StarNet: towards Weakly Supervised Few-Shot Object Detection

Leonid Karlinsky*[1], Joseph Shtok*[1], Amit Alfassy*[1,3], Moshe Lichtenstein*[1], Sivan Harary[1], Eli Schwartz[1,2], Sivan Doveh[1], Prasanna Sattigeri[1], Rogerio Feris[1], Alexander Bronstein[3], Raja Giryes[2]

[1] IBM Research AI
[2] Tel-Aviv University
[3] Technion
leonidka@il.ibm.com

Abstract

Few-shot detection and classification have advanced significantly in recent years. Yet, detection approaches require strong annotation (bounding boxes) both for pre-training and for adaptation to novel classes, and classification approaches rarely provide localization of objects in the scene. In this paper, we introduce StarNet - a few-shot model featuring an end-to-end differentiable non-parametric star-model detection and classification head. Through this head, the backbone is meta-trained using only image-level labels to produce good features for jointly localizing and classifying previously unseen categories of few-shot test tasks using a star-model that geometrically matches between the query and support images (to find corresponding object instances). Being a few-shot detector, StarNet does not require any bounding box annotations, neither during pre-training, nor for novel classes adaptation. It can thus be applied to the previously unexplored and challenging task of Weakly Supervised Few-Shot Object Detection (WS-FSOD), where it attains significant improvements over the baselines. In addition, StarNet shows significant gains on few-shot classification benchmarks that are less cropped around the objects (where object localization is key).

Introduction

Recently, great advances have been made in the field of few-shot learning using deep convolutional neural networks (CNNs). This learning regime targets situations where only a handful of examples for the target classes (typically 1 or 5) are available at test time, while the target classes themselves are novel and unseen during pre-training. Commonly, models are pre-trained on a large labeled dataset of 'base' classes, e.g. (Lee et al. 2019; Snell, Swersky, and Zemel 2017; Li et al. 2017). There, depending on the application, label complexity varies from image-level class labels (classification), to labeled boxes (detection), to labeled pixel-masks (segmentation). As shown in (Chen et al. 2019), few-shot methods are highly sensitive to 'domain shift'. For these methods to be effective, the base classes used for pre-training need to be in the same 'visual domain' as the target (test) classes. That said, for applications which require richer annotation, such as detection, entering new visual domains is still prohibitively expensive due to thousands of base classes images that need to be annotated in order to *pre-train* a Few-

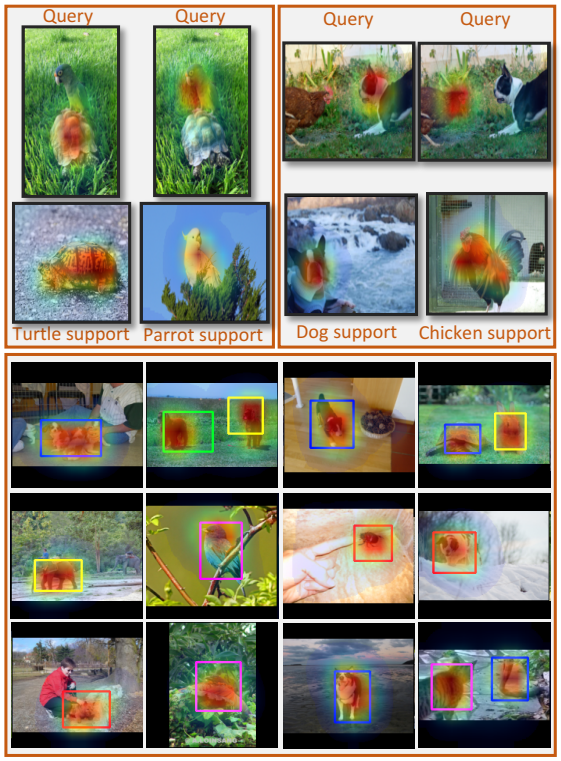


Figure 1: StarNet provides evidence for its predictions by finding (semantically) matching regions between query and support images of a few-shot task, thus effectively detecting object instances. Top: Matching regions are drawn as heatmaps for each query and support pair. Clearly, in this situation there is no single correct class label for these queries. Yet, StarNet successfully highlights the matched objects on both the query and the support images, thus effectively explaining the different possible labels. Bottom: StarNet paves the way towards previously unexplored Weakly-Supervised Few-Shot Object Detection (WS-FSOD) task.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

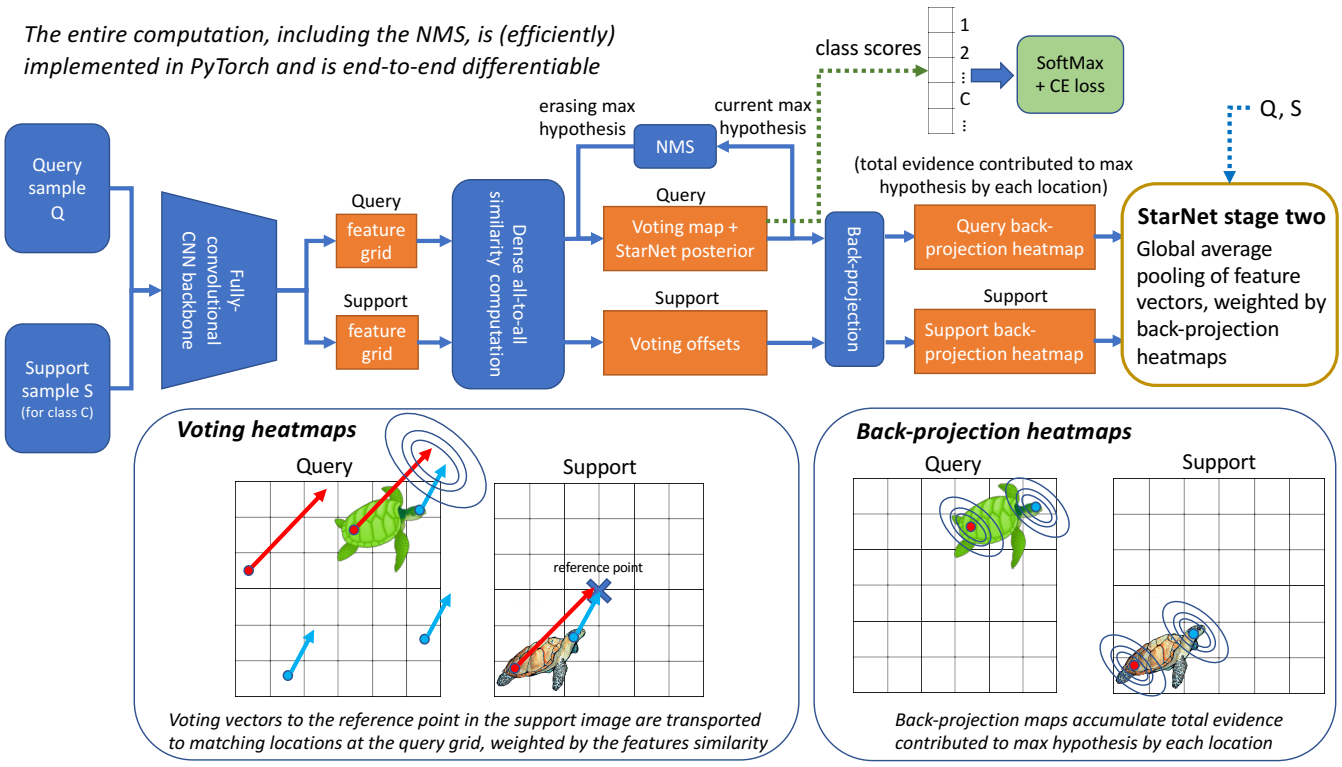


Figure 2: StarNet overview. Query image Q is matched to a candidate support image S jointly localizing instances of a shared category (if exist). NMS iteratively suppresses the max hypothesis allowing matching non-rigid object parts or multiple objects. Back-projection generates decision evidence heatmaps for additional refinement stage. StarNet is end-to-end differentiable.

