

StarNet: towards weakly supervised few-shot detection and explainable few-shot classification

Leonid Karlinsky^{*1}, Joseph Shtok^{*1}, Amit Alfassy^{*1,3}, Moshe Lichtenstein^{*1},
 Sivan Harary¹, Eli Schwartz^{1,2}, Sivan Doveh¹, Prasanna Sattigeri¹,
 Rogerio Feris¹, Alexander Bronstein³, and Raja Giryes²

IBM Research AI¹, Tel-Aviv University², Technion³

Abstract. In this paper, we propose a new few-shot learning method called StarNet, which is an end-to-end trainable non-parametric star-model few-shot classifier. While being meta-trained using only image-level class labels, StarNet learns not only to predict the class labels for each query image of a few-shot task, but also to localize (via a heatmap) what it believes to be the key image regions supporting its prediction, thus effectively detecting the instances of the novel categories. The localization is enabled by the StarNet’s ability to find large, arbitrarily shaped, semantically matching regions between all pairs of support and query images of a few-shot task. We evaluate StarNet on multiple few-shot classification benchmarks attaining significant state-of-the-art improvement on the CUB and ImageNetLOC-FS, and smaller improvements on other benchmarks. At the same time, in many cases, StarNet provides plausible explanations for its class label predictions, by highlighting the correctly paired novel category instances on the query and on its best matching support (for the predicted class). In addition, we test the proposed approach on the previously unexplored and challenging task of Weakly Supervised Few-Shot Object Detection (WS-FSOD), obtaining significant improvements over the baselines.

Keywords: Few-Shot Learning, Weakly-Supervised Few-Shot Detection

1 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 training. In most, if not all, few-shot learning methods (e.g., [18,44,21]), the training is performed on a dataset of ‘base’ classes with a large number of labeled examples for each class. In different few-shot learning applications, label complexity varies from image-level class labels (classification), to labeled boxes (detection), to labeled pixel-masks (segmentation).

It has been shown in [6] that for few-shot learning methods to be effective (even w.r.t. the naive baselines), the base classes need to come from the same

^{*} Equal contribution



Fig. 1. StarNet provides evidence for its predictions by finding large (semantically) matching regions between the query and the support images of a few-shot task, thus providing plausible explanations even for ambiguous cases. Matching regions are drawn as heatmaps for each query (top) and support (bottom) pair.

‘visual domain’ as the novel target classes (i.e., base and novel classes are required to be visually similar in a sense). That said, for few-shot applications which require richer annotation (detection, segmentation), entering new visual domains is still prohibitively expensive due to many thousands of base classes images that need to be annotated in order to meta-train the few-shot approach for the new domain. Few-shot learners that focus on classification require less annotation efforts, but are only able to produce image-level class predictions.

Generic methods, based on differentiating the category score with respect to intermediate level features, such as the popular GradCAM [41], are able (to some extent) to highlight the pixels responsible for the CNN classifier’s prediction. But, as illustrated in Figure 3, and evaluated in Table 2, these are generally much less effective for few-shot classifiers that need to predict novel classes learned from only the handful of support examples available in a typical few-shot task.

In this paper, we propose StarNet - a new type of few shot learner that is capable of localizing the novel class instances while being trained for few-shot classification using only image-level class labels. StarNet is comprised of a novel few-shot classifier head, attached to a standard fully-convolutional CNN feature extractor (e.g. ResNet-12 [11]). The CNN is stripped of the last FC layers and modified to increase the output resolution of its last convolutional layer. StarNet is trained in meta-learning fashion, where k -shot, n -way training episodes are randomly sampled from the base classes data. Each episode consists of k random support samples and q random query samples for each of n random base classes.

StarNet head operates by trying to geometrically match every pair of support and query images. For each pair, a non-parametric star-model [39,19,16] matching is implemented as an end-to-end differentiable neural network. The gradients flowing through the StarNet head drive the CNN backbone to learn (end-to-end) the optimal features for this matching. A star-model is a probabilistic model where the object parts are generated independently given some latent state variables describing the object (e.g. object center location and size).

The star-model allows matching (maximal) partial regions of arbitrary shape between the two images up to arbitrary shift and local deformations accommodating for changes in shape. In StarNet, the training drives the matched regions to correspond to the locations of the class instances present on support and query pairs that share the same class label, thus localizing these instances. Highlighting the found common regions on the query and matched supports also provides an explanation for the model’s prediction. For example, each of the two query images shown in the top of Figure 1 contains instances of two novel classes (‘turkey’, ‘parrot’, ‘chicken’, ‘dog’). Clearly, in this situation there is no single correct class label for these queries. Nevertheless, StarNet can successfully highlight the matched class instances on both the query and the support images for different classes, thus providing plausible explanations for the different possible labels.

