Few-shot Weakly-Supervised Object Detection via Directional Statistics

Amirreza Shaban*,[1] Amir Rahimi*,[2] Thalaiyasingam Ajanthan[2] Byron Boots[1] Richard Hartley[2]

[1]University of Washington [2]ANU & ACRV

Abstract

Detecting novel objects from few examples has become an emerging topic in computer vision recently. However, current methods need fully annotated training images to learn new object categories which limits their applicability in real world scenarios such as field robotics. In this work, we propose a probabilistic multiple-instance learning approach for few-shot Common Object Localization (COL) and few-shot Weakly Supervised Object Detection (WSOD). In these tasks, only image-level labels, which are much cheaper to acquire, are available. We find that operating on features extracted from the last layer of a pre-trained Faster-RCNN is more effective compared to previous episodic learning based few-shot COL methods. Our model simultaneously learns the distribution of the novel objects and localizes them via expectation-maximization steps. As a probabilistic model, we employ von Mises-Fisher (vMF) distribution which captures the semantic information better than Gaussian distribution when applied to the pre-trained embedding space. When the novel objects are localized, we utilize them to learn a linear appearance model to detect novel classes in new images. Our extensive experiments show that the proposed method, despite being simple, outperforms strong baselines in few-shot COL and WSOD, as well as large-scale WSOD tasks.

1. Introduction

In this paper we address the problem of N-way, K-shot Weakly Supervised Object Detection (WSOD), and develop a method with the following capabilities.

Suppose that we are given a set of $N \times K$ previously unseen images consisting of K images of objects from each of N previously unknown (novel) classes. These will be called the "support images." Each training image has image-level labels, indicating which classes are present in the image. Typically, the number of novel classes N may be up to 20 and the number of training images K from each class may be 5 or 10, but there is no requirement that the number of images in each novel class are equal.

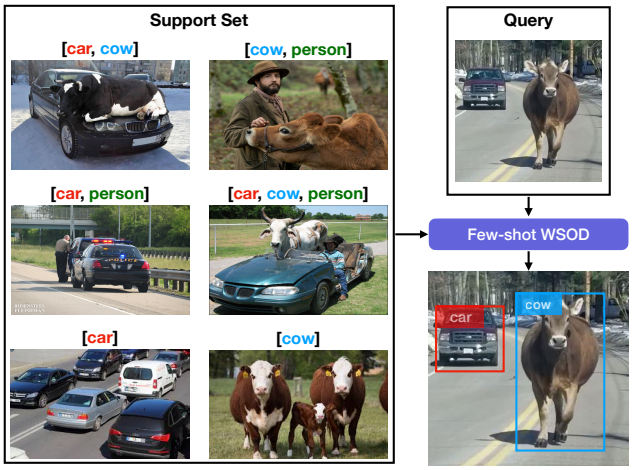

Figure 1. *Few-shot WSOD problem. Similar to the few-shot classification problem, the input training set (support set) only contains image labels (*`car`*,* `cow` *and* `person` *are novel classes in this example). The model learns to detect the target objects in the test (query) image. Few-shot WSOD bridges few-shot classification and object detection by learning to detect the novel objects in the query images while only needs image-level labels for the support images.*

Given this small number of support images, the algorithm learns to find instances of (possibly multiple) objects from any of the novel classes in a query image, and will put a bounding box around all such positive instances. As summarized in Fig. 1, our system provides a flexible object detection algorithm that requires a very small training set of images of novel objects, where each image is annotated only with image-level labels. As such, it is suitable for classifying and detecting objects given only the images provided, for instance, by an internet image search for images of novel classes. In comparison to supervised few-shot object detection approaches, e.g., [40, 39, 38, 23], where manually labeled bounding box annotations are required, this is a more realistic setting to learn an object detector on novel examples with applications like robotics [15] or video

*Equal Contribution. Contact at `ashaban@uw.edu` or `amir.rahimi@anu.edu.au`

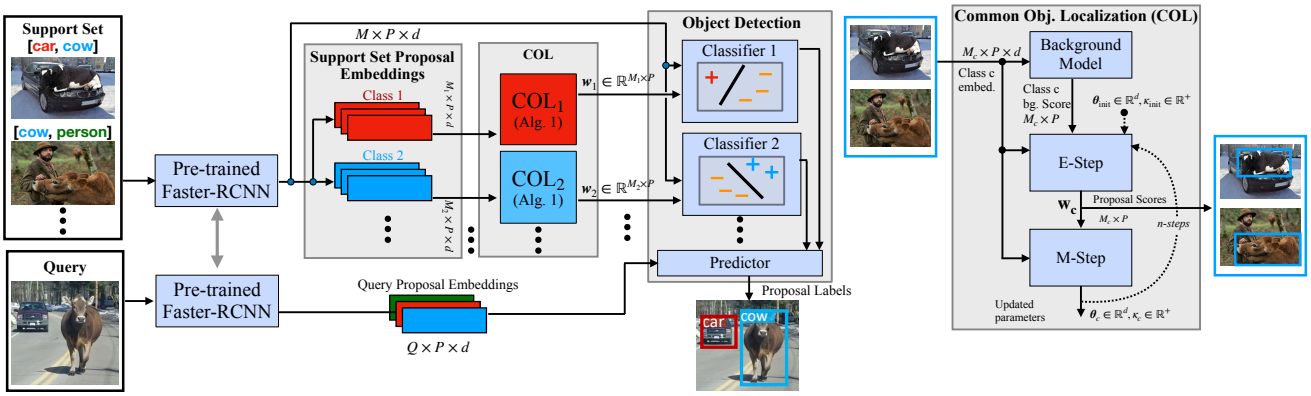

Figure 2. *The feature maps are shown as the shape of their tensors. Q, M, and C denote the number of queries, support images, and classes respectively. A pre-trained Faster-RCNN (shown in Fig. 3) on the base dataset is used to extract P proposals from each input image. The embeddings are grouped based on their corresponding image-level labels and each group is fed into a separate Common-Object Localization (COL) module.* **COL** *module (shown in detail on the right) receives proposal embeddings of images of a class (M_c is the number of images within class c) and simultaneously estimates the common class direction $\boldsymbol{\theta}_c$ and concentration κ_c along with bounding-box level labels $\mathbf{w}_c$ via EM steps.* **The Object Detection** *module uses the top labels of $\mathbf{w}_c$ to learn an appearance model for each novel class in the support set. This appearance model is then tested on the testing proposals to detect novel objects in the query set.*

object segmentation [21].

