taking as input the concatenation of w_{cls}^c and w_{loc}^c and predicting their estimated values. is cross-validated. Following [46], each mini-batch has one random training image. We train the detector and meta-model using SGD with a momentum of 0.9 and a weight decay of 0.0005. During meta-training, we first train the large-sample base detector on the full set of base classes for 6 epochs, with a learning rate of 0.001 which is decreased

by 10 at 5 epochs. We then sample 5,000 few-shot base detection tasks, and for each task we train the corresponding bounding box detection parameters with the remaining parameters frozen to those in the already-trained large-sample base detector. The meta-model is trained at the same time for 10,000 iterations, with a meta-learning rate of 0.0005.

Baselines. We compare against different types of deep CNN based detection models, including Faster R-CNN [46], YOLOv2 [44], and SSD [37]. In particular, we evaluate their variants in the small-sample size regime in the following scenarios. (1) Training from scratch: we directly learn a k-shot detector for the 5 novel classes without leveraging the base class examples. (We still pre-train the feature backbone on ImageNet.) (2) Joint learning: we learn a detector for all the 20 classes from large amounts of base training examples and few novel examples. (3) Fine-tuning transfer: after learning the large-sample detector for the 15 base classes, we fine-tune it to be a k-shot detector for the 5 novel classes. The latter two baselines have access to the same amount of training data as ours, but *are not meta-learned*. In addition, we compare with a recent meta-learning based few-shot detection approach that reweights the feature maps from a pre-trained base YOLOv2 detector [29]. For a fair comparison, we also introduce our meta-model for the last layer of the YOLOv2 detection network.

Table 1 summarizes the main results on the 5 novel classes under the 3 different dataset splits. While CNN based detectors have achieved impressive performance with a large amount of annotated examples (for example, the mAP of Faster R-CNN trained on the original full PASCAL VOC is 73.2% [46]), their performance significantly degrades in the few-shot scenario. As expected, directly training the novel detector from scratch using few examples leads to very poor performance (e.g., with the mAP 1%) due to severe over-fitting. Simply joint training with the base classes cannot mitigate this issue, since the detectors are largely dominated by those data-rich base classes. Fine-tuning transfers knowledge from base classes in a more principled manner and leads to improved performance. However, such an improvement is limited, showing the general difficulty of our few-shot detection task.

By contrast, our MetaDet, *regardless of being built upon* Faster R-CNN or YOLO, consistently and significantly outperforms all the baselines across different sample sizes and dataset splits, especially for extremely limited data. This verifies the effectiveness and generality of our meta-learning mechanism. Unlike [29] which is also meta-learned, our approach is able to extract and leverage parameter-level meta-knowledge that is shared *both across multiple few-shot detection tasks and between few-shot and large-sample detectors*, thus outperforming [29] by a large margin. In addition, ‘Faster R-CNN + MetaDet’ outperforms ‘YOLOv2 + MetaDet’, indicating that the meta-