Shot Object Detector (FSOD), e.g. (Chen et al. 2018; Karlinsky et al. 2019; Kang et al. 2019; Wang, Ramanan, and Hebert 2019; Liu et al. 2019), for the new domain.

Few-shot classifiers require much less annotation efforts for pre-training, but can only produce image-level class predictions. Of course, general purpose methods such as the popular GradCAM (Selvaraju et al. 2017), are able (to some extent) to highlight the pixels supporting the prediction of any classifier. But, as illustrated in Figure 3, and evaluated in Table 1, these are less effective for few-shot classifiers that need to predict novel classes based on only a few labeled support examples available for a few-shot task.

In this paper, we introduce a new few-shot learning task: *Weakly-Supervised* Few-Shot Object Detection (WS-FSOD) - pre-training a few-shot detector and adapting it (with few examples) to novel classes *without bounding boxes* and using only image level class label annotations. We also introduce StarNet - a first weakly-supervised few-shot detector that geometrically matches query and support images, classifying queries by localizing objects contained within (Fig. 1 bottom). StarNet features an end-to-end differentiable head performing non-parametric star-model matching. During training, gradients flowing through the StarNet head teach its underlying CNN backbone to produce features best supporting correct geometric matching. StarNet handles multiple matching hypotheses (e.g. corresponding to multiple objects or object parts), each analyzed by a differentiable back-projection module producing heatmaps of the discovered matching regions (on both query and support images). After training, these heatmaps usually highlight object instances, thus detecting the objects and providing explanations for the model's predictions (Fig. 1 top).

To summarize, our contributions are as follows: (1) we propose WS-FSOD - a new challenging few-shot learning task of pre-training a few-shot detector and adapting it to novel classes without bounding boxes and using only image class labels; (2) as a solution to WS-FSOD, we propose StarNet - a first end-to-end differentiable non-parametric star-model posed as a neural network, demonstrating promising results for WS-FSOD by significantly outperforming a diverse set of baselines for this new task; (3) as a bonus, not requiring bounding boxes allows StarNet to be directly applied to few-shot classification, where we demonstrate it to be especially useful on benchmarks in which images are less cropped around the objects (e.g. CUB and ImageNetLOC-FS), and for which object localization is key.

Related Work

In this section we briefly review the modern few-shot learning focusing on meta-learning, discuss weakly-supervised detection, cover star-model related methods, and review methods for few-shot localization and detection.

Meta-learning methods, e.g. (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Li et al. 2019a, 2017; Zhou, Wu, and Li 2018; Ravi and Larochelle 2017; Rusu et al. 2018; Oreshkin, Rodriguez, and Lacoste 2018; Zhang et al. 2019; Zhang, Zhang, and Koniusz 2019; Gidaris and Komodakis 2019; Alfassy et al. 2019) learn from few-shot tasks (or episodes) rather then from individual labeled samples. Such tasks are small datasets, with few labeled training (support) examples, and a few test (query) examples. The goal is to learn a model that can adapt to new tasks with novel categories, unseen during training. In (Dvornik, Schmid, and Mairal 2019) ensemble methods for few-shot learning are evaluated. MetaOptNet (Lee et al. 2019) utilizes an end-to-end differentiable SVM solver on top of a CNN backbone. (Gidaris et al. 2019) combines few-shot supervision with self-supervision, in order to boost the few-shot performance. In (Qiao et al. 2019; Li et al. 2019b; Kim et al. 2019; Gidaris et al. 2019) additional unlabeled data is used, while (Xing et al. 2019) leverages additional semantic information available for the classes.

Star Models (SM) and **Generalized Hough Transform (GHT)** techniques were popular classification and detection methods before the advent of CNNs. In these techniques, objects were modeled as a collection of parts, independently linked to the object variables via Gaussian priors to allow local deformations. Classically, parts were represented using patch descriptors (Sali and Ullman 1999; Leibe, Leonardis, and Schiele 2006; Maji and Malik 2009; Karlinsky et al. 2017), or SVM part detectors in DPM (Felzenszwalb et al. 2010). DPM was later extended to CNN based DPM in (Girshick et al. 2015). Recently, in (Qi et al. 2019) GHT was used to detect objects in 3D point clouds, in the fully supervised and non-few-shot setting. Unlike DPM (Felzenszwalb et al. 2010; Girshick et al. 2015), StarNet is non-parametric, in a sense that parts are not explicitly learned and are not fixed during inference, and unlike all of the aforementioned methods (Sali and Ullman 1999; Leibe, Leonardis, and Schiele 2006; Maji and Malik 2009; Felzenszwalb et al. 2010; Girshick et al. 2015; Karlinsky et al. 2017; Qi et al. 2019), it is trained using only class labels (no bounding boxes) and targets the few-shot setting. In (Lin, Roychowdhury, and Maji 2017) a *non* few-shot classification network is trained through pairwise local feature matching, but unlike in StarNet, no geometrical constraints on the matches are used. Finally, unlike the classical approaches (Sali and Ullman 1999; Leibe, Leonardis, and Schiele 2006; Maji and Malik 2009; Felzenszwalb et al. 2010), StarNet features (used for local matching) are not handcrafted, but are rather end-to-end optimized by propagating gradients through StarNet head to a CNN backbone.

Weakly-supervised object detection refers to techniques that learn to detect objects despite being trained with only image-level class labels. In a number of works, an external region proposal mechanism (e.g., Selective Search (Uijlings et al. 2013)) is employed to endow a pre-trained CNN classifier with a detection head (Bilen and Vedaldi 2016), or to provide initial proposals for RPN (Zeng et al. 2019). In (Tang et al. 2018a), the proposals are clustered into groups to facilitate iterative training of instance classifiers. More recently, in (Tang et al. 2018b), a region proposal sub-network is trained jointly with the backbone, by refining initial (sliding window) proposals. To the best of our knowledge, no prior works have considered the weakly-supervised detection in the *few-shot* setting.

Few-shot with localization and attention is a relatively recent research direction. Unlike StarNet, most of these methods rely on bounding box supervision during pre-training. Using bounding boxes, several works (Chen et al. 2018; Karlinsky et al. 2019; Kang et al. 2019; Wang, Ramanan, and Hebert 2019; Liu et al. 2019; Wang et al. 2020) have extended object detection techniques (Ren et al. 2015; Liu et al. 2016) to few-shot setting. (Wertheimer and Hariharan 2019) uses an attention module trained using bounding boxes. SILCO (Hu et al. 2019) trains using bounding boxes to localizes objects in 1-way / 5-shot mode only. (Shaban et al. 2019) uses Multiple Instance Learning and an RPN pre-trained using bounding boxes on MS-COCO (Lin et al. 2014). SAML (Hao et al. 2019) and DeepEMD (Zhang et al. 2020) compute a dense feature matching applying MLP or EMD metric as a classifier, but unlike StarNet geometric matching is not employed. In CAN (Hou et al. 2019) attention maps for query and support images are generated by 1×1 convolution applied to a pairwise local feature comparison map. These attention maps are not intended for object localization, so unlike StarNet, geometry of the matches in (Hou et al. 2019) is not modeled. In DC (Lifchitz et al. 2019) a classifier is applied densely on each of the local features in the feature map, their decisions are globally averaged, unlike StarNet, without employing geometry.