We first use a Faster-RCNN network to produce bounding box proposals of the possible regions containing an object with their associated feature vectors. This network is pre-trained on a fully annotated base dataset, with bounding boxes of objects of various classes; the base dataset does not contain any of the novel training classes.

Our approach has two main modules: 1) A *common object localization* (COL) module is used first to localize the novel objects in the support images. 2) An object detection module to learn novel object appearances from the annotated support images. We explain each of these modules in more details below.

COL module. To localize the novel objects in the support images, COL finds the common object in the K images provided for each of the N novel classes. The input to the common object detector is the set of normalized feature vectors from the images of a novel class c. This set of normalized feature vectors corresponds to the bounding boxes provided by the proposal network. An EM algorithm on these feature vectors determines the mean direction and concentration of a probability distribution on the sphere (von Mises-Fisher (vMF) distribution) that is most likely to favour a common object representative from each image. The closest feature from each image identifies the bounding box containing the common object. A distribution for a background class is also trained, using the base dataset to steer the COL away from selecting background objects. This common object detector is run separately on the images from each of the N novel classes.

Detection module. Once the novel objects are found by the COL module, these annotations are used to train a box classifier for each novel class c. The classification is done by a 2-class (contains / does not contain the object) classification algorithm, once again working on normalized feature vectors. These bounding boxes (and their associated features) are labeled as either positive or negative for containing the object of the novel class. The positive bounding boxes are those that are determined by the COL module to contain the common object from class c; the negative samples are chosen from proposals selected from the images of the other classes. Thus, the classifier for class c is trained to distinguish features corresponding to bounding boxes containing an object of class c from those that do not.

Finally, at test time, a query image is passed through the proposal network to provide bounding boxes (and their features). These bounding boxes are then evaluated by each of the classifiers to determine whether they contain a novel class object or belong to the background.

The proposed method is summarized in Fig. 2. We make several contributions and important observations: 1) We propose a simple yet powerful COL that uses directional statistics for modeling. Our COL module can be built on top of off-the-shelf pre-trained Faster-RCNN models without extra parameters. We observe that by using feature vector directions in our probabilistic model, we can better capture the semantic information compared to a Gaussian probabilistic model. To our knowledge, employing directional statistics for multiple-instance learning is new. 2) We employ a detection module to extend COL to few-shot WSOD. To the best of our knowledge few-shot WSOD has not been studied in the literature before. 3) Despite its simplicity, our method outperforms sophisticated few-shot COL algorithms [28, 12] on PASCAL VOC [6], MS COCO [19], and ILSVRC detection [3] benchmarks. In WSOD, our method outperforms recent knowledge-transfer based approaches [34, 11] in both few-shot and large-scale settings.

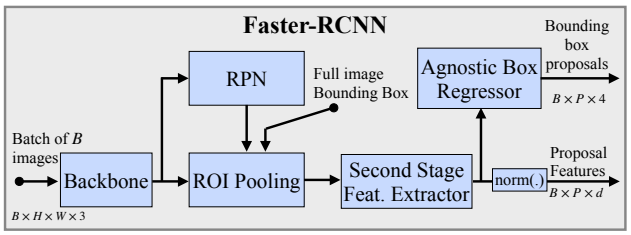

Figure 3. *Feature Extraction. We use a pre-trained Faster-RCNN on the base dataset to extract P proposals from each input image. A ℓ_2 normalization layer is employed to project all the features onto the unit hypersphere.*

2. Details of Methodology

2.1. Few-shot WSOD and COL Tasks Definition

The goal of our few-shot WSOD is to learn a model that predicts bounding boxes of query set images $\mathcal{D}_{\text{test}}$ when given a set of support images $\mathcal{D}_{\text{train}}$. Let $\mathcal{L}$ be the set of classes in the support set. The support set consists of image-label pairs $(\mathbf{I}, \mathbf{y}) \in \mathcal{D}_{\text{train}}$ where image-level label $\mathbf{y} \subseteq \mathcal{L}$ is a subset of classes present in the image $\mathbf{I}$[1]. The support set is typically a small K-shot, N-way set sampled from a large dataset $\mathcal{D}_{\text{novel}}$ with a variety of novel classes $\mathcal{C}_{\text{novel}}$. The sampling process for few-shot WSOD follows rules that are similar to few-shot classification problems [16, 32]. A set of N classes $\mathcal{L} \subset \mathcal{C}_{\text{novel}}$, called target classes, are first sampled. Then, for each target class $c \in \mathcal{L}$, K images containing at least an instance of class c are sampled *without replacement* to create the support set $\mathcal{D}_{\text{train}}$. The query set $\mathcal{D}_{\text{test}}$ is sampled similarly, but unlike the support set, query labels also contain bounding box annotations in addition to the image-level labels, as the goal is to detect target objects in the query data. These bounding box annotations are only used for evaluation. Few-shot COL [28, 12] is a special case of few-shot WSOD where there is only one target class in the support set, i.e., $N = 1$.

For pre-training, the algorithm has access to a large dataset $\mathcal{D}_{\text{base}}$ with a set of base classes $\mathcal{C}_{\text{base}}$. Typically, there is no image in common between the base and novel datasets. Moreover, the set of base classes is disjoint from the set of novel classes used in evaluation, i.e., $\mathcal{C}_{\text{base}} \cap \mathcal{C}_{\text{novel}} = \varnothing$.