In our experiments, we explore how the proposed StarNet few-shot learner can be used to approach a new and challenging task of *Weakly Supervised Few-Shot Object Detection* (WS-FSOD). In WS-FSOD, we train a few-shot object detector using only classification data in base training, and are able to detect instances of novel classes (unseen during training) without location information provided for support samples of these classes in the test few-shot tasks. In addition, we show that StarNet few-shot learner is effective at few-shot classification, significantly improving the state-of-the-art (SOTA) baselines on the CUB [50] and ImageNetLOC-FS [15] few-shot benchmarks, and comparing favorably to the SOTA methods on: *miniImageNet* [49], CIFAR-FS [2] and FC100 [30]. At

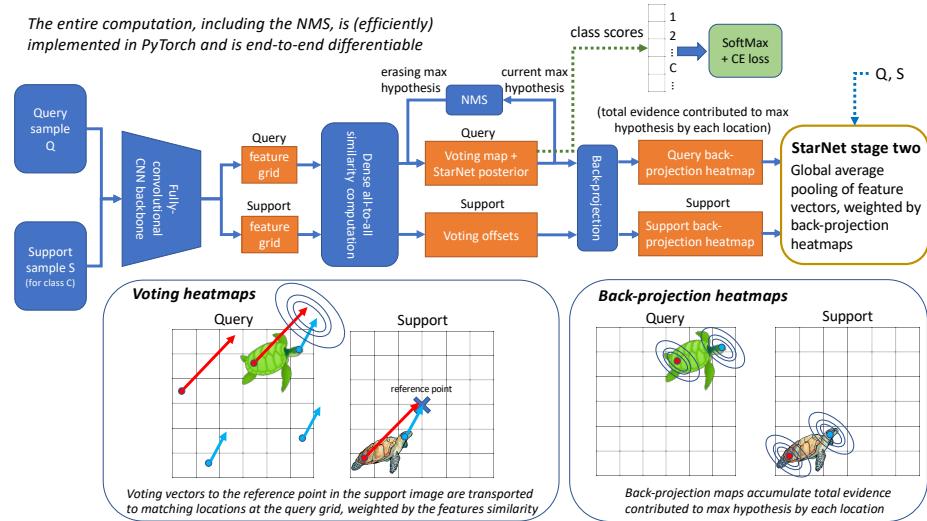


Fig. 2. StarNet overview. Query image Q is being matched to a candidate support image S jointly localizing instances of a shared category (if exist). The maximal hypothesis score is used as logit for the class of S for training or inference. The NMS module iteratively suppresses the max hypothesis allowing to match multiple non-rigid parts of the same object or multiple objects. The entire process is end-to-end differentiable.

the same time, StarNet provides plausible explanations for its predictions by highlighting the matching regions it finds for query-support pairs.

To summarize, our contributions in this paper are the following: (1) we propose a new few-shot classifier head architecture - the StarNet, that classifies query images by direct geometric matching (allowing local deformations) with the support images, implementing the non-parametric star-model as an end-to-end differentiable neural network. StarNet improves SOTA approaches by a large margin in few-shot classification benchmarks with images less cropped around the objects, and also compares favorably in other benchmarks; (2) StarNet is capable of providing some visual explanations for its predictions by highlighting the region on the query that matched the support of the chosen category (and also highlighting the matched region on the support); (3) we explore how the StarNet approach can be used for WS-FSOD - a new, challenging, and previously unexplored task in few-shot learning, and provide promising results for this task significantly improving over the baselines.

2 Related Work

In this section we briefly review the modern few-shot learning focusing on the meta-learning methods, survey the classical and modern star-model (or generalized Hough transform) methods, discuss weakly-supervised detection works, and review methods for localization and detection in few-shot learning.

Meta-learning methods learn from few-shot tasks (or episodes) rather than from individual labeled samples. Each such task is a small dataset, 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 *metric learning based few-shot* methods [49,44,45,20], a deep metric space (embedding) is meta-trained, for distance-based classification in the test. The metric may be further adapted to the support set of the new task like in [49], or learned optimal comparison between query and support samples as in [45], or using a latent space attention module as in [20].

Gradient-based meta learners [8,21,21,55,36,29,38] search for models that are good initialization for transfer to novel few-shot tasks. Typically, in these methods higher order derivatives are used for meta-training, optimizing the loss the model would have after applying one or several gradient steps. At test time, the model is fine-tuned to the novel few shot-tasks. In [6] the *baseline* and *baseline++* were introduced for evaluating different forms of simple fine-tuning as a simple meta-learner alternative. In [7] ensemble methods for few-shot learning are evaluated. In [40] auto-encoder like network is used to expand the support set with synthetic examples. In [1] synthetic examples are generated in multi-label few-shot setting. In TADAM [30] a each convolutional layer output is scaled and shifted adaptively to task. In MetaOptNet [18] an SVM solver on top of CNN backbone is posed as an end-to-end neural network. In TEAM [35] transductive learning is used, allowing query samples to learn from each other. In [53], a class variational approach is used by estimating the distributions for the novel categories in the latent space.

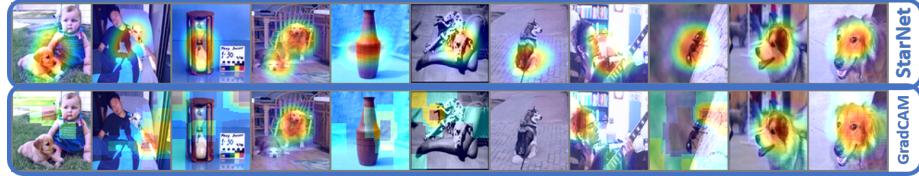


Fig. 3. Comparison with GradCAM: StarNet back-projection maps (top row) and GradCAM [41] attention maps (bottom row) computed for MetaOptNet+SVM [18] on *miniImageNet* test episodes queries. GradCAM failures are likely due to the few-shot setting, or presence of multiple objects. Best viewed in color and zoom.