Recently, (Choe et al. 2020) proposed a few-shot protocol for Weakly Supervised Object *Localization* (WSOL) - given (a single) true class label of the test image, localizing an object of that class in it. In their protocol *ImageNet pre-trained* models are fine-tuned using 5-shots with *bounding boxes* supervision. In contrast, in this paper we propose Weakly Supervised Few-Shot Object *Detection* (WS-FSOD) protocol, where: test images (potentially multiple) class labels are not given; models are pre-trained from scratch on the train portions of the benchmarks and adapted to novel classes using 1 or 5 shots; and no bounding boxes are used for training. We believe our WS-FSOD protocol to be more fitting situations of entering a new visual domain, where ImageNet-scale pre-training and box annotations are not available.

Method

Here we provide the details of the StarNet method. First we describe the approach for calculating the StarNet posterior for each query-support pair and using it to predict the class scores for every query image in a single-stage StarNet. Next we explain how to revert StarNet posterior computation using back-projection, obtaining evidence maps (on both query and support) for any hypothesis. Then we show how to enhance StarNet performance by adding a second-stage hypothesis classifier utilizing the evidence maps to pool features from the (query and support) matched regions, effectively suppressing background clutter. Finally, we provide implementation details and running times. Figure 2 and

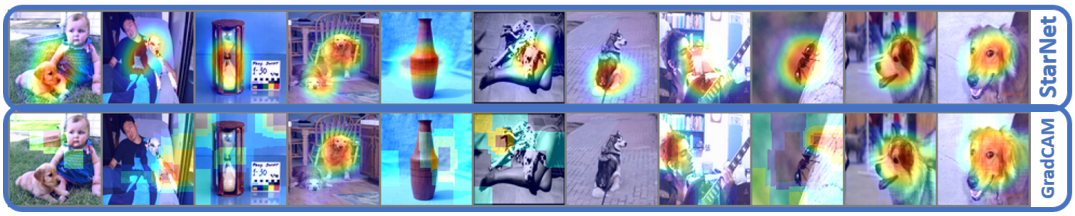


Figure 3: Comparison with GradCAM: StarNet back-projection maps (top row) and GradCAM (Selvaraju et al. 2017) attention maps (bottom row) computed for MetaOptNet+SVM (Lee et al. 2019) on *mini*ImageNet test images. GradCAM failures are likely due to the few-shot setting, or presence of multiple objects.

Algorithm 1 provide an overview of our approach [1].

Single-Stage StarNet

StarNet is trained in a meta-learning fashion, where k-shot, n-way training episodes are randomly sampled from the train data. Each episode (a.k.a few-shot task) E consists of k random support samples (k-shot) and q random query samples for each of n random classes (n-way). Denote by Q and S a pair of query and support images belonging to E. Let ϕ be a fully convolutional CNN feature extractor, taking a square RGB image input and producing a feature grid tensor of dimensions $r \times r \times f$ (here r is the spatial dimension, and f is the number of channels). Applying ϕ on Q and S computes the query and support grids of feature vectors:

$$\begin{aligned} &\{\phi(Q)_{i,j} \in \mathcal{R}^f | \ 1 \leq i,j \leq r\} \\ &\{\phi(S)_{l,m} \in \mathcal{R}^f | \ 1 \leq l,m \leq r\} \end{aligned} \tag{1}$$

For brevity we will drop ϕ in further notation and write $Q_{i,j}$ and $S_{l,m}$ instead of $\phi(Q)_{i,j}$ and $\phi(S)_{l,m}$. We first L_2 normalize $Q_{i,j}$ and $S_{l,m}$ for all grid cells, and then compute a tensor D of size $r \times r \times r \times r$ of all pairwise distances between Q and S feature grids cells:

$$D_{i,j,l,m} = ||Q_{i,j} - S_{l,m}||^2 \tag{2}$$

D is efficiently computed for all support-query pairs simultaneously via matrix multiplication with broadcasting. We then convert D into a (same size) tensor of unnormalized probabilities P, where:

$$P_{i,j,l,m} = e^{-0.5 \cdot D_{i,j,l,m}/\sigma_f^2} \tag{3}$$

is the probability that $Q_{i,j}$ matches $S_{l,m}$ in a sense of representing the same part of the same category. Some object part appearances are more rare than others; to accommodate for that, P is normalized to obtain the tensor R of the same size, where $R_{i,j,l,m} = P_{i,j,l,m}/N_{i,j}$ is the likelihood ratio between 'foreground' match probability $P_{i,j,l,m}$, and the 'background' probability $N_{i,j}$ of 'observing' $Q_{i,j}$ in a random image, approximated as:

$$N_{i,j} = \sum_S \sum_{l,m} P_{i,j,l,m} \tag{4}$$

where $\sum_S$ is computed by matching the same query Q to all of the supports in the episode. Note that in $R_{i,j,l,m}$, the normalization factor of unnormalized probabilities P cancels out. Let $w = (r/2, r/2)$ be a reference point in the center of S feature grid. We compute voting offsets as $o_{l,m} = w - (l,m)$ and the voting target as $t_{i,j,l,m} = (i,j) + o_{l,m}$ being the corresponding location to the reference point w on the query image Q assuming that indeed $Q_{i,j}$ matches $S_{l,m}$. By construction, $t_{i,j,l,m}$ can be negative, with values ranging between $(-r/2, -r/2)$ and $(3r/2, 3r/2)$, thus forming a $2r \times 2r$ hypothesis grid of points in coordinates of Q potentially corresponding to point w on S.

Next, for every point (x,y) on the hypothesis grid of Q, StarNet accumulates the overall belief $A(x,y)$ that (x,y) corresponds to w (on S) considering independently the evidence $R_{i,j,l,m}$ from all potential matches between support and query features. In probabilistic sense, $A(x,y)$ relates to Naive-Bayes (Bishop 2006), and hence should accumulate *log*-likelihood ratios $log(R_{i,j,l,m})$. However, as in (Karlinsky et al. 2017), to be more robust to background clutter, in StarNet, likelihood ratios are directly accumulated:

$$A(x,y) = \sum_{\substack{\{i,j,l,m\} \ s.t. \\ t_{i,j,l,m}=(x,y)}} R_{i,j,l,m} \tag{5}$$

For each hypothesis (x,y), the final StarNet posterior $V_{Q,S}(x,y)$ is computed by convolving A with $G(\sigma_g)$ - a symmetric Gaussian kernel: $V_{Q,S} = G(\sigma_g) \circledast A$. This efficiently accounts for any random relative location shift (local object part deformation) allowed to occur with the $G(\sigma_g)$ Gaussian prior for any matched pair of $Q_{i,j}$ and $S_{l,m}$.

We compute the score (logit) of predicting the category label c for Q as:

$$SC_1(c;Q) = \frac{1}{k} \cdot \sum_{\substack{S \in E \ s.t. \\ C(S)=c}} \max_{x,y} V_{Q,S}(x,y) \tag{6}$$

where $C(S)$ is the class label of S, and k is the number of shots (support samples per class) in the episode E. During meta-training the CNN backbone ϕ is end-to-end trained using Cross Entropy (CE) loss between $SC_1(c;Q)$ (after softmax) and the known category label of Q in the training episode. The need to only match images with the same class label, drives the optimization to maximally match the regions that correspond to the only thing that is in fact shared between such images - the instances of the shared category (please see Appendix for examples and video illustrations).

