Appendix

Few-shot Weakly-Supervised Object Detection via Directional Statistics

Paper #1337

A. Expectation-Maximization Derivation

Recall that each proposal has a (latent) binary label $\mathbf{z}_{ij} \in \{0, 1\}$ that indicates whether the proposal tightly encloses the common object. Following the best practice of the previous works in WSOD [28, 37, 10], we assume there is exactly one proposal with label 1 (positive proposal) in each image and the rest are negative proposals, i.e., $\mathbf{z}_i \in \{\mathbf{e}_1, \dots \mathbf{e}_P\}$ where $\mathbf{e}_j \in \{0, 1\}^P$ is the j-th canonical basis.

Assuming that the images are sampled independently given the common class c, the full likelihood function is given by

$$\begin{aligned} l(\boldsymbol{\theta}; \mathcal{F}) = p(\mathcal{F}|\boldsymbol{\theta}) &= \prod_{i=1}^{M} p(F_i|\boldsymbol{\theta}) \\ &= \prod_{i=1}^{M} \sum_{j=1}^{P} p(F_i, \mathbf{z}_i = \mathbf{e}_j|\boldsymbol{\theta}) \ , \end{aligned} \tag{9}$$

where $\boldsymbol{\theta} \in \mathbb{S}^{d-1}$ is the mean direction of the common class distribution. Note that the last equation integrates over all possible values of $\mathbf{z}_i$. Assuming that proposals are i.i.d samples from their corresponding distributions given their labels $\mathbf{z}_i$[2], we can write

$$p(F_i|\mathbf{z}_i = \mathbf{e}_j, \boldsymbol{\theta}) = p^+_{\boldsymbol{\theta}}(F_{ij}) \prod_{\substack{k=1 \\ k \neq j}}^{P} p^-_{\boldsymbol{\omega}}(F_{ik}) \ , \tag{10}$$

where F_{ij} is the feature of the j-th proposal in F_i, $p^+_{\boldsymbol{\theta}}$ is the generic distribution that generates the common class proposals, and $p^-_{\boldsymbol{\omega}}$ represents background proposals' distribution. For brevity, let us re-write Eq. (10) in a more compact form

$$p(F_i|\mathbf{z}_i = \mathbf{e}_j, \boldsymbol{\theta}) = q_{\boldsymbol{\theta}}(F_{ij}) p^-_{\boldsymbol{\omega}}(F_i) \ , \tag{11}$$

where $q_{\boldsymbol{\theta}}$ is quotient of object and background distributions $q_{\boldsymbol{\theta}}(\mathbf{x}) = p^+_{\boldsymbol{\theta}}(\mathbf{x}) / p^-_{\boldsymbol{\omega}}(\mathbf{x})$, and $p^-_{\boldsymbol{\omega}}(F_i) = \prod_{j=1}^{P} p^-_{\boldsymbol{\omega}}(F_{ij})$.

We adopt the EM algorithm to maximize the likelihood in Eq. (9) by iteratively optimizing the surrogate expected log-likelihood which is easier to compute. In the E-step, the posterior distribution of the latent variables $\mathbf{w}_{ik} = p(\mathbf{z}_i = \mathbf{e}_k|F_i, \boldsymbol{\theta})$ are computed for the current $\boldsymbol{\theta}$. By Eq. (11), using Bayes' theorem, and assuming a uniform distribution over image labels $\mathbf{z}_i$, the posterior can be expressed in terms of quotient of distributions defined above

$$\mathbf{w}_{ik} = \frac{q_{\boldsymbol{\theta}}(F_{ik})}{\sum_{j=1}^{P} q_{\boldsymbol{\theta}}(F_{ij})} \ , \tag{12}$$

yielding soft label vector $\mathbf{w}_i \in \mathbb{R}^P$ for the i-th image proposals.

By plugging in the vMF probability density function of the common class and background probability density function, the quotient $q_{\boldsymbol{\theta}}$ can be written as

$$q_{\boldsymbol{\theta}}(\mathbf{x}) \propto \exp\left(\kappa \boldsymbol{\theta}^\top \mathbf{x} - \log u^-_{\boldsymbol{\omega}}(\mathbf{x})\right) . \tag{13}$$

As shown in Algorithm 1, one can compute the soft labels via the softmax operation, resembling the attention mechanism recently used for MIL [14].