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 represented as a collection of parts, independently linked to the object model via Gaussian priors to allow local deformations. Classically, parts were represented using patch descriptors [39,19,27,16], or SVM part detectors in the Deformable Part Model (DPM) [33]. DPM was later extended to CNN based DPM in [9]. Recently, in [34] GHT based approach was posed as a neural network object detector in 3D point clouds, in the fully supervised and non-few-shot setting. Unlike DPM [33,9], StarNet is non-parametric, in a sense that parts are not explicit and are not fixed during inference, and unlike all of the aforementioned methods [39,19,27,33,9,16,34], it is trained with only weak supervision to work in the few-shot setting. In [24] 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 [39,19,27,33], StarNet features used for local matching are not handcrafted, but are rather end-to-end optimized (for matching novel class instances in few-shot setting) by a CNN backbone.

Weakly-supervised object detection refers to techniques that learn to localize objects despite being trained with only image-level class labels [3,47,31]. However, to the best of our knowledge, no prior works have considered this challenging problem in the few-shot setting.

Few-shot with localization and attention is a recent and less explored research direction. Most of these methods rely on bounding box supervision. Using (full) labeled bounding boxes supervision, several works have extended the popular object detection techniques [43,26] to the few-shot setting [5,15,14,51,25]. Recently, several works have further explored the use of geometric relations between intermediate convolutional features in few-shot classification. In SAML [10], last convolutional layer outputs are compared between all pairs of query and support samples, and the resulting relations matrix is fed into an MLP for final classification. In [52] features for the support and query images are collected using an attention mask coming from a localizer module trained using object bounding boxes. SILCO [13] addresses localization of novel class objects using 5 support examples in 1-way / 5-shot mode, the method is trained using bounding box supervision. In [42] Multiple Instance Learning paradigm is used

to co-localize common objects in small image collections, their box proposals are generated by an RPN pre-trained on MS-COCO [23]. In CAN [12] 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 [12] is not modeled. In DC [22] a classifier is applied densely on each of the local features in the feature map.

3 Method

In this section we provide the details of the StarNet method and its variants. In Section 3.1 we explain the basic 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. In Section 3.2 we explain how we can revert the StarNet posterior computation and obtain evidence maps (on both query and support) for any hypothesis using back-projection. In Section 3.3 we explain how to enhance the StarNet performance, by adding a second-stage classifier utilizing the evidence maps to pool features from the matched regions (in query and in support images) in the attention-like manner, effectively suppressing background clutter. Finally, in section 3.4 we provide the implementation details and running times. Figure 2 provides an overview of our approach. Our code will be released upon acceptance.

3.1 Single-stage StarNet

Denote by Q and S a pair of query and support images belonging to a k -shot, n -way episode E (a few-shot task) sampled during either meta-training or meta-testing. 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:

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

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:

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

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

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



Fig. 4. Examples of regions matched by StarNet between support and query images in few-shot episodes of different datasets. Top 3 rows shows a mix of examples from miniImageNet, ImageNetLOC-FS, and CUB. Bottom row shows some query examples from CIFAR-FS and FC100 with localized objects, despite the tiny 32×32 resolution. Best viewed in color and in zoom.

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, the tensor 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:

$$N_{i,j} = \sum_S \sum_{l,m} P_{i,j,l,m} \quad (4)$$

where \sum_S is computed by matching the same query Q to all of the supports in the episode. Note that P are unnormalized probabilities, but the normalization factor cancels out when we compute the likelihood ratios. Let w be a reference point on S . We set $w = (r/2, r/2)$ to be 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 $S_{l,m}$. Please note that by construction, $t_{i,j,l,m}$ can be negative, and its values range 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 .

Our proposed StarNet implements the non-parametric star-model [39,19,16] for matching the Q and S feature grids. For every hypothesis location (x, y) on the $2r \times 2r$ hypothesis grid, the star model computes the overall belief $A(x, y)$ for that hypothesis considering independently all the possible matches between support and query feature grids. In probabilistic sense, star-model is a variant of the Naive-Bayes model [4], and hence to compute $A(x, y)$, should have accumulated *log*-likelihood ratios from the potential matches. However, as in [16], to make the star-model more robust to background noise, in StarNet likelihood ratios are directly accumulated:

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

Following accumulation, the final StarNet posterior for each location hypothesis is computed by convolution of A with a symmetric Gaussian kernel $G(\sigma_g)$:

$$V(x, y) = G(\sigma_g) \circledast A(x, y) \quad (6)$$

this efficiently accounts for any random relative location shift allowed to occur with the $G(\sigma_g)$ Gaussian prior for any matched pair of $Q_{i,j}$ and $S_{l,m}$.

For the following, we will slightly abuse notation and denote by $V_{Q,S}(x, y)$ the StarNet posterior $V(x, y)$ computed for the pair of query image Q and support image S from the k -shot, n -way episode E . We compute the score (logit) of predicting the category label c for Q as:

$$SC_1(c; Q) = \frac{1}{k} \cdot \sum_{\substack{S \in E \text{ s.t.} \\ C(S)=c}} \max_{x,y} V_{Q,S}(x, y) \quad (7)$$

where $C(S)$ is the class label of S . 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 ground truth category label of Q in the training episode. The need to only match images with the same 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 (figure 4).

3.2 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 $(\hat{x}, \hat{y}) = \arg \max_{x,y} V(x, y)$, 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 the corresponding feature grid cell on Q or S gave to the posterior probability $V(\hat{x}, \hat{y})$:

$$BP_{Q|S}(i, j) = \sum_{l, m} R_{i, j, l, m} \cdot e^{-0.5 \cdot \|t_{i, j, l, m} - (\hat{x}, \hat{y})\|^2 / \sigma_g^2} \quad (8)$$

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 that share the same category labels are in most cases the instances of that category (figure 4).