[1] Our code is avaialble at: https://github.com/leokarlin/StarNet

Back-Projection Maps

For any pair of query Q and support S, and any hypothesis location $(\hat{x}, \hat{y})$ on the $2r \times 2r$ grid, and in particular one with the maximal StarNet posterior value $(\hat{x}, \hat{y}) = \arg\max_{x,y} V_{Q,S}(x, y)$, we can compute two back-projection heatmaps (one for Q and one for S). These are $r \times r$ matrices in the feature grid coordinates of Q and S respectively, whose entries contain the amount of contribution that the corresponding feature grid cell on Q or S gave to the posterior probability $V_{Q,S}(\hat{x}, \hat{y})$:

$$BP_{Q|S}(i,j) = \sum_{l,m} R_{i,j,l,m} \cdot e^{-0.5 \cdot ||t_{i,j,l,m} - (\hat{x},\hat{y})||^2 / \sigma_g^2} \tag{7}$$

the $BP_{S|Q}(l, m)$ is computed in completely symmetrical fashion by replacing summation by l, m with summation by i, j. After training, the back-projection heatmaps are highlighting the matching regions on Q and S that correspond to the hypothesis $(\hat{x}, \hat{y})$, which for query-support pairs sharing the same category label are in most cases the instances of that category (examples provided in Appendix).

The back-projection is iteratively repeated by suppressing $(\hat{x}, \hat{y})$ (and its 3×3 neighborhood) in $V_{Q,S}(x, y)$ as part of the Non-Maximal Suppression (NMS) process implemented as part of the neural network. NMS allows for better coverage of non-rigid objects detected as sum of parts and for discovering additional objects of the same category. Please see Fig. 1, Fig. 3 (image 4, top row), and the Appendix, for examples of images with multiple objects detected by StarNet. In our implementation, NMS repeats until the next maximal point is less then an $\eta = 0.5$ from the global maximum.

Two-Stage StarNet

Having computed the $BP_{Q|S}$ and $BP_{S|Q}$ back-projection heatmaps, we take inspiration from the 2-stage CNN detectors (e.g. FasterRCNN (Ren et al. 2015)) to enhance the StarNet performance with a second stage classifier benefiting from category instances localization produced by StarNet (in $BP_{Q|S}$ and $BP_{S|Q}$). We first normalize each of the $BP_{Q|S}$ and $BP_{S|Q}$ to sum to 1, and then generate the following pooled feature vectors by weighted global average pooling with $BP_{Q|S}$ and $BP_{S|Q}$ weights:

$$\begin{aligned} F_{Q|S} &= \sum_{i,j} BP_{Q|S}(i,j) \cdot Q_{i,j} \\ F_{S|Q} &= \sum_{l,m} BP_{S|Q}(l,m) \cdot S_{l,m} \end{aligned} \tag{8}$$

here the feature grids $Q_{i,j}$ and $S_{l,m}$ can be computed using ϕ or using a separate CNN backbone trained jointly with the first stage network (we evaluate both in experiments section). Our second stage is a variant of the Prototypical Network (PN) classifier (Snell, Swersky, and Zemel 2017). We compute the prototypes for class c and the query Q as:

$$\begin{aligned} F^P_{c|Q} &= \frac{1}{k} \cdot \sum_{S \in E \ s.t. \ C(S)=c} F_{S|Q} \\ F^P_{Q|c} &= \frac{1}{k} \cdot \sum_{S \in E \ s.t. \ C(S)=c} F_{Q|S} \end{aligned} \tag{9}$$

Algorithm 1: StarNet training

I_Q = query, I_S = support, L_S, L_Q = 1-hot, ϕ = backbone;
$S, Q = \phi(I_S), \phi(I_Q)$ # eq. 1;
$SC_1, V, R, T =$ Matching(S, Q, L_S) #eq. 2-6 + inline;
$L_1 = CE(SC_1, L_Q)$ #Stage 1 loss;
foreach *support/query pair of indices* (s, q) **do**
  $m_0 = max(V^{s,q})$, $BP^{s,q}_{S|Q}, BP^{s,q}_{Q|S} = 0$ #
  $\cdot^{s,q}$ = slice;
  while $max(V^{s,q}) >= \eta * m_0$ **do**
    $m_{xy} = argmax(V)$;
    $BP^{s,q}_{S|Q}, BP^{s,q}_{Q|S}$ += BP$(R^{s,q}, T^{s,q}, m_{xy})$ #eq. 7;
    $V^{s,q} = NMS(V^{s,q}, m_{xy})$ #suppress near m_{xy};
  end
end
$SC_2 = Stage2(S, Q, BP_{S|Q}, BP_{Q|S})$ #eq. 8-11;
$L_2 = CE(SC_2, L_Q)$ #Stage 2 loss;
$L = L_1 + L_2$ #final loss;

Note that as opposed to PN, our query ($F^P_{Q|c}$) and class ($F^P_{c|Q}$) prototypes are different for each query + class pair. Finally, the score of the second stage classifier for assigning label c to the query Q is:

$$SC_2(c; Q) = -||F^P_{Q|c} - F^P_{c|Q}||_2 \tag{10}$$

The predictions of the classifiers of the two stages of StarNet are fused using geometric mean to compute the joint prediction as ($sm = softmax$):

$$SC(c; Q) = \sqrt{sm(SC_1(c; Q)) \cdot sm(SC_2(c; Q))} \tag{11}$$

Implementation Details

Our implementation is in PyTorch 1.1.0 (Paszke et al. 2017), and is based on the public code of (Lee et al. 2019). In all experiments the CNN backbone is ResNet-12 with 4 convolutional blocks (in 2-stage StarNet we evaluated both single shared ResNet-12 backbone and a separate ResNet-12 backbone per stage). To increase the output resolution of the backbone we reduce the strides of some of its blocks. Thus, for benchmarks with 84×84 input image resolution, the block strides were $[2, 2, 2, 1]$ resulting in 10×10 feature grids, and for 32×32 input resolution (in Appendix), we used $[2, 2, 1, 1]$ strides resulting in 8×8 feature grids. This establishes naturally the value for r, we intend to explore other values in future work. We use four 1-shot, 5-way episodes per training batch, each episodes with 20 queries. The hyper-parameters $\sigma_f = 0.2$, $\sigma_g = 2$, and $\eta = 0.5$ were determined using validation. As in (Lee et al. 2019), we use 1000 batches per training epoch, 2000 episodes for validation, and 1000 episodes for testing. We train for 60 epochs, changing our base $LR = 1$ to $0.06, 0.012, 0.0024$

at epochs $20, 40, 50$ respectively. The best model for testing is determined by validation. On a single NVidia K40 GPU, our running times are: 1.15s/batch in 1-stage StarNet training; 2.2 s/batch in 2-stage StarNet training (in same settings (Lee et al. 2019) trains in 2.1s/batch); and 0.01s per query in inference. GPU peak memory was ~ 30MB per image.

Experiments

In all of experiments, only the class labels were used for training, validation, and for the support images of the test few-shot tasks. The bounding boxes were used only for performance evaluation. For each dataset we used the standard train / validation / test splits, which are completely disjoint in terms of contained classes. Only episodes generated from the training split were used for meta-training; the hyper-parameters and the best model were chosen using the validation split; and test split was used for measuring performance. Results on additional datasets are provided in Appendix.