2.2. Pre-training and Feature Extraction

We pre-train a Faster-RCNN [26] on the base dataset for bounding box and feature extraction. The overall architecture is shown in Fig. 3. To train the network, we use the original bounding box labels within the base dataset to define the Region Proposal Network (RPN) and second-stage losses of the Faster-RCNN. We adapt a class-agnostic bounding box regression model in the second-stage to get one bounding box per feature proposal regardless of the number of base classes. Once trained, we use the trained Faster-RCNN to extract P bounding box proposals $B \in \mathbb{R}^{P\times 4}$ and their corresponding d-dimensional features $F \in \mathbb{R}^{P\times d}$ from each input image $\mathbf{I}$. We also apply an ℓ_2 normalization layer to project all the features to the unit hypersphere. As discussed later, the normalization step is important as our model uses the cosine similarity measure for better generalization.

We need the feature extracted from the full image bounding box to initialize our COL method. This is accomplished by manually appending the full image bounding box to the box proposals of the RPN, thus its feature is extracted by the Faster-RCNN second-stage feature extractor. We denote the first proposal in B and F, the complete image bounding box and its feature, respectively.

2.3. Statistical Model Assumptions

Since the support set $\mathcal{D}_{\text{train}}$ provided to the learner is limited, it is crucial to employ proper learning biases in the model to combat overfitting. Inspired by the success of prototypical networks [32], we design our model based on the assumption that features of each object class form a single cluster in the embedding space. We propose to use directional data based on the von Mises-Fisher (vMF) distribution, which arises naturally when each cluster is distributed on the unit hypersphere. Formally, we assume features of each foreground class follow von Mises-Fisher distribution with mean direction $\boldsymbol{\theta}$ and positive concentration parameter κ

$$p^{+}_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{1}{Z} \exp\left(\kappa \boldsymbol{\theta}^{\top} \mathbf{x}\right) \text{ s.t. } \|\boldsymbol{\theta}\| = 1 \,, \tag{1}$$

where Z is the normalizing constant and input $\mathbf{x} \in \mathbb{R}^d$ is a unit vector, i.e., $\|\mathbf{x}\| = 1$ or equivalently $\mathbf{x} \in \mathbb{S}^{d-1}$. In Section 2.4, we propose an expectation maximization algorithm to estimate the mean direction and concentration parameter of a novel class from the support set.

We could also use Gaussian distribution for our model which has an analogous effect to using Euclidean distance. We empirically show that vMF provides superior results to Gaussian distribution when using pre-trained features. Our results support related works in supervised few-shot learning [24, 8] where using the cosine similarity outperforms the Euclidean distance measure. The underlying reason for this is well-studied by Wang *et al*. [37]; Softmax loss used in the pre-training tends to create a 'radial' feature distribution where direction specifies the semantic classes while magnitude decides the classification confidence.

Additionally, a background class distribution is learned to steer the learner toward objects and away from back-

[1] In contrast to few-shot image classification, few-shot WSOD images can have multiple labels.

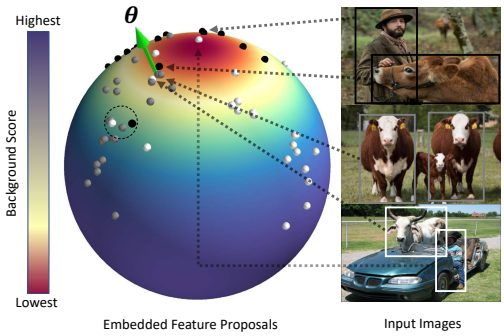

Figure 4. *Example of COL across three images. Data points on the unit sphere represent feature proposals extracted from all input images. Features extracted from each image are colored the same (shown in white, gray, black colors). Background score function $u_{\omega}^{-}(\mathbf{x})$ is also shown on the unit sphere where blue and red indicate the highest and lowest background scores, respectively. The COL unit goal is to find a common object representation $\boldsymbol{\theta}$ (shown by green arrow) which is close to at least a white, gray, and black data point. Note that the area marked with dashed circle is also close to proposals from all three images but direction $\boldsymbol{\theta}$ is favored as it has a lower background score.*

ground proposals. Let

$$p_{\omega}^{-}(\mathbf{x}) = \frac{1}{U} u_{\omega}^{-}(\mathbf{x}) \,, \tag{2}$$

represent the background class distribution where U is a constant normalizer. As the base dataset provides a reach set of examples for learning the background model, the background distribution is learned from the base dataset and remains fixed when evaluating on WSOD examples sampled from the novel data.

To learn the background distribution, we collect a set of background proposals with low intersection-over-union (IoU) score (< 0.3) to the objects within the base dataset and use maximum likelihood estimation in [2] to find the parameters of vMF distribution for the background data.

2.4. Few-shot COL

We first explain the method for few-shot COL with a single novel common object within the support set and employ it for few-shot WSOD later. As shown in Fig. 4, COL module's goal is to find the common object representation across a set of images with one novel object in common. Let $\mathcal{F} = \{F_i\}_{i=1}^{M}$ denote the Faster-RCNN feature proposals extracted from the input images where M is number of images. Each proposal has a (latent) binary label that indicates whether the proposal tightly encloses the common object. Namely, $\mathbf{z}_{ij} \in \{0, 1\}$ is the label of the j-th proposal in the i-th image. Starting from an initial guess for the direction $\boldsymbol{\theta}$ and concentration κ parameters of the novel common class, the algorithm alternately refines the statistical model parameters and label estimations in an expectation-maximization optimization framework. We present the update rules here and defer the derivations that bring interesting insights into the proposed method to **??**.