In the M-step, parameters $\boldsymbol{\theta}$ are updated by maximizing the surrogate expected log-likelihood using the posteriors computed in the E-step

$$l(\boldsymbol{\theta}'; \boldsymbol{\theta}) = \sum_{i=1}^{M} \mathbb{E}_{p(\mathbf{z}_i|F_i, \boldsymbol{\theta})}\Big[\log p(F_i|\mathbf{z}_i, \boldsymbol{\theta}')\Big] = \sum_{i=1}^{M} \sum_{k=1}^{P} \mathbf{w}_{ik} \log p(F_i|\mathbf{z}_i = \mathbf{e}_k, \boldsymbol{\theta}') \ , \tag{14}$$

[2] i.i.d assumption is a standard approach in MIL and works well in practice. See [24] for more details.

where the weights $\mathbf{w}_{ik}$ are computed in Eq. (29). Lagrangian function is written as

$$\mathcal{L}(\boldsymbol{\theta}', \lambda) = l(\boldsymbol{\theta}'; \boldsymbol{\theta}) - \lambda(\|\boldsymbol{\theta}'\|^2 - 1) \tag{15}$$

By plugging in the log-likelihood term in Eq. (15) and computing the derivative w.r.t. $\boldsymbol{\theta}'$ and λ we have

$$\begin{aligned} \nabla_{\boldsymbol{\theta}'}\mathcal{L}(\boldsymbol{\theta}', \lambda) &= \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}\nabla_{\boldsymbol{\theta}'} \log\Big(p^{+}_{\boldsymbol{\theta}'}(F_{ij}) \prod_{\substack{k=1 \\ k\neq j}}^{P} p^{-}_{\boldsymbol{\omega}}(F_{ik})\Big) - 2\lambda\boldsymbol{\theta}' = \kappa \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}F_{ik} - 2\lambda\boldsymbol{\theta}' \,. \\ \nabla_{\lambda}\mathcal{L}(\boldsymbol{\theta}', \lambda) &= -\|\boldsymbol{\theta}'\|^2 + 1. \end{aligned} \tag{16}$$

Finally, closed-form update rule

$$\begin{aligned} \boldsymbol{\theta} &\leftarrow \mathrm{norm}(\sum_{\mathrm{i}=1}^{\mathrm{M}} \tilde{\mathbf{x}}_{\mathrm{i}}) \,, \\ \text{where} \quad \tilde{\mathbf{x}}_i &= \mathbf{w}_i^{\top} F_i = \sum_{k=1}^{P} \mathbf{w}_{ik} F_{ik} \,, \end{aligned} \tag{17}$$

is derived by setting the derivatives to zero and solving for $\boldsymbol{\theta}'$ and λ.

A.1. Updating κ in M-Step

In this section, we propose a simple update rule for parameter κ that can be used along Eq. (17) in the M-step. As shown in Table 7, updating κ with a our order-0 rule further improves our vMF-MIL COL results of the paper.

To find the optimal κ, we compute the derivative of the Lagrangian function in Eq. (15) w.r.t. κ

$$\begin{aligned} \partial_{\kappa}\mathcal{L}(\boldsymbol{\theta}', \lambda) &= \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}\partial_{\kappa} \log\Big(p^{+}_{\boldsymbol{\theta}'}(F_{ij}) \prod_{\substack{k=1 \\ k\neq j}}^{P} p^{-}_{\boldsymbol{\omega}}(F_{ik})\Big) = \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}\partial_{\kappa} \log p^{+}_{\boldsymbol{\theta}'}(F_{ik}) \\ &= \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}\partial_{\kappa}(-\log Z(\kappa) + \kappa F_{ik}^{\top}\boldsymbol{\theta}') = -M\frac{\partial_{\kappa}Z(\kappa)}{Z(\kappa)} + \mathbf{r}^{\top}\boldsymbol{\theta}' \,, \end{aligned} \tag{18}$$

where $\mathbf{r} = \sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik}\mathbf{x}$ and $Z(\kappa)$ is vMF distribution normalization factor. A precise formula is known for $Z(\kappa)$, namely

$$Z(\kappa) = \frac{(2\pi)^{d/2}\, I_{d/2-1}(\kappa)}{\kappa^{d/2-1}} \,. \tag{19}$$

where d is the feature dimension and I is the modified Bessel function. This formula is quoted in [3](2.2), but it is upside-down compared to Eq. (19), since we are defining $c_d(\kappa) = 1/Z(\kappa)$.