| Method | | Novel Set 1 | | | | | Novel Set 2 | | | | | Novel Set 3 | | | | |
|----------------------|-------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | k=1 | 2 | 3 | 5 | 10 | k=1 | 2 | 3 | 5 | 10 | k=1 | 2 | 3 | 5 | 10 |
| Faster R-CNN [46] w/ | Scratch | 0.0 | 0.2 | 0.5 | 2.0 | 1.7 | 0.0 | 0.0 | 0.7 | 1.3 | 1.7 | 0.0 | 0.9 | 0.5 | 1.8 | 1.5 |
| | Joint | 0.3 | 0.0 | 1.2 | 0.9 | 1.7 | 0.0 | 0.0 | 1.1 | 1.9 | 1.7 | 0.2 | 0.5 | 1.2 | 1.9 | 2.8 |
| | Transfer | 9.1 | 10.9 | 13.7 | 25.0 | 39.5 | 10.9 | 13.2 | 17.6 | 19.5 | 36.5 | 15.0 | 15.1 | 18.3 | 33.1 | 35.9 |
| YOLOv2 [44, 29] w/ | Scratch | 0.0 | 0.3 | 0.9 | 0.5 | 0.9 | 0.0 | 0.0 | 0.1 | 1.0 | 0.6 | 0.0 | 0.0 | 0.6 | 0.6 | 1.2 |
| | Joint | 0.0 | 0.0 | 1.8 | 1.8 | 1.8 | 0.0 | 0.1 | 0.0 | 1.8 | 0.0 | 1.8 | 1.8 | 1.8 | 3.6 | 3.9 |
| | Transfer | 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 |
| SSD [37] w/ | Scratch | 0.0 | 0.0 | 0.3 | 0.9 | 0.9 | 0.0 | 0.0 | 0.6 | 0.9 | 1.0 | 0.0 | 0.3 | 0.6 | 1.0 | 1.1 |
| | Joint | 0.1 | 0.0 | 0.4 | 0.3 | 1.0 | 0.0 | 0.0 | 0.2 | 0.8 | 1.3 | 0.0 | 0.5 | 1.1 | 2.0 | 2.2 |
| | Transfer | 8.2 | 8.9 | 11.3 | 19.7 | 28.8 | 13.2 | 9.0 | 13.2 | 21.3 | 35.5 | 15.6 | 13.4 | 16.2 | 28.6 | 36.2 |
| Meta-Learning | YOLOv2 + FeatReweight [29] | 14.8 | 15.5 | 26.7 | 33.9 | 47.2 | 15.7 | 15.3 | 22.7 | 30.1 | 39.2 | 19.2 | 21.7 | 25.7 | 40.6 | 41.3 |
| | YOLOv2 + MetaDet (Ours) | 17.1 | 19.1 | 28.9 | 35.0 | 48.8 | 18.2 | 20.6 | 25.9 | 30.6 | 41.5 | 20.1 | 22.3 | 27.9 | 41.9 | 42.9 |
| | Faster R-CNN + MetaDet (Ours) | 18.9 | 20.6 | 30.2 | 36.8 | 49.6 | 21.8 | 23.1 | 27.8 | 31.7 | 43.0 | 20.6 | 23.9 | 29.4 | 43.9 | 44.1 |

Table 1: k-shot detection performance (mAP) on the 5 novel classes under 3 different splits of PASCAL VOC. We systemically evaluate the variants of modern detectors in the small-sample size regime. Our MetaDet, *regardless of being built upon* Faster R-CNN or YOLO, consistently outperforms all the baselines. In particular, it is superior to a recent meta-learning based few-shot detection approach [29].

learned two-stage detection network tends to generalize better for few-shot detection, which might benefit from explicit learning of class-agnostic RPN.

Note that due to ImageNet pre-training, we have actually seen global image-level labels of the novel classes. We also run experiments where we remove all the classes from ImageNet that are contained in PASCAL. We use this reduced ImageNet for pre-training and find that the performance of all methods only slightly drops (0.5% mAP drop), but our approach still significantly outperforms the baselines.

Ablation studies. In Tables 2 and 3 we evaluate the contributions of different factors in our approach to the results.

Input to T : few-shot classification vs. few-shot detection. In Table 2, we analyze the impact of the input to our weight prediction meta-model T : only the RoI classification weights w_{cls}^c ('cls'), only the bounding box regression weights w_{loc}^c ('loc'), and the concatenation of both weights ('cls'+ 'loc'). For those category-specific weights which are not used as input to T, we directly learn them on the training data. We can see that all the three types of inputs are superior to the baselines without using T reported in Table 1. This shows that parameter-level meta-learning, in general, enables us to generate the detector parameters for novel classes in a sample-efficient way. More importantly, 'cls'+ 'loc' consistently outperforms its variants, indicating the importance of *addressing both classification and localization for few-shot detection*. And our approach provides such a simple, coherent mechanism, in which we do not need to tackle the two problems using separate techniques.