In the E-step, the algorithm uses the current model parameters to estimate soft labels $\mathbf{w}$, where $\mathbf{w}_{ik} \in [0, 1]$ is the soft label for the k-th proposal within the i-image, via attention over the proposals within the image

$$\mathbf{w}_{ik} = \frac{p_{\boldsymbol{\theta}}^{+}(F_{ik}) / p_{\boldsymbol{\omega}}^{-}(F_{ik})}{\sum_{j=1}^{P} p_{\boldsymbol{\theta}}^{+}(F_{ij}) / p_{\boldsymbol{\omega}}^{-}(F_{ij})} \tag{3}$$

where $F_{ij} \in \mathbb{S}^{d-1}$ is the feature of the j-th proposal in F_i. Recall that $u_{\boldsymbol{\omega}}^{-}$ is our trained background scoring function introduced in Eq. (2). In this step, the proposal with a high cosine similarity to the current direction $\boldsymbol{\theta}$ and a low background score gets the highest label value within each image. For the vFM distribution, E-step is done using a linear layer and a softmax (lines 3-5 in Algorithm 1).

In the M-step, the direction $\boldsymbol{\theta}$ and concentration κ are updated given the new labels

$$\begin{aligned} \boldsymbol{\theta} &\leftarrow \frac{\mathbf{r}}{\|\mathbf{r}\|} \,, \kappa \leftarrow d\|\mathbf{r}\| \\ \text{where } \mathbf{r} &= \frac{1}{M} \sum_{i=1}^{M} \mathbf{w}_i^{\top} F_i = \frac{1}{M} \sum_{i=1}^{M} \sum_{k=1}^{P} \mathbf{w}_{ik} F_{ik} \,, \end{aligned} \tag{4}$$

where d is the feature dimension. Note that as we only need to know the multiplication of κ and $\boldsymbol{\theta}$, the update rule simplifies to $\kappa\boldsymbol{\theta} \leftarrow d\mathbf{r}$. Intuitively, one can see $\tilde{\mathbf{x}}_i = \mathbf{w}_i^{\top} F_i$ as the common object representation within the i-th image; $\tilde{\mathbf{x}}_i$ is estimated by computing the weighted average over all the proposals where the contribution of proposals are controlled by their soft labels. Given $\tilde{\mathbf{x}}_i$, the novel class direction $\boldsymbol{\theta}$ is estimated as the mean of the common object representations, similar to the prototypical networks [32], and the estimated mean is projected back onto the unit hypersphere. Updating κ is more involved and is numerically difficult where d and κ are large. We propose several approximations in **??** and show that the approximation in Eq. (4) achieves the best performance in practice.

Algorithm 1 summarizes our COL method. The problem is solved in an iterative fashion by alternating between E-step in Eq. (3) and M-step in Eq. (4) until convergence. Following the common practice in WSOD [25, 34, 22], we use the bounding box feature extracted from the complete support images to initialize our model. Recall that we use the first proposal in F_i to represent the complete image feature. Thus, the initialization step can be written as

$$(\kappa\boldsymbol{\theta})_{\text{init}} \leftarrow \frac{d}{M} \sum_{i=1}^{M} F_{i1}. \tag{5}$$

We remark that our initial direction is similar to what is used

Algorithm 1: Common Object Localization

Input: $\mathcal{F} = \{F_1, \ldots, F_M\}$, $u^-_{\boldsymbol{\omega}}$
Output: Common class mean direction $\boldsymbol{\theta}$ and concentration κ

$\kappa\boldsymbol{\theta} \leftarrow \frac{d}{M}\sum_{i=1}^{M} F_{i1}$ // Initialization
for $t \leftarrow 1$ ***to*** T **do** // Iterations
  for $i \leftarrow 1$ ***to*** M **do** // E-step
    $\mathbf{o}_{ij} \leftarrow \kappa\boldsymbol{\theta}^\top F_{ij} - \log u^-_{\boldsymbol{\omega}}(F_{ij}) \quad \forall j \in [1, P]$
    $\mathbf{w}_i \leftarrow \text{softmax}(\mathbf{o}_i)$ // Update Soft labels
  $\mathbf{r} \leftarrow \frac{1}{M}\sum_{i=1}^{M} \mathbf{w}_i^\top F_i$
  $\kappa\boldsymbol{\theta} \leftarrow d\mathbf{r}$ // M-step

as class mean in prototypical networks [32]. What makes us different is EM steps that refine the estimated mean by focusing on the common objects and discarding background parts of the image.

Finding the Common Object in the Query Set For a single feature proposal $\mathbf{x} \in \mathbb{S}^{d-1}$ extracted from query image $\mathbf{I}$, our goal is to estimate the class label $c \in \{0, 1\}$ which indicates if the query proposal tightly encloses the target object. Given the estimated common object mean $\boldsymbol{\theta}$, concentration κ, and the background class distribution $p^-_{\boldsymbol{\omega}}$, we compute conditional class distribution function $P(c|\mathbf{x}) \propto P(c)p(\mathbf{x}|c)$ where $P(c)$ and $p(\mathbf{x}|c)$ are the class prior and likelihood, respectively. Assuming $P(c = 1) = \alpha$, and using the background and vMF foreground class likelihoods, the conditional class distribution is written as

$$P(c|\mathbf{x}) \propto \begin{cases} \exp\left(\kappa\boldsymbol{\theta}^\top\mathbf{x} - \log u^-_{\boldsymbol{\omega}}(\mathbf{x})\right) & c = 1 \\ \lambda & c = 0\,, \end{cases} \tag{6}$$

where $\lambda = (1-\alpha)/\alpha \times Z/U$ encapsulates all the constants (Z is vMF normalizer.) Equivalently, proposal $\mathbf{x}$ can be classified via a softmax over the logits

$$\text{logit}(c|\mathbf{x}) = \begin{cases} \kappa\boldsymbol{\theta}^\top\mathbf{x} - \log u^-_{\boldsymbol{\omega}}(\mathbf{x}) & c = 1 \\ \log\lambda & c = 0\,. \end{cases} \tag{7}$$

We set $\lambda = 1$ for all the COL experiments. Changing λ adjusts the confidence values but keeps the order of the final scores the same, therefore, its value does not affect the mean Average Precision (mAP) or Correct Localization (CorLoc) metrics.

2.5. Few-shot WSOD

For the task of WSOD where we have more than one target class, our COL algorithm is first used to label instances of each class. Once the support set is labeled, an off-the-shelf few-shot object detection model can be used for learning novel classes. Inspired by the success of the recent few-shot object detection method in [38], we employ a single layer cosine similarity classifier for learning.