By plugging in $\boldsymbol{\theta}' = \mathbf{r}/\|\mathbf{r}\|$ from Eq. (17) and setting the derivative to zero we get

$$\frac{\partial_{\kappa}Z(\kappa)}{Z(\kappa)} = \frac{\|\mathbf{r}\|}{M} = \bar{r} \,. \tag{20}$$

Equation (20) is similar to what we see in vMF maximum-likelihood estimation, therefore, we can use the maximum-likelihood derivations from now on (See Appendix of [3] equations A.7 to A.8) which leads to maximum-likelihood estimation

$$\kappa = A_d^{-1}(\bar{r}) \,, \tag{21}$$

where A_d is the ratio of Bessel functions,

$$A_d(\kappa) = \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)} \,. \tag{22}$$

It is still numerically difficult to compute the Bessel function in cases where d and κ are large. We are able to compute it in python using the `scipy.special.iv` function, only for values of d up to about 120 and κ up to about 700.

To address this difficulty, different formulae are given for estimating the optimal value of $\hat{\kappa}$.

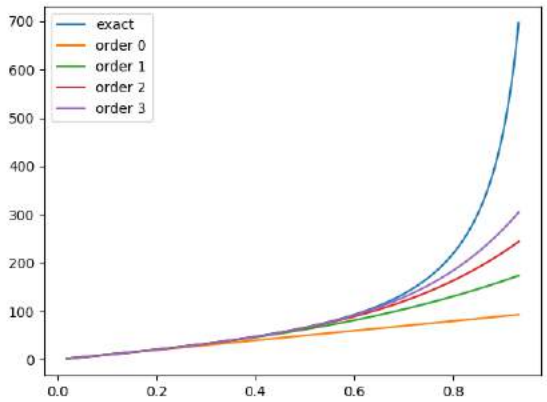

Figure 7. *Plot of different estimates of $\hat{\kappa}$ as a function of $\bar{r}$, for dimension $d = 100$. At this resolution, the exact estimate is indistinguishable from the estimate Eq.* (24)*. The graph also shows approximations of different orders, such as Eq.* (26) *and Eq.* (27)*, which are accurate for small-to-medium values of $\bar{r}$, but not for larger values. However, the exact value of $\bar{\kappa}$ is extremely sensitive to small variations in the value of $\bar{r}$, and it diverges to infinity as $\bar{r}$ approaches* 1*. For this reason it may not be good practice (as is verified by our experiments) to use the exact estimate of $\bar{\kappa}$ in clustering.*

1. A formula due to Mardia and Jupp [23] is given in [3](4.2) as

$$\hat{\kappa} \approx d\bar{r}\left(1 + \frac{d}{d+2}\bar{r}^2 + \frac{d^2(d+8)}{(d+2)^2(d+4)}\bar{r}^4\right) .$$

   For large values of d, this is almost the same as

$$\hat{\kappa} \approx d\bar{r}\left(1 + \bar{r}^2 + \bar{r}^4\right) . \tag{23}$$

2. Banerjee *et al*. [3] derive an estimate (unnumbered, above [3] (4.4)),

$$\hat{\kappa} \approx \frac{d\bar{r}}{1 - \bar{r}^2} \tag{24}$$

   which has series expansion

$$\hat{\kappa} \approx d\bar{r}(1 + \bar{r}^2 + \bar{r}^4 + \ldots) . \tag{25}$$

   Since $0 \leq \bar{r} < 1$, this series will converge, albeit slowly for $\bar{r}$ close to 1. Thus, it is seen that Eq. (23) is simply a truncated approximation to the infinite series Eq. (25). We shall refer to Eq. (23) (perhaps somewhat inexactly) as the "order-2" approximation to Eq. (25), since the approximation to $1/(1 - \bar{r}^2)$ contains terms up to second order in $\bar{r}^2$.