Meta-strategies: category-specific vs. category-agnostic bounding box regression. Our approach designs different meta-strategies for category-agnostic and category-specific components, and we treat the bounding box regression weights in the detection branch as category-specific. An alternative is to consider them as being category-agnostic, train them on the base classes, and use as initialization for

| Method | Novel Set 1 | | Novel Set 2 | | Novel Set 3 | |
|----------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | k = 3 | 10 | k = 3 | 10 | k = 3 | 10 |
| Meta-model w/ cls | 28.0 | 47.7 | 26.1 | 41.8 | 26.9 | 43.1 |
| Meta-model w/ cls, loc-agn | 28.5 | 48.2 | 26.8 | 42.2 | 27.3 | 43.3 |
| Meta-model w/ loc | 28.8 | 48.4 | 27.2 | 42.6 | 27.9 | 43.9 |
| Meta-model w/ cls+loc | 30.2 | 49.6 | 27.8 | 43.0 | 29.4 | 44.1 |

Table 2: Ablation (mAP) on the meta-strategies regarding category-specific components. (1) Input to meta-model T : 'cls' is RoI classification weights, 'loc' is bounding box regression weights, and 'cls'+ 'loc' is both weights. (2) 'loc-agn': training category-agnostic bounding box regression weights. Our strategy, treating both weights as category-specific and using both as input to T, performs the best.

| Method | Novel Set 1 | | Novel Set 2 | | Novel Set 3 | |
|---------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | k = 3 | 10 | k = 3 | 10 | k = 3 | 10 |
| 1-layer, None | 28.9 | 48.5 | 26.3 | 41.9 | 28.2 | 43.0 |
| 2-layer, ReLU | 29.4 | 48.9 | 26.9 | 42.3 | 28.7 | 43.4 |
| 2-layer, Leaky ReLU | 29.6 | 49.1 | 27.2 | 42.4 | 28.9 | 43.6 |
| Meta-model w/ | | | | | | |
| 3-layer, ReLU | 29.8 | 49.3 | 27.3 | 42.5 | 29.0 | 43.8 |
| 3-layer, Leaky ReLU | 30.2 | 49.6 | 27.8 | 43.0 | 29.4 | 44.1 |
| 4-layer, ReLU | 29.6 | 49.2 | 27.5 | 42.4 | 28.9 | 43.6 |
| 4-layer, Leaky ReLU | 29.5 | 49.0 | 27.1 | 42.2 | 28.7 | 43.5 |

Table 3: Ablation (mAP) on the structure of the weight prediction meta-model T. '3-layer, Leaky ReLU' performs the best, but in general T is robust to specific implementation choices.

fine-tuning on the novel classes. Table 2 shows the performance of dealing with few-shot localization in such a way ('loc-agn'). While outperforming the baselines in Table 1, it is worse than our category-specific counterpart. This indicates that our meta-strategies effectively identify the *intrinsic generality and specialization* within a detection model.

Structure of T. In Table 3, we compare different implementations of our weight prediction meta-model T : as a simple affine transformation, or as a fully-connected network with 2-4 layers. Since Leaky ReLU is used in [70], we evaluate both ReLU and Leaky ReLU as activation function in the hidden layers. This study shows that a 3-layer network with Leaky ReLU achieves the best mAP. We use

this design of T in all subsequent experiments.

Qualitative visualizations. Figure 7 shows examples of 10-shot detection test results.

4.1.2 Evaluation on MS-COCO

Dataset and task. We now evaluate on a more challenging MS-COCO dataset [36]. This dataset involves 80k training, 40k validation, and 20k test images over 80 object categories. Consistent with [29], we use 5k images from the validation set for evaluation, and the remaining trainval images for training. We choose the 20 categories that also present in PASCAL VOC as novel classes and the remaining 60 categories as base classes. The k-shot detection task is constructed in a similar way as before, and k is set to be 10 and 30. Our detector is Faster R-CNN trained from scratch. In this setting, we might have seen novel classes as background without any labels. Following the standard evaluation metric on COCO [46, 37], we report the mAP averaged over different IoU thresholds from 0.5 to 0.95.

Comparison with state-of-the-art. Table 4 summarizes the results. Similar to Section 4.1.1, we compare against the top performing baselines of fine-tuning transfer with Faster R-CNN [46] and YOLOv2 [44]. We also compare with the meta-learning approach [29]. Again, our MetaDet *consistently and significantly* outperforms all the baselines for different number of shots k. In addition, comparing with the PASCAL VOC results in Table 1, the detection performance on the challenging COCO benchmark drops, showing the difficulty of few-shot detection in realistic scenarios.