The **CUB** fine-grained dataset (Wah et al. 2011) consists of $11,788$ images of birds of 200 species. We use the standard train, validation, and test splits, created by randomly splitting the 200 species into 100 for training, 50 for validation, and 50 for testing and used in all few-shot works. All images are downsampled to 84×84. Images are *not cropped* around the birds, which appear on cluttered backgrounds.

The **ImageNetLOC-FS** dataset (Karlinsky et al. 2019) contains 331 animal categories from ImageNetLOC (Russakovsky et al. 2015) split into: 101 for train, 214 for test, and 16 for validation. Since animals are typically photographed from afar, and as the images in this dataset are pre-processed to 84×84 square size with aspect ratio preserving padding (thus adding random padding boundaries), commonly images in this dataset are *not cropped* around the objects (some examples are in figure 1 bottom).

Weakly-Supervised Few-Shot Object Detection

We used ImageNetLOC-FS and CUB few-shot datasets, as well as PASCAL VOC (Everingham et al. 2010) experiment from (Wang et al. 2020), to evaluate StarNet performance on the proposed WS-FSOD task. All datasets have bounding box annotations, that in our case were used *only* for evaluating the detection quality. The ImageNetLOC-FS and the PASCAL VOC experiments allow comparing StarNet's performance directly to *Fully-Supervised* FSOD SOTA: RepMet (Karlinsky et al. 2019) and TFA (Wang et al. 2020) respectively, both serving as a natural performance upper bound for the *Weakly-Supervised* StarNet. Since, to the best of our knowledge, StarNet is the first method proposed for WS-FSOD, we also compare its performance to a wide range of weakly-supervised baselines.

Two baselines are based on a popular few-shot classifier MetaOpt (Lee et al. 2019) combined with GradCAM or SelectiveSearch (Uijlings et al. 2013) for localizing the classified categories. Third baseline is PCL (Tang et al. 2018a) - recent (non few-shot) WSOD method. Using official PCL code, we pre-trained it on the same training split as used for training StarNet, and adapted it by finetuning on support set of each of the test few-shot tasks. Fourth is the SOTA attention based few-shot method of CAN (Hou et al. 2019), that also has some ability to localize the objects. Finally, as a form of ablation, we offer two baselines evaluating the (non-parametric) StarNet head on top of ResNet-12 backbone that is: (i) randomly initialized, or (ii) pre-trained using a linear classifier. These baselines underline the importance of training the backbone end-to-end *through StarNet head* for the WS-FSOD higher gains. The results for WS-FSOD experiments and comparisons (averaged over 500 5-way test episodes) are summarized in Table 1, and qualitative examples of StarNet detections are shown in Figure 1(bottom).

For all methods and FS-WSOD experiments, we use the standard detection metric where detected bounding box is considered correct if its Intersection-over-Union (IoU) with a ground truth box is above threshold and its top-scoring class prediction is correct. We report Average Precision (AP) under this metric using 0.3 and 0.5 IoU thresholds. For all methods producing heatmaps, the bounding boxes were obtained using the CAM algorithm from (Zhou et al. 2016; Zhang et al. 2018) (as in most WSOD works).

StarNet results are higher by a large margin than results obtained by all the compared baselines. This is likely due to StarNet being directly end-to-end optimized for classifying images by detecting the objects within (using the proposed star-model geometric matching), while the other methods are either: not intended for few-shot (PCL), or optimized attention for classification and not for detection (CAN), or intended for classification and not detection (MetaOpt) - which cannot be easily bridged using the standard techniques for localization in classifiers (GradCAM, SelectiveSearch). As can be seen from Table 1, for $IoU \geq 0.3$ the StarNet is close to the fully supervised few-shot RepMet detector with about 10 AP points gap in 1-shot and about 7 points gap in 5-shot. However, the gap increases substantially for $IoU \geq 0.5$. We suggest that this gap is mainly due to partial detections (bounding box covering only part of an object) - a common issue with most WSOD methods. Analysis corroborating this claim is provided in the Appendix.

Finally, we performed (Wang et al. 2020)'s few-shot PASCAL VOC evaluation (three 5-way novel category sets), comparing to the fully-supervised (with boxes) SOTA FSOD method TFA proposed in that paper (Table 1 bottom). As TFA uses a (ResNet-101) backbone pre-trained on ImageNet (as common in FSOD works), in this experiment we used StarNet pre-trained on ImageNetLOC-FS (weakly supervised, without boxes) excluding PASCAL overlapping classes. Consistently with comparison to RepMet upper bound, under a more relaxed boxes tightness requirement of $IoU \geq 0.3$ (as discussed, used mostly due to partial detections), the AP of the weakly-supervised StarNet is close to the fully supervised TFA upper bound. Qualitative results from PASCAL experiment are provided in the Appendix.

Limitations StarNet detects multiple objects of different classes on the same query image via matching to different support images. It can also detect multiple instances of the same class via its (differentiable) NMS if their back-projection heatmap blobs are non-overlapping or if they are matched to different support images for that class. Yet in

dataset	method	1-shot $IoU \geq 0.3$	1-shot $IoU \geq 0.5$	5-shot $IoU \geq 0.3$	5-shot $IoU \geq 0.5$
Imagenet LOC-FS	RepMet (fully supervised *upper bound*)	59.5[(1)]	56.9	70.7[(1)]	68.8
	MetaOpt[(1)]+GC	32.4	13.8	51.9	22.1
	MetaOpt[(1)]+SS	16.1	4.9	27.4	10.2
	PCL[(1)] (Tang et al. 2018a)	25.4	9.2	37.5	11.3
	CAN[(1)] (Hou et al. 2019)	23.2	10.3	38.2	12.7
	random+StarHead	2.1	0.6	3.6	0.8
	pretrained+StarHead	22.9	10.2	31.0	21.3
	StarNet (ours)	**50.0**	**26.4**	**63.6**	**34.9**
CUB	MetaOpt[(1)]+GC	53.3	12.0	72.8	14.4
	MetaOpt[(1)]+SS	19.4	6.0	26.2	6.4
	PCL[(1)] (Tang et al. 2018a)	29.1	11.4	41.1	14.7
	CAN[(1)] (Hou et al. 2019)	60.7	19.3	74.8	26.0
	random+StarHead	3.5	0.6	6.0	0.9
	pretrained+StarHead	47.6	13.2	62.2	17.3
	StarNet (ours)	**77.1**	**27.2**	**86.1**	**32.7**
Pascal VOC (average over 5-way sets)	TFA (fully-supervised *upper bound*)	-	31.4	-	46.8
	StarNet (ours)	34.1	16.0	52.9	23.0

Table 1: WS-FSOD performance: comparing to baselines, performance measured in Average Precision (AP%). GC = GradCAM, SS = SelectiveSearch. RepMet (Karlinsky et al. 2019) and TFA (Wang et al. 2020) are fully-supervised upper bounds. [(1)]using official code and best hyper-parameters between defaults and those found by tuning on val. set for each benchmark.

some situations, if same class instances are overlapping on the query image and are matched to the same support image (as is bound to happen in 1-shot tests) - they would be detected as a single box by StarNet. Enhancing StarNet to detect overlapping instances of the same class is beyond the scope of this paper and an interesting future work direction.