Learning is performed on one target class $c \in \mathcal{L}$ at a time. Let $\mathbf{v}_c \in \mathbb{R}^d$ denote the classifier weight for class c. The classification score for this class is computed as

$$s_c(\mathbf{x}) = \frac{\tau\mathbf{v}_c^\top\mathbf{x}}{\|\mathbf{v}_c\|}\,, \tag{8}$$

where $\mathbf{x} \in \mathbb{S}^{d-1}$ is the ℓ_2 normalized feature proposal extracted by our Faster-RCNN model and τ is temperature hyperparameter. For class c, the input training set $\mathcal{D}_{\text{train}}$ is split into positive images $\mathcal{D}^c_{\text{train}}$ of images that have the target class and negative set $\mathcal{D}_{\text{train}} \setminus \mathcal{D}^c_{\text{train}}$, images without the target class. Then, we label $\mathcal{D}^c_{\text{train}}$ by running the COL algorithm on the positive images and select the proposal with the highest soft label from each image as the common object representative. All the proposals in the negative set are used as negative examples. Finally, $\mathbf{v}_c$ is learned by minimizing the sigmoid cross entropy loss over the positive and negative proposals. We use the L-BFGS optimizer with strong Wolfe line search for faster convergence.

At the test time, test proposal $\mathbf{x}$ from the query set $\mathcal{D}_{\text{test}}$ is scored using the classifiers learned for each novel class.

3. Related Work

Multiple-instance learning methods such as MI-SVM [1] have been extensively used for large-scale weakly supervised object detection. In a standard multiple-instance learning framework, latent bounding box labels and the appearance model are estimated jointly in an alternating optimization process with the constraint that at least one bounding box should be positive in each image. Alternating optimization combined with the modern deep neural network architectures is a dominant technique in the literature showing the state-of-the-art performance in WSOD [25, 34, 9, 31]. Ilse *et al*. [13] propose an attention-based deep multiple-instance learning architecture where bag label probability distribution is learned by neural networks. More related to our work are a class of WSOD algorithms that use knowledge transfer from a fully annotated base dataset to aid WSOD for a set of novel classes [25, 34, 11, 4]. In [34], a class-agnostic objectness score is learned from the base dataset and is utilized to guide the multiple-instance learning optimization by steering toward objects and away from the background. These methods rely on a relatively large dataset to learn novel categories.

Co-localization [18], co-segmentation [17, 35], and co-saliency [43] methods have the same kind of output as weakly-supervised object localization but they typically do not utilize negative examples. More recently, several methods were developed for localizing the common novel object under the few-shot setting [28, 12, 29]. Shaban *et al*. [28] learn a pairwise potential function between proposals of the base classes and use this pairwise metric to solve

a minimum-energy labeling problem over a bidirectional graphical model to co-localize novel classes. SILCO [12] finds the common object by computing a dense similarity map between each support image and the query while only exploring the similarity among support images using their coarse image-level features via a global average pooling. Although using global average pooling reduces the computation, ignoring the dense similarities among support images negatively affects the common object localization.

Few-shot learning has gained a lot of attention in image classification [5, 27, 20, 7, 36]. Prototypical networks [32] use the mean of embedded support examples to represent novel class prototypes and classify query examples by comparing their distances to the class prototypes. More recently, Qi *et al*. [24] propose a weight imprinting process to learn the prototypes on the unit hypersphere. Learning on the unit hypersphere has been employed by other few-shot learning algorithms for better generalization [8] and to stabilize the training [38]. Most recently, Yang *et al*. [42] propose to make the distributions more Gaussian by transforming the features of the support set and query set using Tukey's Ladder of Powers transformation [33]. It is shown that Tukey's normalization significantly improves the performance of few-shot prototypical learning. As the scope of these methods is limited to supervised learning, we compare different normalizations and transformations used in the literature for the weakly-supervised task of COL in Section 4.4.

4. Experiments

We evaluate the proposed method in few-shot COL and WSOD problems. We compare our work (vMF-MIL) with Greedy Tree [28] and SILCO [12], two state-of-the-art methods for the task of few-shot common object localization.

To the best of our knowledge there is no WSOD algorithm for few-shot setting in the literature. However, WSOD with knowledge-transfer methods [25, 34, 11, 4] are closely related to our work. We describe a slightly modified version of [34], called MI-SVM in our experiments, in **??**, and discuss its differences to the proposed method. The MI-SVM baseline is not applicable to the COL problem as it always requires negative examples for training. To compare MI-SVM against other COL methods, we provide MI-SVM with an extra set of K negative images that do not have the target class when sampling the support set.

The original version of Greedy Tree selects only one proposal from each image in the support set and does not perform detection on a new query image. To make it compatible with other methods, we add a simple inference step to Greedy Tree. Let $\mathcal{O} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ denote the set of selected proposals, one from each image in the support set. We score feature proposal $\mathbf{x}$ from the query image as a sum of its pairwise similarities to all the selected proposals, i.e., $\text{score}(\mathbf{x}) = \sum_{j=1}^{M} r(\mathbf{x}, \mathbf{x}_j)$, where r is the learned pairwise similarity function by Greedy Tree. The computed score measures the negative change in the energy value if $\mathbf{x}$ were added as a new node to the graph labeling problem used in [28].

In all the methods, we first hold out 20 base classes for validation and hyperparameter tuning and then re-train on all the base classes with the best found parameters. For evaluation, we compute the correct localization (CorLoc) rate [4] and mean Average Precision (mAP) with IoU overlap threshold of 0.5 on the query image.

4.1. Common Object Localization

We use the official implementations of SILCO and Greedy Tree for this experiment. To have a fair comparison with SILCO, we employ Faster-RCNN with a VGG16 [30] backbone architecture for feature extraction in both Greedy Tree and our method.