This back-projection process can be iteratively repeated by suppressing (\hat{x}, \hat{y}) (and its 3×3 neighborhood) in $V(x, y)$ as part of the Non-Maximal Suppression (NMS) process, also implemented as part of the neural network. NMS allows for better coverage of the localized objects (detected as sum of parts) and for discovering additional object instances of the same category. Please see Figures 4 and 3 (image 4, top row) for some examples of multiple objects highlighted by StarNet. In our implementation, we repeat the NMS until the next maximal point is less than an $\eta = 0.5$ from the original maximal value.

3.3 Two-stage StarNet

Having computed the $BP_{Q|S}$ and $BP_{S|Q}$ back-projection heatmaps, we follow the best practice of the modern 2-stage CNN detectors, such as FasterRCNN [43], to enhance the StarNet performance with a second stage classifier that benefits 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:

$$F_{Q|S} = \sum_{i, j} BP_{Q|S}(i, j) \cdot Q_{i, j} \quad (9)$$

$$F_{S|Q} = \sum_{l, m} BP_{S|Q}(l, m) \cdot S_{l, m} \quad (10)$$

here the feature grids $Q_{i, j}$ and $S_{l, m}$ can be computed using ϕ as above, or using a separate CNN backbone trained jointly with the first stage network. Our second stage is a variant of the Prototypical Network (PN) classifier [44]. We compute the prototype for class c and the query Q embedding to be compared to it as:

$$F_{c|Q}^P = \frac{1}{k} \cdot \sum_{S \in E \text{ s.t. } C(S)=c} F_{S|Q} \quad (11)$$

$$F_{Q|c}^P = \frac{1}{k} \cdot \sum_{S \in E \text{ s.t. } C(S)=c} F_{Q|S} \quad (12)$$

Note that as opposed to PN, our query embedding and class prototypes are jointly query and class dependent. For a given query Q , we have a different

query embedding $F_{Q|c}^P$ and a different class prototype $F_{c|Q}^P$ for each class-query pair. Finally, the score (logit, input to SoftMax) of the second stage classifier for assigning label c to the query Q is computed as:

$$SC_2(c; Q) = -\|F_{Q|c}^P - F_{c|Q}^P\|_2 \quad (13)$$

In our experiments, whenever 2-stage StarNet is used, we compute its final score as a geometric mean of the first and the second stage classifiers ($sm = softmax$):

$$SC(c; Q) = \sqrt{sm(SC_1(c; Q)) \cdot sm(SC_2(c; Q))} \quad (14)$$

3.4 Implementation details

Our implementation is written in PyTorch 1.1.0 [32], and is based on the public code of [18]. In all experiments the CNN backbone is ResNet-12 with 4 convolutional blocks. 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, we used $[2, 2, 1, 1]$ strides resulting in 8×8 feature grids. We use four 1-shot, 5-way episodes per training batch, each episodes with 20 queries. Using validation we set $\sigma_f = 0.2$ and $\sigma_g = 2$. As in [18], we use 1000 batches per training epoch, 2000 episodes for validation, and 1000 episodes for testing. We run validation (on the validation set) at the end of every epoch, and it is used to determine the best model to retain for testing. Using a single NVidia K40 GPU, our running times are: 1.15s/batch for single-stage StarNet training; 2.2 s/batch for 2-stage StarNet training (on the same GPU and batch size [18] trains in 2.1s/batch); and 0.01s per query during inference. Train-time GPU peak memory use was ~ 30 MB per image (including gradients).

4 Experiments

4.1 Datasets

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 object bounding boxes were used only to evaluate the predicted bounding boxes in weakly-supervised few-shot detection experiments. For each dataset we used the standard train / validation / test splits, which are completely disjoint in terms of contained classes. Episodes generated from the training split were used for meta-training (from scratch); the hyper-parameters and the best model were chosen using the validation split; and test split was used for measuring the performance.

The **miniImageNet** dataset [49] has 100 randomly chosen classes from ILSVRC-2012 [37]. They are randomly split into 64 meta-training, 16 meta-validation, and 20 meta-testing classes. Each class has 600 84×84 images.

The **CIFAR-FS** dataset [2], consists of all 100 classes from CIFAR-100 [17]. The classes are randomly split into 64, 16 and 20 for meta-training, meta-validation, and meta-testing respectively. Each class contains 600 32×32 images.

Table 1. Few-shot classification results, in % (accuracy). For fair comparison, showing only results that *do not* use the validation set for training, *do not* use the transductive setting, use standard input resolution, use plain ResNet backbones (ours uses ResNet-12, others use same or deeper ResNets), and do not use additional information such as class label or class attributes embedding. ⁽¹⁾As we base our implementation on MetaOpt public code, for fair comparison MetaOpt results are reported as they are reproduced using their public code using their reported optimal hyper-parameters; MetaOpt paper reported results are in parenthesis. ⁽²⁾Results according to [6]. ⁽³⁾using official code. RT: Run Time per query in a 5-way 1-shot task (ours on one NVIDIA K40 GPU)