Few-Shot Classification

StarNet is a WS-FSOD, trainable just from image class labels, and hence is readily applicable to standard few-shot classification testing. We used the standard few-shot classification evaluation protocol, exactly as in (Lee et al. 2019), using 1000 random 5-way episodes, with 1 or 5 shots. StarNet is optimized to classify the images *by finding the objects*, and hence has an advantage for benchmarks where objects appear at random locations and over cluttered backgrounds. Hence, as expected, StarNet attains large performance gains (of 4% and 5% above SOTA baselines in 1-shot setting) on CUB and ImageNetLOC-FS few-shot benchmarks, where images are less cropped around the objects. Notably, on these benchmarks we observe these gains also above the SOTA attention based and dense-matching based methods. The results of the evaluation, together with comparison to previous methods, are given in Table 2. Additional few-shot classification experiments showing StarNet's comparable performance on (cropped) *mini*ImageNet and CIFAR-FS few-shot benchmarks are provided in Appendix.

Ablation Study

We perform an ablation study to verify the contribution of the different components of StarNet and some of the design choices. We ablate using the 1-shot, 5-way CUB few-shot classification experiment, results are summarized in Table 3.

To test the contribution of object detection performed by the StarNet (stage-1), we use the same global average pooling for the prototype features as in StarNet stage-2, only without weighting by $BP_{Q|S}$ and $BP_{S|Q}$ (*'unattended stage-2'* in the table). We separately evaluate the performance of StarNet stage-1 and StarNet stage-2, this time stage-2 does use weighted pooling with $BP_{Q|S}$ and $BP_{S|Q}$. We then evaluate the full StarNet method (*'full StarNet'*). As expected we get a performance boost as this combines the structured (geometric) evidence from stage-1 with unstructured evidence pooled from the object regions in stage-2. Finally, using the NMS process to iteratively extend the back-projected query region matched to the support attains the best performance.

Conclusions

We have proposed a new Weakly-Supervised Few-Shot Object Detection (WS-FSOD) few-shot task, intended to significantly expedite building few-shot detectors for new visual domains, alleviating the need to obtain expensive bounding box annotations for a large number of base classes images in the new domain. We have introduced StarNet, a first WS-FSOD method. StarNet can also be used for few-shot classification, being especially beneficial for less-cropped objects in cluttered scenes and providing plausible explanations for its predictions by highlighting image regions corresponding to objects shared between the query and the matched support images. We hope that our work would inspire lots of future research on the important and challenging WS-FSOD task, further advancing its performance.

		ImageNetLOC-FS		CUB	
method	**backbone architecture**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
SAML (Hao et al. 2019)	conv4	-	-	69.35	81.56
Baseline[(1)] (Chen et al. 2019)	resnet-34	-	-	67.96	84.27
Baseline++[(1)] (Chen et al. 2019)	resnet-34	-	-	69.55	85.17
MatchingNet[(1)] (Vinyals et al. 2016)	resnet-34	-	-	73.49	86.51
ProtoNet[(1)] (Snell, Swersky, and Zemel 2017)	resnet-34	-	-	73.22	87.86
MAML[(1)] (Finn, Abbeel, and Levine 2017)	resnet-34	-	-	70.32	83.47
RelationNet[(1)] (Sung et al. 2018)	resnet-34	-	-	70.47	84.05
Dist. ensemble (Dvornik, Schmid, and Mairal 2019)	ensemble of 20× resnet18	-	-	70.07	85.2
Δ-encoder (Schwartz et al. 2018)	resnet-18	-	-	69.80	82.60
DeepEMD (Zhang et al. 2020)	resnet-12	-	-	75.65	88.69
CAN (Hou et al. 2019)	resnet-12	57.1[(2)]	73.9[(2)]	75.01[(2)]	86.8[(2)]
MetaOpt (Lee et al. 2019)	resnet-12	57.7[(2)]	74.8[(2)]	72.75[(2)]	85.83[(2)]
StarNet - shared backbone (ours)[(3)]	resnet-12	61.0	77.0	79.44	88.8
StarNet (ours)	2× resnet-12 = resnet-18	**63.0**	**78.0**	**79.58**	**89.5**

Table 2: Few-shot classification accuracy (%), for all methods the 0.95 confidence intervals are $< 1\%$ (omitted for brevity). For fair comparison, showing only results that *do not* use the validation set for training, *do not* use the transductive or semi-supervised setting, use standard input resolution 84×84, and *do not use* additional information such as class label or class attributes embedding. Results on additional few-shot classification benchmarks are provided in Appendix. [(1)]Results from (Chen et al. 2019), best result among resnet-10/18/34. [(2)]using official code and best hyper-parameters between defaults and those found by tuning on validation set for each benchmark. [(3)] shared backbone between StarNet stage-1 and stage-2.

unattended stage-2	72.92
StarNet stage-1	75.86
StarNet stage-2	76.74
full StarNet	78.78
full StarNet with iterative NMS	**79.58**

Table 3: Ablation study on CUB 1-shot / 5-way

Acknowledgments

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-19-C-1001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of DARPA. Raja Giryes was supported by ERC-StG grant no. 757497 (SPADE). The authors would also like to thank Assaf Arbelle, Rameswar Panda, and Richard Chen for helpful discussions.

Appendix

An extended version of the appendix is available on https://arxiv.org/abs/2003.06798.

Additional Qualitative Examples

Figure 4 shows some detection examples and some failure cases from different episodes and different novel category sets used in the PASCAL VOC WS-FSOD experiments described in the paper. Videos depicting the evolution of the back-projection heatmaps during StarNet training are available on Youtube https://tinyurl.com/4y7fdfvf .

Failure Cases Analysis - Partial Detections

A common weakness of WSOD methods is that the predicted bounding boxes cover only a part of the object, usually the most salient one. For pointing at objects, rather than exactly bounding them, the $IoU \geq 0.5$ matching criteria is too restrictive. To analyze whether partial detection are responsible for the AP drop observed for StarNet and all the baselines when moving from $IoU \geq 0.3$ to $IoU \geq 0.5$, we consider the following pair of related measures. For a ground truth (GT) bounding box G and a predicted box P we define $IoP = \frac{G \cap P}{P}$ (*Intersection over Predicted*) and $IoG = \frac{G \cap P}{G}$ (*Intersection over Ground Truth*). The IoP and IoG provide the precision and recall information, respectively, for object coverage. For equal-area P and G, the $IoU = 0.5$ corresponds to $IoP = \frac{2}{3}$. We use this intuition to substitute the $IoU \geq 0.5$ criterion with $IoP \geq \frac{2}{3}$, as a criterion better accounting for partial detection when computing the Average Precision (AP). The values of AP for $IoP \geq \frac{2}{3}$ are provided in Table 4.

The AP of StarNet, using $IoP = \frac{2}{3}$, is substantially higher than that computed for $IoU \geq 0.5$, corroborating our assumption that the performance drop between $IoU \geq 0.3$ and $IoU \geq 0.5$ is mostly due to partial detections. Additionally, we found that the StarNet bounding boxes that pass $IoP = \frac{2}{3}$ and have correct predicted class label still cover a significant portion of more than 32% of the GT boxes for objects on average. As can be seen from the table, for $IoP \geq \frac{2}{3}$ StarNet has large advantages over the baselines, consistent with StarNet's advantage observed for IoU-based criteria.