We evaluate on a popular MS COCO 2014 [19] split used in few-shot object detection methods [38, 23, 40, 41, 14], named COCO60. In the COCO60 split, 60 categories disjoint with the PASCAL VOC dataset are used as base classes and the remaining 20 classes are used as novel classes. This allows us to also perform a cross-dataset evaluation on the PASCAL VOC07 [6] test set. We evaluate the performance of each method over 2000 randomly sampled tasks.

Table 1 and Table 2 summarize the results on PASCAL VOC and MS COCO datasets, respectively. Despite its simplicity, our method outperforms all the methods by a large margin, followed by Greedy Tree and SILCO. Specifically, we gain between 10% to 20% relative improvement in mAP metric against the second best performing method. The proposed method and Greedy Tree both estimate latent proposal-level labels of the support images to find the common object. However, SILCO explores the dense similarity between each support image and the query image while using coarse image-level features via a global average pooling to estimate the relation of support images. This experiment confirms that estimating proposal-level labels within the support images is quite important for common object localization.

4.1.1 Direct Comparison to Greedy Tree

To ensure a fair comparison, we also compare our common object localization unit to Greedy Tree by exactly following the original experimental protocol in [28]. Their algorithm utilizes a split of the COCO 2017 dataset with 63 base classes for training and 17 held-out novel classes for testing the algorithm. The trained model is also tested on a subset of the ILSVRC 2013 detection dataset with 148 novel

Table 1. *CorLoc (**top**) and mAP (**bottom**) performance of different few-shot common object localization methods on VOC07 test set. All of the models are trained on COCO60 and evaluated on a test query with $K = 5$ images in the support set. The best and second best performing methods are shown in **bold** and gray backgrounds respectively. *MI-SVM receives K extra negative images.*

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	CorLoc
MI-SVM* [34]	29.4	13.2	53.7	32.7	12.9	70.4	66.4	67.4	15.7	81.6	10.0	**67.6**	67.1	27.1	10.1	**16.7**	84.2	38.9	43.5	41.1	42.5
SILCO [12]	51.0	30.3	50.7	34.5	11.3	72.2	63.6	58.9	11.2	86.8	6.7	56.9	51.9	**49.2**	13.0	**16.7**	52.6	41.1	46.8	34.2	42.0
Greedy Tree [28]	35.3	21.1	**59.7**	34.5	**24.2**	**77.8**	73.4	61.1	**23.1**	89.5	15.0	64.7	73.4	25.4	12.8	13.3	**100.0**	**64.2**	61.3	**46.6**	48.8
vMF-MIL (ours)	**62.7**	**42.1**	53.7	**49.1**	6.5	68.5	**73.8**	**69.5**	19.4	**97.4**	**36.7**	65.7	**82.3**	40.7	**21.7**	15.0	94.7	**64.2**	**69.4**	31.5	**53.2**

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
MI-SVM* [34]	17.7	7.7	31.6	10.6	4.3	46.5	40.1	53.3	3.6	56.8	3.3	**56.3**	42.3	17.1	1.7	**8.1**	37.9	25.9	27.3	19.2	25.6
SILCO [12]	33.0	13.4	34.5	**14.8**	3.9	48.8	38.4	55.5	4.0	52.8	4.5	54.2	36.3	**27.3**	3.3	7.7	27.0	31.3	36.7	23.5	27.5
Greedy Tree [28]	26.0	8.7	37.4	11.5	**7.5**	52.4	47.7	45.8	**7.4**	61.1	4.5	47.3	50.7	15.6	2.3	5.0	**40.0**	**46.3**	47.3	**25.7**	29.5
vMF-MIL (ours)	**36.7**	**20.6**	**38.1**	14.2	1.9	**55.4**	**50.2**	**56.5**	**7.4**	**71.4**	**9.5**	56.2	**63.4**	16.8	**4.9**	3.4	39.3	43.0	**51.2**	23.0	**33.1**

Table 2. *CorLoc(%) and mAP(%) results of different methods for the task of common object localization on novel object classes on the COCO60 dataset with support set size $K = 5$ and $K = 10$. *MI-SVM receives K extra negative images.*

Model	K = 5		K=10	
	CorLoc@0.5	mAP@0.5	CorLoc@0.5	mAP@0.5
MI-SVM* [34]	30.5	15.9	33.2	16.2
SILCO [12]	29.7	14.8	31.3	15.8
Greedy Tree [28]	32.7	16.0	33.8	16.4
vMF-MIL (ours)	**35.7**	**19.6**	**38.2**	**20.2**

classes that have no overlap with the base classes. In Greedy Tree, a Faster-RCNN with ResNet50 [10] backbone is first trained on the base classes and used to extract features from all the images. To allow a fair comparison, we use the same feature set provided by the authors. To mimic common object localization during training, we sample tasks with $N = 1$ and $K = 8$ for training.

Similar to [28], we evaluate our model over 1000 randomly sampled tasks each containing $K = 8$ images with an object class in common. For each image, the proposal with the highest soft label in Eq. (3) is returned as the common object. We report the class-agnostic CorLoc ratio on COCO and ILSVRC datasets in Table 3 and compare it with the results in [28]. vMF-MIL outperforms Greedy Tree by 2.20% and 1.75% in MS COCO and ILSVRC datasets, respectively.

Table 3. *Class-agnostic CorLoc(%) with 95% confidence interval of the methods in [28] compared to our method. All methods use $K = 8$ positive images for finding the common object.*

method	COCO	ILSVRC13
TRWS [28]	64.53 ± 1.05	52.95 ± 1.09
ASTAR [28]	64.54 ± 1.05	52.89 ± 1.09
Greedy Tree [28]	64.65 ± 1.05	53.00 ± 1.10
vMF-MIL (ours)	$\mathbf{66.85 \pm 1.03}$	$\mathbf{54.75 \pm 1.09}$

4.2. Few-shot WSOD

We train our model on COCO60 for the task of few-shot WSOD with different N-way, K-shot problems and compare it with the knowledge-transfer MI-SVM model described in **??** on PASCAL VOC 2007 and MS COCO novel classes in Table 4. To highlight the importance of EM refinement, we also train our model with full image prototypical initialization without EM refinement. In both datasets, vMF-MIL outperforms MI-SVM in all the scenarios, demonstrating the strong generalization ability of our learning approach.