4.2. Cross-Domain Few-Shot Detection

Thus far, we have experimented with a widely-used setting in which we learn from base classes to detect novel classes in the same dataset. In a more practical scenario, we need to evaluate the cross-domain generalization ability [27, 7]. Hence, we use a source dataset as base and another target dataset as novel with two disjoint sets of classes. Such cross-domain scenarios allow us to understand the effects of domain shifts to few-shot detection approaches.

COCO PASCAL. We use the 60 categories on COCO as base classes as in Section 4.1.2, and use all the 20 categories on PASCAL as novel classes. We focus on 10-shot detection of the 20 novel classes [29]. Figure 3 shows that our MetaDet achieves the best performance. Moreover, comparing with the PASCAL within-domain results in Section 4.1.1, we notice that the performance on the PASCAL novel classes becomes worse in the cross-domain setting, despite that we have more data from base classes. This indicates that, to fully address few-shot detection in practice, effectively overcoming domain shift issues is critical, which is an interesting direction for further investigation.

COCO ImageNet. We now evaluate on the larger-

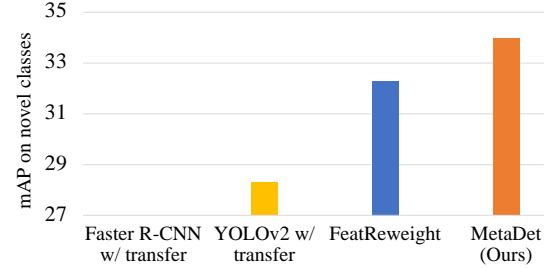


Figure 3: 10-shot cross-domain detection performance (mAP) on the 20 novel classes under COCO PASCAL. Our MetaDet significantly outperforms all the baselines, including the meta-learning based approach (FeatReweight) [29].

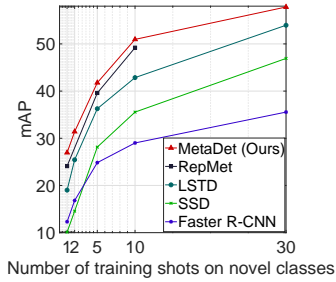


Figure 4: k-shot cross-domain detection performance on the 50 novel classes under COCO ImageNet. Our MetaDet significantly outperforms existing transfer (LSTD) [6] or metric (RepMet) learning [54] based few-shot detection approaches.

scale, few-shot detection benchmark in [6]. The base classes are the entire 80 categories on COCO, and the novel classes are 50 non-overlapping categories selected on ImageNet2015 [47]. k is set to be 1, 5, and 10. The test set consists of 100 images per novel class on ImageNet2015. Following [54], we train our detector on COCO from scratch.

Comparison with state-of-the-art. In addition to Faster R-CNN [46] and SSD [37], we compare against two recent, state-of-the-art few-shot detection approaches on this benchmark. (1) LSTD [6]: a regularized fine-tuning method that transfers a pre-trained detector to few-shot target tasks. (2) RepMet [54]: a distance metric learning method that models category multi-modal distribution.

Figure 4 shows that our MetaDet *consistently* outperforms these approaches by large margins. In particular, our MetaDet, built upon Faster R-CNN with VGG16, outperforms RepMet which uses a more powerful detector architecture (*i.e.*, the FPN backbone [35] in its deformable convolutions variant [10]). This demonstrates that leveraging parameter-level meta-knowledge helps address the domain shift issues. Moreover, RepMet simply introduces a few-shot classifier into a detector. Our superior performance again shows the importance of jointly tackling few-shot classification and localization for few-shot detection.

4.3. Long-Tail Detection

To further show the generality of our approach in real-world scenarios, we consider using it to detect objects from