method	ImageNetLOC-FS		CUB		<i>mini</i> ImageNet		CIFAR-FS		FC100		RT(s)
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	
Baseline [6]	-	-	67.96	84.27	52.37	74.69	-	-	-	-	-
Baseline++ [6]	-	-	69.55	85.17	53.97	76.16	-	-	-	-	-
MatchingNet ⁽²⁾ [49]	-	-	72.36	83.78	54.49	68.88	-	-	-	-	0.008
MAML ⁽²⁾ [8]	-	-	72.36	83.78	54.69	66.62	-	-	-	-	0.103
ProtoNet ⁽²⁾ [44]	-	-	72.03	87.42	54.16	74.65	-	-	-	-	0.018
RelationNet ⁽²⁾ [45]	-	-	68.65	82.75	52.48	69.83	-	-	-	-	0.025
SNAIL [28]	-	-	-	-	55.71	68.88	-	-	-	-	-
SAML [10]	-	-	69.35	81.56	57.69	73.03	-	-	-	-	0.02
TADAM [30]	-	-	-	-	58.50	76.70	-	-	40.1	56.1	0.079
Variat. FSL [53]	-	-	-	-	61.23	77.69	-	-	-	-	-
FSL with Loc. [52]	-	-	-	-	51.1	69.45	-	-	-	-	-
CTM [20]	-	-	-	-	62.05	78.63	-	-	-	-	0.01
Dist. ensemble [7]	-	-	68.77	83.57	59.38	76.9	-	-	-	-	-
Δ -encoder [40]	-	-	69.80	82.60	59.90	69.70	66.70	79.80	-	-	0.10
DC [22]	-	-	-	-	61.26	79.01	-	-	42.04	57.05	-
CAN [12]	57.1 ⁽³⁾	73.9	75.01 ⁽³⁾	86.8	63.85	79.44	-	-	-	-	0.04
MetaOpt ⁽¹⁾ [18]	57.7 ⁽³⁾	74.8	72.75 ⁽³⁾	85.83	61.77	77.9	71.7	84.12	41.37	55.3	0.075
				(62.64)	(78.63)	(72.6)	(85.3)	(41.1)	(55.5)		
StarNet (ours)	63.0	78.0	79.58	89.5	62.33	79.6	72.83	85.5	40.5	57.1	0.01

The **FC100** dataset [30] is another dataset constructed from CIFAR-100 [17], it contains 100 classes which are grouped into 20 superclasses, partitioned into groups of (12, 4, 4) superclasses for meta training, validation and testing, accordingly. Each class contains 600 32×32 images.

The **CUB** fine-grained dataset [50] consists of 11,788 images of birds of 200 species. We use the standard train, validation, and test splits, which were created by randomly splitting the 200 species into 100 for training, 50 for validation, and 50 for testing. All images are downsampled to 84×84 . Images are generally not-cropped around the birds and appear on cluttered backgrounds.

The **ImageNetLOC-FS** dataset [15] contains 331 animal categories from ImageNetLOC [37] split into 3 disjoint sets: 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 in figure 5).

4.2 Few-shot classification

We evaluated the StartNet few-shot classification performance on all of the benchmarks listed in section 4.1. Standard protocol, exactly as in [18], were used for this evaluation. The results of the evaluation, together with comparison to previous methods, are summarized in Table 1. All the performance numbers are given in accuracy %; for all methods the 0.95 confidence intervals are less than 1% and hence are omitted for brevity. The tests are performed on a 1000 random 5-way episodes, with 1 or 5 shots (number of support examples per class). For fair comparison, in Table 1 we only list results that were obtained: (1) without using validation data for training; (2) without using transductive setting; (3) with images at standard 84×84 (or 32×32) resolution; (4) using plain ResNet backbones (we use ResNet-12); and (5) not using additional semantic information such as class label or class attributes embedding.

StarNet main performance gains, of over 4.5% and 5% above SOTA baselines in 1-shot setting, are observed in the CUB and ImageNetLOC-FS respectively. This is expected, as StarNet is optimized to classify the objects through their localization, and hence has an advantage for benchmarks with images that are less cropped around the objects. Interestingly, we observe these gains also above the SOTA attention based method of [12] on these benchmarks.

4.3 Weakly-Supervised Few-Shot Object Detection (WS-FSOD)

To test the StarNet’s ability for WS-FSOD, we have used ImageNetLOC-FS [15] and CUB [50] datasets. Both have bounding box annotations, that in our case were used only for evaluating the detection quality. The ImageNetLOC-FS [15] is a diverse dataset with over 200 test categories, also allowing to compare StarNet’s WS-FSOD performance directly to the performance of the fully-supervised RepMet few-shot object detector [15], which serves as a natural performance upper bound. For the CUB, the same split as for the few-shot classification experiments was used. Since, to the best of our knowledge, StarNet is the first method proposed for WS-FSOD, we compare its performance to several baselines. The first two baselines are based on one of the SOTA few-shot classifiers MetaOpt [18] combined with GradCAM or SelectiveSearch [48] for localizing the classified categories. In addition, we compare to PCL [46] - recent WSOD method pre-trained on the same training split as used for training StarNet and adapted using finetuning to novel classes in test few-shot tasks. Finally, we also compare to the SOTA attention based few-shot method of CAN [12], that also has some ability to localize the objects. The results for WS-FSOD experiments and comparisons (averaged over 500 test episodes) are summarized in Table 2, and qualitative examples of StarNet detections are shown in Figure 5. Average Precision (AP) under two Intersection-over-Union (IoU) thresholds 0.3 and 0.5 was used for these evaluations. For StarNet, MetaOpt+GradCAM, and CAN

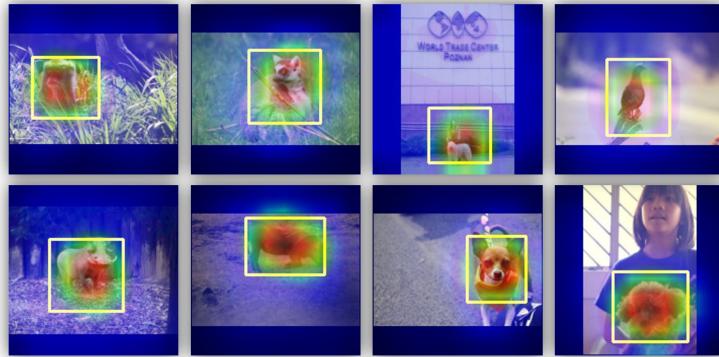


Fig. 5. ImagenetLOC-FS query images from test data, over-imposed with back-projection heatmaps and the corresponding detection bounding boxes.

baselines, the bounding boxes were obtained from the predicted query images heatmaps, using the CAM algorithm from [54] (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 classification through localization of the objects (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 localization (CAN), or intended for classification and not localization (MetaOpt) - which cannot be easily bridged using the standard techniques for localization in classifiers (GradCAM, SelectiveSearch).