3. It is also possible to consider approximations of other orders for Eq. (24), including in particular the 0-order approximation

$$\hat{\kappa} \approx d\bar{r} , \tag{26}$$

   first order approximation

$$\hat{\kappa} \approx d\bar{r}\left(1 + \bar{r}^2\right) , \tag{27}$$

   and the third-order approximation.

4. Another empirically derived formula is also given in [3](4.4). However, we observe (see Fig. 7) that the approximation Eq. (24) is already a very close approximation, and the use of [3](4.4) is not warranted.

We show graphs of the approximations of $\hat{\kappa}$ for various approximation, and the exact solution in Fig. 7.

Table 7. *CorLoc(%) and mAP(%) results with κ estimations for the task of COL on novel object classes on the COCO60 dataset with support set size $K = 5$ and $K = 10$.*

$\hat{\kappa}$	K = 5		K=10	
	CorLoc@0.5	mAP@0.5	CorLoc@0.5	mAP@0.5
Constant	34.8	18.6	36.9	20.0
$d\bar{r}(1+\bar{r}^2+\bar{r}^4+\ldots)$	20.1	11.3	24.6	13.7
$d\bar{r}(1+\bar{r}^2+\bar{r}^4+\bar{r}^6)$	31.8	17.7	34.7	19.0
$d\bar{r}(1+\bar{r}^2+\bar{r}^4)$	33.1	18.6	36.3	19.5
$d\bar{r}(1+\bar{r}^2)$	34.7	19.0	37.5	19.9
$d\bar{r}$	**35.7**	**19.6**	**38.2**	**20.2**

Results As pointed out in the caption to Fig. 7, the exact estimate of $\hat{\kappa}$ may not be a good choice for clustering, in the case where $\bar{r}$ approaches 1 (meaning that the data has small spread).

In Table 7, we try all the different formulas described above to estimate a value of $\hat{\kappa}$. We also report the results in the main paper where κ is kept constant. The experiments show the following outcomes.

1. Order-∞ in Eq. (24) performs notably worse, presumably because of the sensitivity to the value of $\bar{r}$, which is computed from a relatively small number of samples in our few-shot learning scenario.

2. The order-1 to order-3 estimates perform approximately the same as fixing $\hat{\kappa}$.

3. The order-0 approximation, $\hat{\kappa} = d\bar{r}$, gives the best results. In our COL experiments, $d = 512$, so setting $\hat{\kappa} = d\bar{r}$ places an upper bound of 512 on the value of $\hat{\kappa}$.

B. Modeling with Gaussian Distribution

In Section 4.4, we conduct experiment with Gaussian distribution used to model the common object distribution. We assume the common object distribution is Gaussian with mean $\boldsymbol{\theta}$ and diagonal covariance matrix $\sigma^2 I$, i.e., $\mathbf{x} \sim \mathcal{N}(\boldsymbol{\theta}, \sigma I)$

$$p^{+}_{\boldsymbol{\theta}_c}(\mathbf{x}) = \frac{1}{Z} \exp\left(-\frac{\|\mathbf{x}-\boldsymbol{\theta}\|^2}{2\sigma^2}\right) , \tag{28}$$

and plug the distribution into Eq. (12) to get the soft label update rule as

$$\mathbf{w}_{ik} = \frac{\exp\left(-\frac{\|F_{ik}-\boldsymbol{\theta}\|^2}{2\sigma^2} - \log u^{-}_{\boldsymbol{\omega}}(F_{ik})\right)}{\sum_{j=1}^{P} \exp\left(-\frac{\|F_{ij}-\boldsymbol{\theta}\|^2}{2\sigma^2} - \log u^{-}_{\boldsymbol{\omega}}(F_{ij})\right)} . \tag{29}$$

E-step is computed by setting the derivative of Eq. (14) w.r.t. $\boldsymbol{\theta}$ to zero

$$\boldsymbol{\theta} \leftarrow \frac{1}{M}\sum_{i=1}^{M}\sum_{k=1}^{P} \mathbf{w}_{ik} F_{ik} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{w}_i^{\top} F_i , \tag{30}$$

C. MI-SVM WSOD Baseline

To the best of our knowledge there is no WSOD algorithm for few-shot setting in the literature. However, WSOD with knowledge transfer methods [28, 37, 12, 5] are closely related to our work. In this section, we describe a slightly modified version of [37] and discuss its differences to the proposed method. In Section 4, we empirically compare our work against [37].