| | | Avg. Precision, IoU | | | Avg. Precision, Area | | | Avg. Recall, #Dets | | | Avg. Recall, Area | | |
|--------|-------------------------------|---------------------|-------------|------------|----------------------|------------|-------------|--------------------|-------------|-------------|-------------------|-------------|-------------|
| Method | | 0.5:0.95 | 0.5 | 0.75 | S | M | L | 1 | 10 | 100 | S | M | L |
| k=10 | Faster R-CNN [46] w/ transfer | 3.3 | 7.8 | 1.9 | 0.8 | 2.2 | 6.7 | 8.1 | 10.8 | 10.9 | 1.3 | 5.2 | 21.7 |
| | YOLOv2 [44, 29] w/ transfer | 3.1 | 7.9 | 1.7 | 0.7 | 2.0 | 6.3 | 7.8 | 10.5 | 10.5 | 1.1 | 5.5 | 20 |
| | FeatReweight [29] | 5.6 | 12.3 | 4.6 | 0.9 | 3.5 | 10.5 | 10.1 | 14.3 | 14.4 | 1.5 | 8.4 | 28.2 |
| | MedaDet (Ours) | 7.1 | 14.6 | 6.1 | 1.0 | 4.1 | 12.2 | 11.9 | 15.1 | 15.5 | 1.7 | 9.7 | 30.1 |
| k=30 | Faster R-CNN [46] w/ transfer | 7.8 | 17.8 | 6.0 | 0.3 | 4.1 | 13.9 | 12.2 | 15.9 | 15.7 | 1.1 | 8.0 | 30.9 |
| | YOLOv2 [44, 29] w/ transfer | 7.7 | 16.7 | 6.4 | 0.4 | 3.3 | 14.4 | 11.7 | 15.3 | 15.3 | 1.0 | 7.7 | 29.2 |
| | FeatReweight [29] | 9.1 | 19.0 | 7.6 | 0.8 | 4.9 | 16.8 | 13.2 | 17.7 | 17.8 | 1.5 | 10.4 | 33.5 |
| | MetaDet (Ours) | 11.3 | 21.7 | 8.1 | 1.1 | 6.2 | 17.3 | 14.5 | 18.9 | 19.2 | 1.8 | 11.1 | 34.4 |

Table 4: k-shot detection performance on the 20 novel classes of COCO, evaluated under COCO’s metric. Our MetaDet significantly outperforms all the baselines. In particular, it is superior to a recent meta-learning based approach [29].

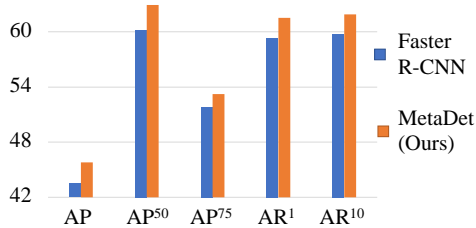


Figure 5: Long-tail detection performance, Average Precision (AP) and Average Recall (AR), on *all classes* of iNaturalist. AP, AP⁵⁰, and AP⁷⁵ denote AP@[IoU=.50:.05:.95], AP@[IoU=.50], and AP@[IoU=.75], respectively; AR¹ and AR¹⁰ denote AR given 1 detection and 10 detections per image, respectively. Our MetaDet shows its generality in realistic long-tail detection.

long-tailed, imbalanced datasets, in which a few dominant (‘head’) classes claim most of the examples, while most of the other (‘tail’) classes are represented by relatively few examples [74, 71, 9]. We treat the data-rich head categories as base classes and the data-poor tail categories as novel classes. Our approach can be extended by learning the meta-model on the head and then transferring it to the tail. Note that different from the previous experiments, here we report the detection performance *over all classes*.

Evaluation on iNaturalist. We evaluate on iNaturalist [65], a fine-grained, long-tailed species dataset. Its detection benchmark contains 2,854 classes. We use the detection metric in [65], and evaluate on the validation images that contain a single instance as in [65]. We use a fixed head and tail split, selected by cross-validation, and compare with Faster R-CNN [46, 65]. We use pre-training on ImageNet, which might see certain coarse concepts like bird. However, we have never before seen the fine-grained bird species. Figure 5 summarizes the performance averaged over all classes and Figure 6 details the per-class performance. The performance improvement of our approach mainly comes from that on tail classes, showing that we learn accurate few-shot models. Meanwhile, our approach effectively deals with the significantly imbalanced distribution, thus remaining accurate on the head (base) classes.