Table 4. *mAP(%) of different few-shot WSOD methods on COCO60 and PASCAL VOC datasets.*

Method	Dataset	$N = 5$		$N = 10$		$N = 20$	
		$K = 5$	K=10	K=5	K=10	K=5	K=10
Prototypical Init	VOC07	16.01	17.93	10.56	11.02	5.41	6.72
MI-SVM [34]		17.99	20.27	12.09	13.07	7.04	8.32
vMF-MIL (ours)		**21.22**	**22.01**	**14.54**	**15.83**	**8.83**	**10.19**
Prototypical Init	COCO60	8.90	9.28	4.65	6.07	2.99	3.26
MI-SVM [34]		11.40	11.60	7.30	7.80	2.97	3.70
vMF-MIL (ours)		**12.35**	**13.19**	**8.53**	**10.07**	**4.23**	**4.85**

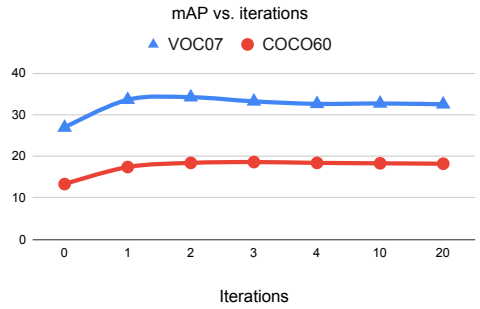

Figure 5. *mAP(%) vs. number of EM iterations in common object localization task with $K = 5$ on COCO60 and VOC07 datasets. The performance reaches a plateau at step 4.*

4.3. Large-Scale WSOD

Although our method is designed for low-shot settings, it is interesting to evaluate its performance in the standard WSOD setting as well. Related to our work is transfer learning approaches for large-scale WSOD [34, 11] with ImageNet detection as the standard benchmark. Typically, the first 100 classes are used as the base dataset and the remaining 100 classes with 65k images are used as novel

objects. We follow the setup in [34] and use the pre-trained Inception-Resnet Faster-RCNN model and weights provided by the authors to extract proposals, features, and the objectness scores. We apply our EM algorithm to the extracted features and use $u^-_{\boldsymbol{\omega}}(\mathbf{x}) = \alpha(1 - \mathrm{obj}(\mathbf{x}))$ for each proposal $\mathbf{x}$ where $\mathrm{obj}(\mathbf{x})$ is the Faster-RCNN objectness score and α is a hyper-parameter we tuned for the task. To our surprise, vFM-MIL outperforms [34] in Table 5 while being about $100\times$ faster. We believe this result can be further improved by relaxing some of the assumptions in our statistical model as overfitting may not be as significant in large-scale settings. For instance, we can learn a separate concentration parameter for each novel class in the EM steps. Furthermore, we can utilize the novel dataset to update the background scoring function u^-_{ω}. We defer these improvements and further analysis to the future work.

Table 5. *Large-Scale WSOD on ImageNet Detection.*

Model	CorLoc@0.5	Time (min.)
LSDA (JMLR 2016) [11]	28.8	-
Uijlings et al. (CVPR 18) [34]	74.2	900 (estimated)
vMF-MIL (ours)	**76.5**	10

4.4. Ablation Study

To understand which parts of the proposed method are critical for common object localization, we analyzed results in Table 2 with $K = 5$ for each of the important components of the proposed method in Table 6. These components are: initializing $\boldsymbol{\theta}$ and κ using features extracted from the complete image (Prototypical Init), updating $\boldsymbol{\theta}$, updating κ, and learning background distribution $p^-_{\boldsymbol{\omega}}$ to steer the algorithm toward objects. The first entry (#1) in Table 6 shows that there is a huge performance gap when the background model is not used. This is expected, since without using the background model it may localize non-object patterns such as grass, water, building, etc. with similar appearances as the common object. Comparing the third entry (#3) with #5 and #6 reveals that updating both $\boldsymbol{\theta}$ and κ in the EM refinements is important and that increases CorLoc by 4.8% and mAP by 6.3%. The fourth entry shows the importance of initialization; the EM steps are only effective if $\boldsymbol{\theta}$ is initialized with the complete image proposal otherwise EM reaches a low quality local minimum.

The second part of Table 6 shows the advantage of using vMF to Gaussian distribution in the EM algorithm (see **??** for the details). Tukey's transformation [33] furthur improves the performance of the Gaussian model but vMF distribution still exhibits the best performance. We believe this is because feature vectors' direction better captures the semantic information.

Finally, we illustrate the performance improvement vs. the number of EM steps in Fig. 5. In both VOC07 and COCO60 datasets, mAP reaches a plateau showing that the algorithm converges quickly. Qualitative results in Fig. 6 depict successful cases where EM refinements improve the top prediction.

Table 6. *Ablation study on COCO60 dataset. #1-6 show the importance of initialization, iterative EM updates, and learning the background model. #7-9 compare different statistical models in the EM algorithm.*

#	Random Init	Prototypical Init	Update $\boldsymbol{\theta}$	Update κ	$p^-_{\boldsymbol{\omega}}$	CorLoc	mAP
1		✓	✓			1.9	0.6
2					✓	22.8	9.3
3		✓			✓	30.9	13.3
4	✓		✓		✓	30.1	14.2
5		✓	✓		✓	34.8	18.6
6		✓	✓	✓	✓	**35.7**	**19.6**
	Gaussian	Tukey+Gaussian	vMF				
7	✓					29.8	13.9
8		✓				34.0	17.3
9			✓			**35.7**	**19.6**