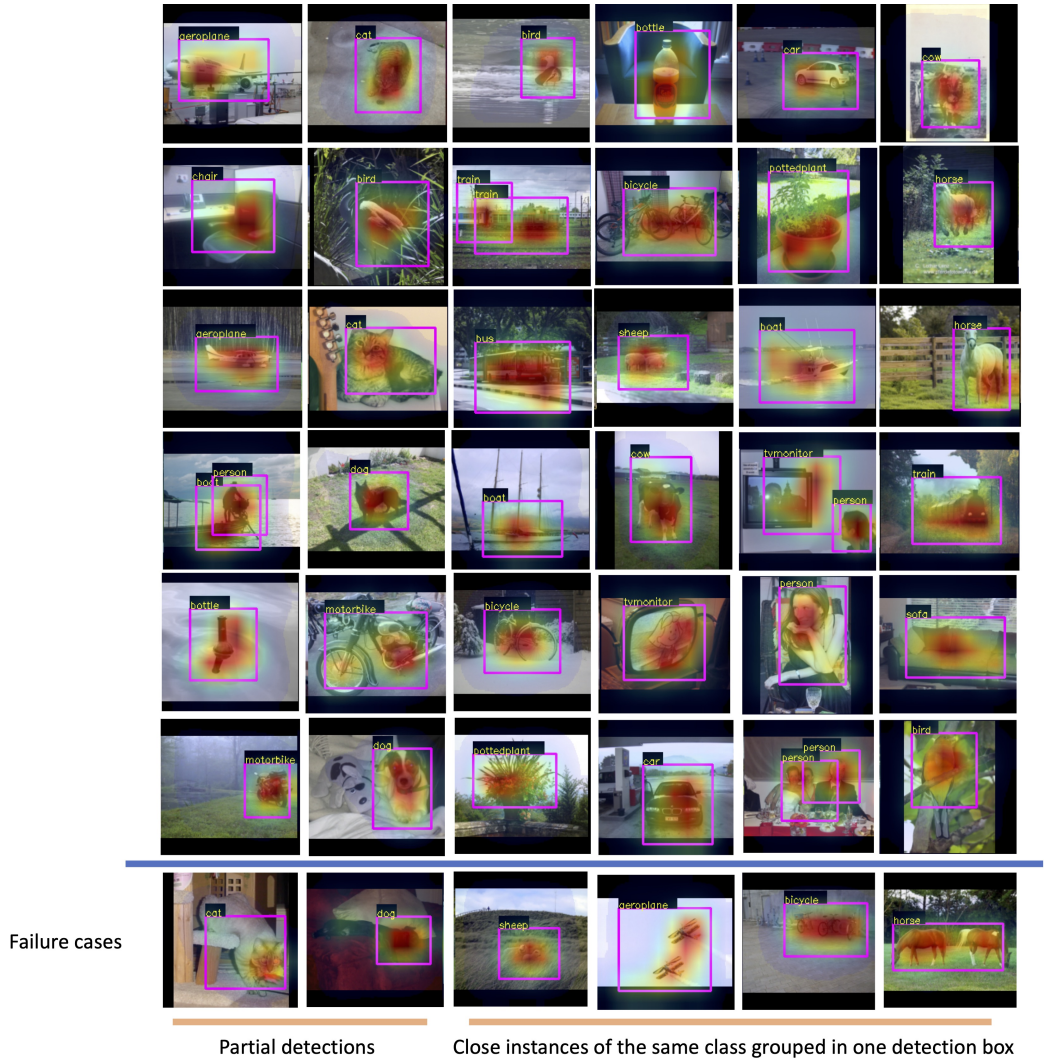


Figure 4: Example detections (on query images) from the PASCAL VOC WS-FSOD experiment described in the paper. Both detected object bounding boxes and a union of all their detected heatmaps produced by StarNet are visualized. Best viewed in color and in zoom.

		1-shot			5-shot		
dataset	method	$IoU \geq 0.3$	$IoU \geq 0.5$	$IoP \geq \frac{2}{3}$	$IoU \geq 0.3$	$IoU \geq 0.5$	$IoP \geq \frac{2}{3}$
Imagenet-LOC	MetaOpt+GC	32.4	13.8	29.2	51.9	22.1	41.4
	MetaOpt+SS	16.1	4.9	6.7	27.4	10.2	12.7
	PCL (Tang et al. 2018a)	25.4	9.2	23.8	37.5	11.3	34.3
	CAN (Hou et al. 2019)	23.2	10.3	20.1	38.2	12.7	35.1
	StarNet (ours)	**50.0**	**26.4**	**43.6**	**63.6**	**34.9**	**54.8**
CUB	MetaOpt+GC	53.3	12.0	52.5	72.8	14.4	62.6
	MetaOpt+SS	19.4	6.0	7.8	26.2	6.4	4.2
	PCL (Tang et al. 2018a)	29.1	11.4	29.0	41.1	14.7	37.0
	CAN (Hou et al. 2019)	60.7	19.3	55.4	74.8	26.0	66.1
	StarNet (ours)	**77.1**	**27.2**	**71.4**	**86.1**	**32.7**	**78.7**

Table 4: Average precision (AP, %) of weakly supervised few-shot detection and comparison to baselines on the ImagenetLOC-FS and CUB datasets. GC = GradCAM, SS = SelectiveSearch.

References

Alfassy, A.; Karlinsky, L.; Aides, A.; Shtok, J.; Harary, S.; Feris, R.; Giryes, R.; and Bronstein, A. M. 2019. LaSO: Label-Set Operations networks for multi-label few-shot learning. In *CVPR*.

Bilen, H.; and Vedaldi, A. 2016. Weakly Supervised Deep Detection Networks. *CVPR* 2846–2854.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics.

Chen, H.; Wang, Y.; Wang, G.; and Qiao, Y. 2018. LSTD: A Low-Shot Transfer Detector for Object Detection. *AAAI* .

Chen, W.-Y.; Liu, Y.-C.; Kira, Z.; Wang, Y.-C.; and Huang, J.-B. 2019. A Closer Look At Few-Shot Classification. In *ICLR*.

Choe, J.; Oh, S. J.; Lee, S.; Chun, S.; Akata, Z.; and Shim, H. 2020. Evaluating Weakly Supervised Object Localization Methods Right. In *CVPR*, 3130–3139.

Dvornik, N.; Schmid, C.; and Mairal, J. 2019. Diversity with Cooperation: Ensemble Methods for Few-Shot Classification. In *ICCV*.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision* 88(2): 303–338. ISSN 09205691. doi:10.1007/s11263-009-0275-4.

Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; and Ramanan, D. 2010. Object Detection with Discriminatively Trained Part Based Models. *PAMI* 32(9): 1627–1645.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*.

Gidaris, S.; Bursuc, A.; Komodakis, N.; Pérez, P.; and Cord, M. 2019. Boosting Few-Shot Visual Learning with Self-Supervision. In *ICCV*.

Gidaris, S.; and Komodakis, N. 2019. Generating Classification Weights with GNN Denoising Autoencoders for Few-Shot Learning. In *CVPR*.

Girshick, R.; Iandola, F.; Darrell, T.; and Malik, J. 2015. Deformable part models are convolutional neural networks. *CVPR* 437–446.

Hao, F.; He, F.; Cheng, J.; Wang, L.; Cao, J.; and Tao, D. 2019. Collect and Select : Semantic Alignment Metric Learning for Few-Shot Learning. *ICCV* 8460–8469.

Hou, R.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2019. Cross Attention Network for Few-shot Classification. *NeurIPS* .

Hu, T.; Mettes, P.; Huang, J.-H.; and Snoek, C. G. M. 2019. SILCO : Show a Few Images , Localize the Common Object. *ICCV* .