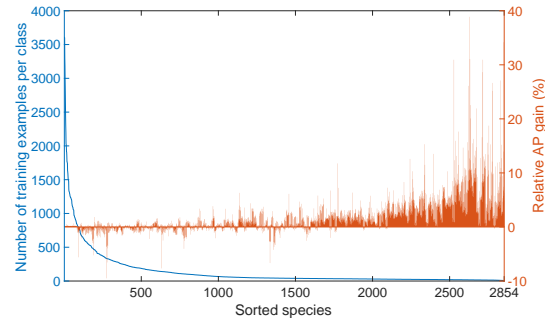


Figure 6: Per-class performance of our MetaDet on iNaturalist. Y-axis (left) and the blue curve: long-tail distribution. Y-axis (right) and the orange curve: AP improvement of our MetaDet relative to the Faster R-CNN baseline. Our MetaDet significantly improves detection in the tail, in particular by large margins for tail classes with extremely limited data, while remaining accurate in the head.

Figure 7: Examples of 10-shot detection results of our MetaDet on novel classes of PASCAL VOC.

5. Conclusion

In this work, we have presented an approach to few-shot detection for novel classes that simultaneously tackles few-shot classification and localization in a unified, coherent way via meta-learning. We propose specialized meta-strategies to disentangle the learning of category-agnostic and category-specific components in a CNN based detection model. Our approach achieves state-of-the-art detection performance in a variety of realistic scenarios, including within-domain, cross-domain, and long-tailed settings, and under different notions of novel classes.

Acknowledgments: We thank Liangyan Gui for valuable and insightful discussions. This work was supported in part by ONR MURI N000141612007, U.S. Army Research Laboratory (ARL) under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016, and NSF Grant 1618903. We also thank NVIDIA for donating GPUs, Facebook Research and Academic Relations program, and AWS Cloud Credits for Research program.

References

- [1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *NeurIPS*, 2016. **2**
- [2] Ankan Bansal, Karan Sikka, Gaurav Sharma, Rama Chellappa, and Ajay Divakaran. Zero-shot object detection. In *ECCV*, 2018. **3**
- [3] Abhijit Bendale and Terrance Boulton. Towards open world recognition. In *CVPR*, 2015. **1**
- [4] Luca Bertinetto, João F. Henriques, Jack Valmadre, Philip Torr, and Andrea Vedaldi. Learning feed-forward one-shot learners. In *NeurIPS*, 2016. **2**
- [5] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. **3**
- [6] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. LSTD: A low-shot transfer detector for object detection. In *AAAI*, 2018. **1, 2, 3, 7**
- [7] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. **7**
- [8] Zitian Chen, Yanwei Fu, Yu-Xiong Wang, Lin Ma, Wei Liu, and Martial Hebert. Image deformation meta-networks for one-shot learning. In *CVPR*, 2019. **2**
- [9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. **8**
- [10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, 2017. **2, 7**
- [11] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *CVPR*, 2017. **3**
- [12] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. *IEEE TPAMI*, 41(7):1641–1654, 2019. **1, 3**
- [13] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. **2, 4, 5**
- [14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. **2, 4, 5**
- [15] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE TPAMI*, 28(4):594–611, 2006. **2**
- [16] Michael Fink. *Acquiring a new class from a few examples: Learning recurrent domain structures in humans and machines*. PhD thesis, The Hebrew University of Jerusalem, 2011. **1**
- [17] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017. **2, 3**
- [18] Ross Girshick. Fast R-CNN. In *ICCV*, 2015. **2, 3, 5**
- [19] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. **1, 2**
- [20] Liang-Yan Gui, Yu-Xiong Wang, Deva Ramanan, and José M. F. Moura. Few-shot human motion prediction via meta-learning. In *ECCV*, 2018. **2**
- [21] Agrim Gupta, Piotr Dollár, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. **1**
- [22] David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks. In *ICLR*, 2017. **2**
- [23] Bharath Hariharan and Ross Girshick. Low-shot visual recognition by shrinking and hallucinating features. In *ICCV*, 2017. **2**
- [24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017. **2**
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE TPAMI*, 37(9):1904–1916, 2015. **2**
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. **1**
- [27] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. In *ICLR*, 2018. **7**
- [28] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross Girshick. Learning to segment every thing. In *CVPR*, 2018. **2**
- [29] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019. **1, 2, 3, 5, 6, 7, 8**
- [30] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Workshops*, 2015. **2**
- [31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NeurIPS*, 2012. **1**
- [32] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015. **2**
- [33] Kwonjoon Lee, Subhansu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019. **2**
- [34] Ke Li and Jitendra Malik. Learning to optimize. In *ICLR*, 2017. **2**
- [35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. **2, 7**
- [36] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. **2, 4, 5, 7**
- [37] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016. **2, 5, 6, 7**
- [38] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X. Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. **1**