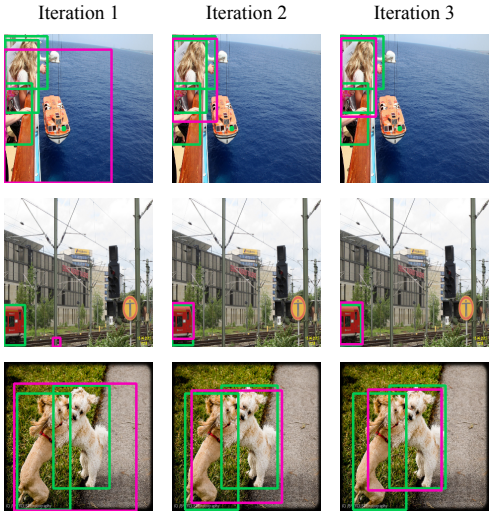

Figure 6. *Bounding box adjustments at each iteration for the common object localization experiment on COCO60 with $K = 5$. Only the top prediction in the query image is shown (in pink color) for each iteration. Ground-truth bounding boxes of the target classes are shown in green. EM refinements improve the target object localization in the query image.*

5. Conclusion

We have presented vMF-MIL, a multiple-instance learning framework to address the problem of few-shot common object localization and WSOD. vMF-MIL uses a simple inductive bias in learning to combat the overfitting issue in few-shot learning. Specifically, instances of each class are assumed to form a cluster on a unit hypersphere, where the mean corresponds to the class prototype. Our experiments on few-shot common object localization illustrate the advantage of our simple approach over several state-of-the-art methods, improving the few-shot WSOD performance compared with the strong MI-SVM baseline.

References

[1] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Adv. Neural Inform. Process. Syst.*, 2003.

[2] Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 2005.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

[4] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Localizing objects while learning their appearance. In *Eur. Conf. Comput. Vis.*, 2010.

[5] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. *Adv. Neural Inform. Process. Syst.*, 2020.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *Int. Conf. Mach. Learn.*, 2017.

[8] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[9] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Multi-fold mil training for weakly supervised object localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2014.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[11] Judy Hoffman, Deepak Pathak, Eric Tzeng, Jonathan Long, Sergio Guadarrama, Trevor Darrell, and Kate Saenko. Large scale visual recognition through adaptation using joint representation and multiple instance learning. *The Journal of Machine Learning Research*, 2016.

[12] Tao Hu, Pascal Mettes, Jia-Hong Huang, and Cees GM Snoek. Silco: Show a few images, localize the common object. In *Int. Conf. Comput. Vis.*, 2019.

[13] Maximilian Ilse, Jakub M Tomczak, and Max Welling. Attention-based deep multiple instance learning. *Int. Conf. Mach. Learn.*, 2018.

[14] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and T revor Darrell. Few-shot object detection via feature reweighting. In *Int. Conf. Comput. Vis.*, 2019.

[15] Daesik Kim, Gyujeong Lee, Jisoo Jeong, and Nojun Kwak. Tell me what they're holding: Weakly-supervised object detection with transferable knowledge from human-object interaction. In *AAAI*, 2020.

[16] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[17] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *ACCV*, 2018.

[18] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Image co-localization by mimicking a good detector's confidence score distribution. In *Eur. Conf. Comput. Vis.*, 2016.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Eur. Conf. Comput. Vis.*, 2014.

[20] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *Int. Conf. Learn. Represent.*, 2019.

[21] Xiankai Lu, Wenguan Wang, Chao Ma, Jianbing Shen, Ling Shao, and Fatih Porikli. See more, know more: Unsupervised video object segmentation with co-attention siamese networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[22] Minh Hoai Nguyen, Lorenzo Torresani, Fernando De La Torre, and Carsten Rother. Weakly supervised discriminative localization and classification: a joint learning process. In *Int. Conf. Comput. Vis.*, 2009.

[23] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[24] Hang Qi, Matthew Brown, and David G Lowe. Low-shot learning with imprinted weights. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[25] Amir Rahimi, Amirreza Shaban, Thalaiyasingam Ajanthan, Richard Hartley, and Byron Boots. Pairwise similarity knowledge transfer for weakly supervised object localization. In *Eur. Conf. Comput. Vis.*, 2020.

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[27] Pau Rodríguez, Issam Laradji, Alexandre Drouin, and Alexandre Lacoste. Embedding propagation: Smoother manifold for few-shot classification. *Eur. Conf. Comput. Vis.*, 2020.

[28] Amirreza Shaban, Amir Rahimi, Shray Bansal, Stephen Gould, Byron Boots, and Richard Hartley. Learning to find common objects across few image collections. In *Int. Conf. Comput. Vis.*, 2019.

[29] Mennatullah Siam, Naren Doraiswamy, Boris N Oreshkin, Hengshuai Yao, and Martin Jagersand. Weakly supervised few-shot object segmentation using co-attention with visual and semantic embeddings. *IJCAI*, 2020.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Int. Conf. Learn. Represent.*, 2015.

[31] Parthipan Siva and Tao Xiang. Weakly supervised object detector learning with model drift detection. In *Int. Conf. Comput. Vis.*, 2011.

[32] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Adv. Neural Inform. Process. Syst.*, 2017.

[33] John W Tukey. *Exploratory data analysis*. Addison-Wesley Series in Behavioral Science. Addison-Wesley, Reading, MA, 1977.

[34] Jasper Uijlings, Stefan Popov, and Vittorio Ferrari. Revisiting knowledge transfer for training object class detectors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[35] Sara Vicente, Carsten Rother, and Vladimir Kolmogorov. Object cosegmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011.

[36] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Adv. Neural Inform. Process. Syst.*, 2016.

[37] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1041–1049, 2017.

[38] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *Int. Conf. Mach. Learn.*, 2020.

[39] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Int. Conf. Comput. Vis.*, 2019.

[40] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. *Eur. Conf. Comput. Vis.*, 2020.

[41] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Int. Conf. Comput. Vis.*, 2019.

[42] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *Int. Conf. Learn. Represent.*, 2021.

[43] Dingwen Zhang, Junwei Han, Chao Li, and Jingdong Wang. Co-saliency detection via looking deep and wide. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.