As can be seen from Table 2, on ImageNetLOC-FS, for $IoU \geq 0.3$ the StarNet performance is almost on par with 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 $IoU \geq 0.5$. To examine the reason for

Table 2. Average precision (AP, %) of weakly supervised few-shot detection and comparison to baselines on the ImagenetLOC-FS and CUB datasets. GC = GradCAM, SS = SelectiveSearch. RepMet is a fully-supervised upper bound

dataset	method	1-shot					5-shot				
		$IoU \geq 0.3$	$IoU \geq 0.5$	$IoP \geq \frac{2}{3}$	GT coverage		$IoU \geq 0.3$	$IoU \geq 0.5$	$IoP \geq \frac{2}{3}$	GT coverage	
Imagenet LOC-FS	RepMet (upper bound)	59.5	56.9	—	—	70.7	68.8	—	—	—	—
	MetaOpt+GC	32.4	13.8	29.2	22.4	51.9	22.1	41.4	32.8		
	MetaOpt+SS	16.1	4.9	6.7	53.6	27.4	10.2	12.7	59.5		
	PCL [46]	25.4	9.2	23.8	21.2	37.5	11.3	34.3	21.1		
	CAN [12]	23.2	10.3	20.1	32.4	38.2	12.7	35.1	30.8		
	StarNet (ours)	50.0	26.4	43.6	32.5	63.6	34.9	54.8	33.4		
CUB	MetaOpt+GC	53.3	12.0	52.5	18.4	72.8	14.4	62.6	31.1		
	MetaOpt+SS	19.4	6.0	7.8	47.15	26.2	6.4	4.2	48.37		
	PCL [46]	29.1	11.4	29.0	25.6	41.1	14.7	37.0	22.3		
	CAN [12]	60.7	19.3	55.4	26.1	74.8	26.0	66.1	26.5		
	StarNet (ours)	77.1	27.2	71.4	34.9	86.1	32.7	78.7	34.7		

this, we have introduced two new coupled measures: $AP@\{IoP \geq \frac{2}{3}\}$ and ‘average GT coverage’. The *Intersection over Predicted* (IoP) computes the portion of the predicted box covered by the ground truth (GT) box. The IoP threshold of $\frac{2}{3}$ is roughly equivalent to $IoU \geq 0.5$ for same size GT and predicted boxes. The ‘average GT coverage’ is the average of predicted and GT boxes intersections, computed over all the boxes that pass $IoP \geq \frac{2}{3}$. In the Table 2, $AP@\{IoP \geq \frac{2}{3}\}$ column recovers most of the AP percents of the fully-supervised detector. Therefore, the majority of predictions responsible for the performance gap (boxes with $0.3 \leq IoU < 0.5$), are boxes that overlap the GT with about 2/3 or more of their area and cover more than 1/3 of the GT box. This suggests that the observed performance gap is mostly due to detected object parts (usually, most prominent ones), which is common for weakly supervised object detectors.

4.4 Ablation study

We perform an ablation study to verify the contribution of the different components of our method and some of the design choices. We used the 1-shot, 5-way few-shot classification experiment on CUB for this study, results are summarized in Table 4.4. To test the contribution of the localization of novel category instances performed by the StarNet (stage-one), we use the same global average pooling for the prototype features as in StarNet stage-two (section 3.3), only without weighting by $BP_{Q|S}$ and $BP_{S|Q}$ (‘unattended stage-two’ in the table). We then follow up with evaluating the performance of StarNet stage-one (section 3.1) and StarNet stage-two (section 3.3) on their own, this time stage-two 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-one with unstructured evidence pooled from the object regions in stage-two. Finally, we test the effect of iteratively extending the back-projected query region matched to the support using the NMS process (section 3.2) to obtain the final and best performance.

Table 3. Ablation study on the CUB, 1-shot, 5-way experiment

unattended stage-two	72.92
StarNet stage-one	75.86
StarNet stage-two	76.74
full StarNet	78.78
full StarNet with iterative NMS	79.58

5 Conclusions

We have introduced StarNet, a model for few-shot classification which naturally includes object localization in the images. Besides addressing the task of visual classification, especially excelling in situations where images are less cropped around the objects, StarNet is capable of providing plausible explanations for its predictions by highlighting the image regions corresponding to them on both the query and the matched support images. Moreover, our method allows to approach, for the first time (to the best of our knowledge) the task of *weakly-supervised* few-shot object detection, with reasonable accuracy. This challenging

task is of particular interest when training the few-shot detectors for new visual domains, as it alleviates the need to obtain expensive bounding box annotations for a large number of base classes images in the new domain. Future work directions include extending StarNet towards efficient end-to-end differentiable multi-scale processing for better handling very small and very large objects; iterative refinement utilizing StarNet’s locations predictions made during training; and applying StarNet for other applications requiring accurate localization using only a few examples, such as visual tracking.