Kang, G.; Jiang, L.; Yang, Y.; and Hauptmann, A. G. 2019. Contrastive Adaptation Network for Unsupervised Domain Adaptation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June: 4888–4897. URL http://arxiv.org/abs/1901.00976.

Karlinsky, L.; Shtok, J.; Harary, S.; Schwartz, E.; Aides, A.; Feris, R.; Giryes, R.; and Bronstein, A. M. 2019. RepMet: Representative-based metric learning for classification and one-shot object detection. *CVPR* 5197–5206. URL http://arxiv.org/abs/1806.04728.

Karlinsky, L.; Shtok, J.; Tzur, Y.; and Tzadok, A. 2017. Fine-grained recognition of thousands of object categories with single-example training. *CVPR* 965–974.

Kim, J.; Kim, T.; Kim, S.; and Yoo, C. D. 2019. Edge-Labeling Graph Neural Network for Few-shot Learning. In *CVPR*.

Lee, K.; Maji, S.; Ravichandran, A.; and Soatto, S. 2019. Meta-Learning with Differentiable Convex Optimization. In *CVPR*.

Leibe, B.; Leonardis, A.; and Schiele, B. 2006. An Implicit Shape Model for Combined Object Categorization and Segmentation. In *Toward Category-Level Object Recognition*, May, 508–524.

Li, H.; Eigen, D.; Dodge, S.; Zeiler, M.; and Wang, X. 2019a. Finding Task-Relevant Features for Few-Shot Learning by Category Traversal 1. URL http://arxiv.org/abs/1905.11116.

Li, X.; Sun, Q.; Liu, Y.; Zheng, S.; Zhou, Q.; Chua, T.-S.; and Schiele, B. 2019b. Learning to Self-Train for Semi-Supervised Few-Shot Classification. In *NeurIPS*, 1–14.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. In *arXiv:1707.09835*.

Lifchitz, Y.; Avrithis, Y.; Picard, S.; and Bursuc, A. 2019. Dense Classification and Implanting for Few-Shot Learning. In *CVPR*.

Lin, T. Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common objects in context. In *Lecture Notes in Computer Science*, volume 8693 LNCS, 740–755.

Lin, T.-Y.; Roychowdhury, A.; and Maji, S. 2017. Bilinear CNNs for Fine-grained Visual Recognition. *TPAMI* .

Liu, L.; Muelly, M.; Deng, J.; Pfister, T.; and Li, J. 2019. Generative Modeling for Small-Data Object Detection. In *ICCV*.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C. Y.; and Berg, A. C. 2016. SSD: Single shot multibox detector. *Lecture Notes in Computer Science* 9905 LNCS: 21–37.

Maji, S.; and Malik, J. 2009. Object Detection using a Max-Margin Hough Transform. In *CVPR*.

Oreshkin, B. N.; Rodriguez, P.; and Lacoste, A. 2018. TADAM: Task dependent adaptive metric for improved few-shot learning. *NeurIPS* .

Paszke, A.; Chanan, G.; Lin, Z.; Gross, S.; Yang, E.; Antiga, L.; and Devito, Z. 2017. Automatic differentiation in PyTorch 1–4.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep Hough Voting for 3D Object Detection in Point Clouds. In *ICCV*.

Qiao, L.; Shi, Y.; Li, J.; Wang, Y.; Huang, T.; and Tian, Y. 2019. Transductive Episodic-Wise Adaptive Metric for Few-Shot Learning. In *ICCV*.

Ravi, S.; and Larochelle, H. 2017. Optimization As a Model for Few-Shot Learning. *ICLR* 1–11.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Neural Information Processing Systems (NIPS)* ISSN 0162-8828. doi:10.1109/TPAMI.2016.2577031.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* URL http://arxiv.org/abs/1409.0575.

Rusu, A. A.; Rao, D.; Sygnowski, J.; Vinyals, O.; Pascanu, R.; Osindero, S.; and Hadsell, R. 2018. Meta-Learning with Latent Embedding Optimization. In *ICLR*.

Sali, E.; and Ullman, S. 1999. Combining Class-Specific Fragments for Object Classification. In *BMVC*.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; and Bronstein, A. M. 2018. Delta-Encoder: an Effective Sample Synthesis Method for Few-Shot Object Recognition. *NeurIPS* .

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *ICCV* 618–626.

Shaban, A.; Rahimi, A.; Bansal, S.; Gould, S.; Boots, B.; and Hartley, R. 2019. Learning to Find Common Objects Across Few Image Collections. In *ICCV*, 5117–5126.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In *NIPS*.

Sung, F.; Yang, Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. 2018. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*, 1199–1208. ISBN 9781538664209. ISSN 10636919. doi:10.1109/CVPR.2018.00131.

Tang, P.; Wang, X.; Bai, S.; Shen, W.; Bai, X.; Liu, W.; and Yuille, A. 2018a. PCL: Proposal Cluster Learning for Weakly Supervised Object Detection. *PAMI* 42(1): 176–191.

Tang, P.; Wang, X.; Wang, A.; Yan, Y.; Liu, W.; Huang, J.; and Yuille, A. 2018b. Weakly Supervised Region Proposal Network and Object Detection. In *ECCV*. ISBN 9783030012519. ISSN 16113349. doi:10.1007/978-3-030-01252-6{_}22.

Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. *IJCV* 104(2): 154–171.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Kavukcuoglu, K.; and Wierstra, D. 2016. Matching Networks for One Shot Learning. *NIPS* .

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset 1–15. doi:CNS-TR-2010-001.2010. URL http://authors.library.caltech.edu/27468/.

Wang, X.; Huang, T. E.; Darrell, T.; Gonzalez, J. E.; and Yu, F. 2020. Frustratingly Simple Few-Shot Object Detection. In *ICML*. URL http://arxiv.org/abs/2003.06957.

Wang, Y.-X.; Ramanan, D.; and Hebert, M. 2019. Meta-Learning to Detect Rare Objects. In *The IEEE International Conference on Computer Vision (ICCV)*, 9925–9934.

Wertheimer, D.; and Hariharan, B. 2019. Few-Shot Learning with Localization in Realistic Settings URL http://arxiv.org/abs/1904.08502.

Xing, C.; Rostamzadeh, N.; Oreshkin, B. N.; and Pinheiro, P. O. 2019. Adaptive Cross-Modal Few-Shot Learning. In *NeurIPS*.

Zeng, Z.; Liu, B.; Fu, J.; Chao, H.; and Zhang, L. 2019. WSOD2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. *CVPR* 8291–8299.

Zhang, C.; Cai, Y.; Lin, G.; and Shen, C. 2020. DeepEMD: Few-Shot Image Classification with Differentiable Earth Mover's Distance and Structured Classifiers. In *CVPR*.

Zhang, H.; Zhang, J.; and Koniusz, P. 2019. Few-shot learning via saliency-guided hallucination of samples. *CVPR* 2019-June: 2765–2774.

Zhang, J.; Zhao, C.; Ni, B.; Xu, M.; and Yang, X. 2019. Variational Few-Shot Learning. In *IEEE International Conference on Computer Vision (ICCV)*.

Zhang, X.; Wei, Y.; Kang, G.; Yang, Y.; and Huang, T. 2018. Self-produced guidance for weakly-supervised object localization. *Lecture Notes in Computer Science* 610–625.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*, 2921–2929.

Zhou, F.; Wu, B.; and Li, Z. 2018. Deep Meta-Learning: Learning to Learn in the Concept Space. Technical report. URL https://arxiv.org/abs/1802.03596.