- [39] Ishan Misra, Abhinav Shrivastava, and Martial Hebert. Watch and learn: Semi-supervised learning for object detectors from video. In *CVPR*, 2015. 3
- [40] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *ICML*, 2017. 2
- [41] Shafin Rahman, Salman Khan, and Fatih Porikli. Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts. In *ACCV*, 2018. 3
- [42] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. In *ICLR*, 2017. 2
- [43] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2
- [44] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *CVPR*, 2017. 2, 3, 5, 6, 7, 8
- [45] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 2
- [46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 2, 3, 4, 5, 6, 7, 8
- [47] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015. 2, 4, 5, 7
- [48] Andrei A. Rusu, Dushyant Rao, Jakub Sygnowski, Oriol Vinyals, Razvan Pascanu, Simon Osindero, and Raia Hadsell. Meta-learning with latent embedding optimization. In *ICLR*, 2019. 2
- [49] Ruslan Salakhutdinov, Josh Tenenbaum, and Antonio Torralba. One-shot learning with a hierarchical nonparametric Bayesian model. In *ICML Workshops*, 2012. 1
- [50] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *ICML*, 2016. 2
- [51] Jürgen Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1987. 2
- [52] Jürgen Schmidhuber, Jieyu Zhao, and Marco Wiering. Shifting inductive bias with success-story algorithm, adaptive levin search, and incremental self-improvement. *Machine Learning*, 28(1):105–130, 1997. 2
- [53] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Abhishek Kumar, Rogerio Feris, Raja Giryes, and Alex Bronstein. Delta-encoder: An effective sample synthesis method for few-shot object recognition. In *NeurIPS*, 2018. 2
- [54] Eli Schwartz, Leonid Karlinsky, Joseph Shtok, Sivan Harary, Mattias Marder, Sharathchandra Pankanti, Rogerio Feris, Abhishek Kumar, Raja Giryes, and Alex M. Bronstein. Rep-Met: Representative-based metric learning for classification and one-shot object detection. In *CVPR*, 2019. 1, 2, 3, 7
- [55] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 2
- [56] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5
- [57] Abhishek Sinha, Mausoom Sarkar, Aahitagni Mukherjee, and Balaji Krishnamurthy. Introspection: Accelerating neural network training by learning weight evolution. In *ICLR*, 2017. 2
- [58] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017. 1, 2, 5
- [59] Hyun Oh Song, Yong Jae Lee, Stefanie Jegelka, and Trevor Darrell. Weakly-supervised discovery of visual pattern configurations. In *NeurIPS*, 2014. 3
- [60] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 2
- [61] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? *NeurIPS*, 1996. 2
- [62] Sebastian Thrun. Lifelong learning algorithms. *Learning to learn*, 8:181–209, 1998. 1, 2
- [63] Sebastian Thrun and Lorian Pratt. *Learning to learn*. Springer Science & Business Media, 2012. 2
- [64] Pavel Tokmakov, Yu-Xiong Wang, and Martial Hebert. Learning compositional representations for few-shot recognition. In *ICCV*, 2019. 2
- [65] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *CVPR*, 2018. 2, 4, 8
- [66] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017. 1
- [67] Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016. 1, 2, 3, 5
- [68] Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In *CVPR*, 2018. 2
- [69] Yu-Xiong Wang and Martial Hebert. Model recommendation: Generating object detectors from few samples. In *CVPR*, 2015. 1, 3
- [70] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 2, 4, 6
- [71] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, 2017. 1, 2, 8
- [72] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NeurIPS*, 2014. 2, 3
- [73] Pengkai Zhu, Hanxiao Wang, and Venkatesh Saligrama. Zero shot detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019. 3
- [74] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *CVPR*, 2014. 1, 8