References

1. Alfassy, A., Karlinsky, L., Aides, A., Shtok, J., Harary, S., Feris, R., Giryes, R., Bronstein, A.M.: LaSO: Label-Set Operations networks for multi-label few-shot learning. In: CVPR (2019)
2. Bertinetto, L., Torr, P.H.S., Vedaldi, A.: Meta-learning with differentiable closed-form solvers. ICLR pp. 1–15 (2019)
3. Bilen, H., Vedaldi, A.: Weakly Supervised Deep Detection Networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 2846–2854 (2016). <https://doi.org/10.1109/CVPR.2016.311>
4. Bishop, C.M.: Pattern Recognition and Machine Learning. Information Science and Statistics (2006), <https://dl.acm.org/citation.cfm?id=1162264>
5. Chen, H., Wang, Y., Wang, G., Qiao, Y.: LSTD: A Low-Shot Transfer Detector for Object Detection. AAAI (2018), www.aaai.org
6. Chen, W.Y., Liu, Y.C., Kira, Z., Wang, Y.C., Huang, J.B.: A Closer Look At Few-Shot Classification. In: ICLR (2019)
7. Dvornik, N., Schmid, C., Mairal, J.: Diversity with Cooperation: Ensemble Methods for Few-Shot Classification. The IEEE International Conference on Computer Vision (ICCV) (2019), <http://arxiv.org/abs/1903.11341>
8. Finn, C., Abbeel, P., Levine, S.: Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400 (2017), <http://arxiv.org/abs/1703.03400>
9. Girshick, R., Iandola, F., Darrell, T., Malik, J.: Deformable part models are convolutional neural networks. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition pp. 437–446 (2015). <https://doi.org/10.1109/CVPR.2015.7298641>
10. Hao, F., He, F., Cheng, J., Wang, L., Cao, J., Tao, D.: Collect and Select : Semantic Alignment Metric Learning for Few-Shot Learning. IEEE International Conference on Computer Vision (ICCV) pp. 8460–8469 (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. arXiv:1512.03385 (2015), <https://arxiv.org/pdf/1512.03385.pdf>
12. Hou, R., Chang, H., Ma, B., Shan, S., Chen, X.: Cross Attention Network for Few-shot Classification. NeurIPS (10 2019), <http://arxiv.org/abs/1910.07677>
13. Hu, T., Mettes, P., Huang, J.H., Snoek, C.G.M.: SILCO : Show a Few Images , Localize the Common Object. IEEE International Conference on Computer Vision (ICCV) (2019), <http://taohu.me/SILCO/>.
14. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot Object Detection via Feature Reweighting. In: The IEEE International Conference on Computer Vision (ICCV) (2019), <http://arxiv.org/abs/1812.01866>
15. Karlinsky, L., Shtok, J., Harary, S., Schwartz, E., Aides, A., Feris, R., Giryes, R., Bronstein, A.M.: RepMet: Representative-based metric learning for classification and one-shot object detection. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5197–5206 (2019), <http://arxiv.org/abs/1806.04728>
16. Karlinsky, L., Shtok, J., Tzur, Y., Tzadok, A.: Fine-grained recognition of thousands of object categories with single-example training. Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 pp. 965–974 (2017). <https://doi.org/10.1109/CVPR.2017.109>
17. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Technical report. Science Department, University of Toronto, Tech. pp. 1–60 (2009). <https://doi.org/10.1.1.222.9220>

18. Lee, K., Maji, S., Ravichandran, A., Soatto, S., Services, W., San Diego, U.C., Amherst, U.: Meta-Learning with Differentiable Convex Optimization. In: CVPR (2019), <https://github.com/kjunelee/MetaOptNet>
19. Leibe, B., Leonardis, A., Schiele, B.: An Implicit Shape Model for Combined Object Categorization and Segmentation (May), 508–524 (2006). https://doi.org/10.1007/11957959_26
20. Li, H., Eigen, D., Dodge, S., Zeiler, M., Wang, X.: Finding Task-Relevant Features for Few-Shot Learning by Category Traversal 1 (2019), <http://arxiv.org/abs/1905.11116>
21. Li, Z., Zhou, F., Chen, F., Li, H.: Meta-SGD: Learning to Learn Quickly for Few-Shot Learning. arXiv:1707.09835 (2017), <http://arxiv.org/abs/1707.09835>
22. Lifchitz, Y., Avrithis, Y., Picard, S., Bursuc, A.: Dense Classification and Implanting for Few-Shot Learning. In: CVPR (2019)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Lecture Notes in Computer Science. vol. 8693 LNCS, pp. 740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48
24. Lin, T.Y., Roychowdhury, A., Maji, S.: Bilinear CNNs for Fine-grained Visual Recognition. TPAMI (2017), <http://vis-www.cs.umass.edu/bcnn>.
25. Liu, L., Muelly, M., Deng, J., Pfister, T., Li, J.: Generative Modeling for Small-Data Object Detection. In: The IEEE International Conference on Computer Vision (ICCV) (2019)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. Lecture Notes in Computer Science **9905 LNCS**, 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
27. Maji, S., Malik, J.: Object Detection using a Max-Margin Hough Transform. In: CVPR (2009)
28. Mishra, N., Rohaninejad, M., Chen, X., Abbeel, P.: A Simple Neural Attentive Meta-Learner. Advances In Neural Information Processing Systems (NIPS) (2017), <https://arxiv.org/pdf/1707.03141.pdf>
29. Munkhdalai, T., Yu, H.: Meta Networks. arXiv:1703.00837 (2017). <https://doi.org/10.1093/mnrasl/slx008>, <http://arxiv.org/abs/1703.00837>
30. Oreshkin, B.N., Rodriguez, P., Lacoste, A.: TADAM: Task dependent adaptive metric for improved few-shot learning. NeurIPS (5 2018), <http://arxiv.org/abs/1805.10123>
31. Pan, T., Wang, B., Ding, G., Han, J., Yong, J.: Low Shot Box Correction for Weakly Supervised Object Detection pp. 890–896 (2019). <https://doi.org/10.24963/ijcai.2019/125>
32. Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., Devito, Z.: Automatic differentiation in PyTorch. 31st Conference on Neural Information Processing Systems (Nips), 1–4 (2017). <https://doi.org/10.1017/CBO9781107707221.009>
33. Pedro F. Felzenszwalb, Ross B. Girshick, D.M., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, **32**(9), 16271645 (2010). <https://doi.org/10.1109/TPAMI.2010.50>
34. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep Hough Voting for 3D Object Detection in Point Clouds (2019), <http://arxiv.org/abs/1904.09664>
35. Qiao, L., Shi, Y., Li, J., Wang, Y., Huang, T., Tian, Y.: Transductive Episodic-Wise Adaptive Metric for Few-Shot Learning (2019), <http://arxiv.org/abs/1910.02224>

36. Ravi, S., Larochelle, H.: Optimization As a Model for Few-Shot Learning. ICLR pp. 1–11 (2017), <https://openreview.net/pdf?id=rJY0-Kc11>
37. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. IJCV (9 2015), <http://arxiv.org/abs/1409.0575>
38. Rusu, A.A., Rao, D., Sygnowski, J., Vinyals, O., Pascanu, R., Osindero, S., Hadsell, R.: Meta-Learning with Latent Embedding Optimization. In: ICLR (7 2018), <http://arxiv.org/abs/1807.05960>
39. Sali, E., Ullman, S.: Combining Class-Specific Fragments for Object Classification (1994), 1–21 (2013). <https://doi.org/10.5244/c.13.21>
40. Schwartz, E., Karlinsky, L., Shtok, J., Harary, S., Marder, M., Kumar, A., Feris, R., Giryes, R., Bronstein, A.M.: Delta-Encoder: an Effective Sample Synthesis Method for Few-Shot Object Recognition. Neural Information Processing Systems (NIPS) (2018), <https://arxiv.org/pdf/1806.04734.pdf>
41. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Proceedings of the IEEE International Conference on Computer Vision pp. 618–626 (2017). <https://doi.org/10.1109/ICCV.2017.74>
42. Shaban, A., Rahimi, A., Bansal, S., Gould, S., Boots, B., Hartley, R.: Learning to Find Common Objects Across Few Image Collections pp. 5117–5126 (2019), <http://arxiv.org/abs/1904.12936>
43. Shaoqing Ren, Kaiming He, R.G.J.S.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Neural Information Processing Systems (NIPS) (2015). <https://doi.org/10.1109/TPAMI.2016.2577031>
44. Snell, J., Swersky, K., Zemel, R.: Prototypical Networks for Few-shot Learning. In: NIPS (2017), <http://arxiv.org/abs/1703.05175>
45. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H.S., Hospedales, T.M.: Learning to Compare: Relation Network for Few-Shot Learning. arXiv:1711.06025 (11 2017), <http://arxiv.org/abs/1711.06025>
46. Tang, P., Wang, X., Bai, S., Shen, W., Bai, X., Member, S., Liu, W., Yuille, A.: PCL: Proposal Cluster Learning for Weakly Supervised Object Detection. Tech. rep.
47. Tang, P., Wang, X., Wang, A., Yan, Y., Liu, W., Huang, J., Yuille, A.: Weakly Supervised Region Proposal Network and Object Detection. In: European Conference on Computer Vision (ECCV) (2018). https://doi.org/10.1007/978-3-030-01252-6_22
48. Uijlings, J.R.R., Van De Sande, K.E.A., Gevers, T., Smeulders, A.W.M.: Selective Search for Object Recognition. Tech. rep., <http://disi.unitn.it/>
49. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching Networks for One Shot Learning. NIPS (2016). <https://doi.org/10.1109/CVPR.2016.95>, <http://arxiv.org/abs/1606.04080>
50. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset pp. 1–15 (2011). <https://doi.org/CNS-TR-2010-001.2010.>, <http://authors.library.caltech.edu/27468/>
51. Wang, Y.X., Ramanan, D., Hebert, M.: Meta-Learning to Detect Rare Objects. In: The IEEE International Conference on Computer Vision (ICCV). pp. 9925–9934 (2019)
52. Wertheimer, D., Hariharan, B.: Few-Shot Learning with Localization in Realistic Settings (2019), <http://arxiv.org/abs/1904.08502>

53. Zhang, J., Zhao, C., Ni, B., Xu, M., Yang, X.: Variational Few-Shot Learning. In: IEEE International Conference on Computer Vision (ICCV) (2019)
54. Zhang, X., Wei, Y., Kang, G., Yang, Y., Huang, T.: Self-produced guidance for weakly-supervised object localization. Lecture Notes in Computer Science **11216 LNCS**, 610–625 (2018). https://doi.org/10.1007/978-3-030-01258-8_37
55. Zhou, F., Wu, B., Li, Z.: Deep Meta-Learning: Learning to Learn in the Concept Space. Tech. rep. (2 2018), <http://arxiv.org/abs/1802.03596>