Similar to our approach, each image is represented as a bag of bounding box proposals B and their features F. Learning is performed on one target class $c \in \mathcal{L}$ at a time. Similar to our WSOD approach, the support set is split into positive images which has the target class and a negative set of images without the target class. Then, a linear SVM appearance model is employed to iteratively learn class c by alternating between two steps:

- **Re-training:** Train a binary SVM given the currently selected proposals from the positive images and the proposals in negative images.
- **Re-localization:** Given the current SVM select the proposal with the highest score from each positive image. In [37], the re-localization is guided by a class-agnostic objectness measure to guide the selection toward objects. Therefore, the selection for a positive image $\mathbf{I}$ with bounding box proposals B is updated as

$$b^* = \arg\max_{b \in B} \quad \text{SVM}(\mathbf{I}, b) + \gamma O(\mathbf{I}, b) \;, \tag{31}$$

where O is the objectness model and γ adjusts its importance.

The algorithm is initialized with complete image bounding box proposal and alternates between above steps until convergence. We use highly efficient GPU solver in [1] for SVM optimization. Finally, test proposal $\mathbf{x}$ from the test set $\mathcal{D}_{\text{test}}$ is scored using the SVM trained for each class. To have a fair comparison, we use the same Faster-RCNN model trained on the base classes as we used in our model to extract bounding box and feature proposals from all images. For the objectness model O, we learn a class-agnostic logistic regression model on the extracted feature. We employ the hard negative mining in [37] to improve the performance of the classifier. In this approach the negative set is initialized with full negative image features and the hardest negative proposal within each image is added to the negative set after each re-training step.

The expectation and maximization steps in our method are analogous to re-localization and re-training steps in [37]. In MI-SVM, only the proposal with the highest score is labeled positive in the re-localization step while our COL method infers soft labels in the expectation step via an attention mechanism. Using soft labels could be beneficial as they reflect the uncertainty in choosing the common object.

D. Qualitative Results

We show some of the success cases of the experiments in the paper. Our first example in Fig. 8 shows vMF-MIL performance on a single few-shot WSOD problem. Given the support set with only image-level annotations the algorithm learns to detect the target objects in the query set. We sample 4 query images to evaluate the algorithm performance in detecting different target objects. Except `person` in the first query image, vMF-MIL successfully detects other target objects. More successful few-shot WSOD results are shown in Fig. 9, 10, 11, 12 and 13.

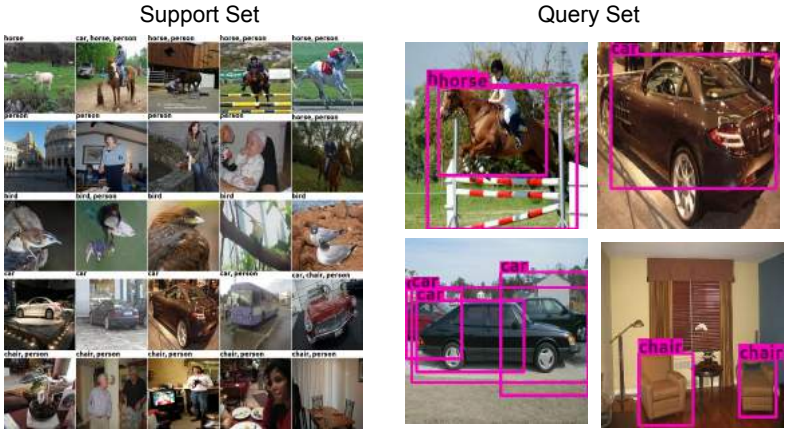

Figure 8. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object on the 4 different query images on the right side. The algorithm fails to detect person in the first query image but successfully detects other target objects.*

Finally, Fig. 14 shows some of the success cases in localizing the target object in the query image for the task of common object localization in Section 4.1. All the target objects (`dog, car, cow, train, boat, bus, sofa, horse, person`) shown in this figure are novel. Also, ground-truth annotations are only shown for better visualization and are not used in learning.

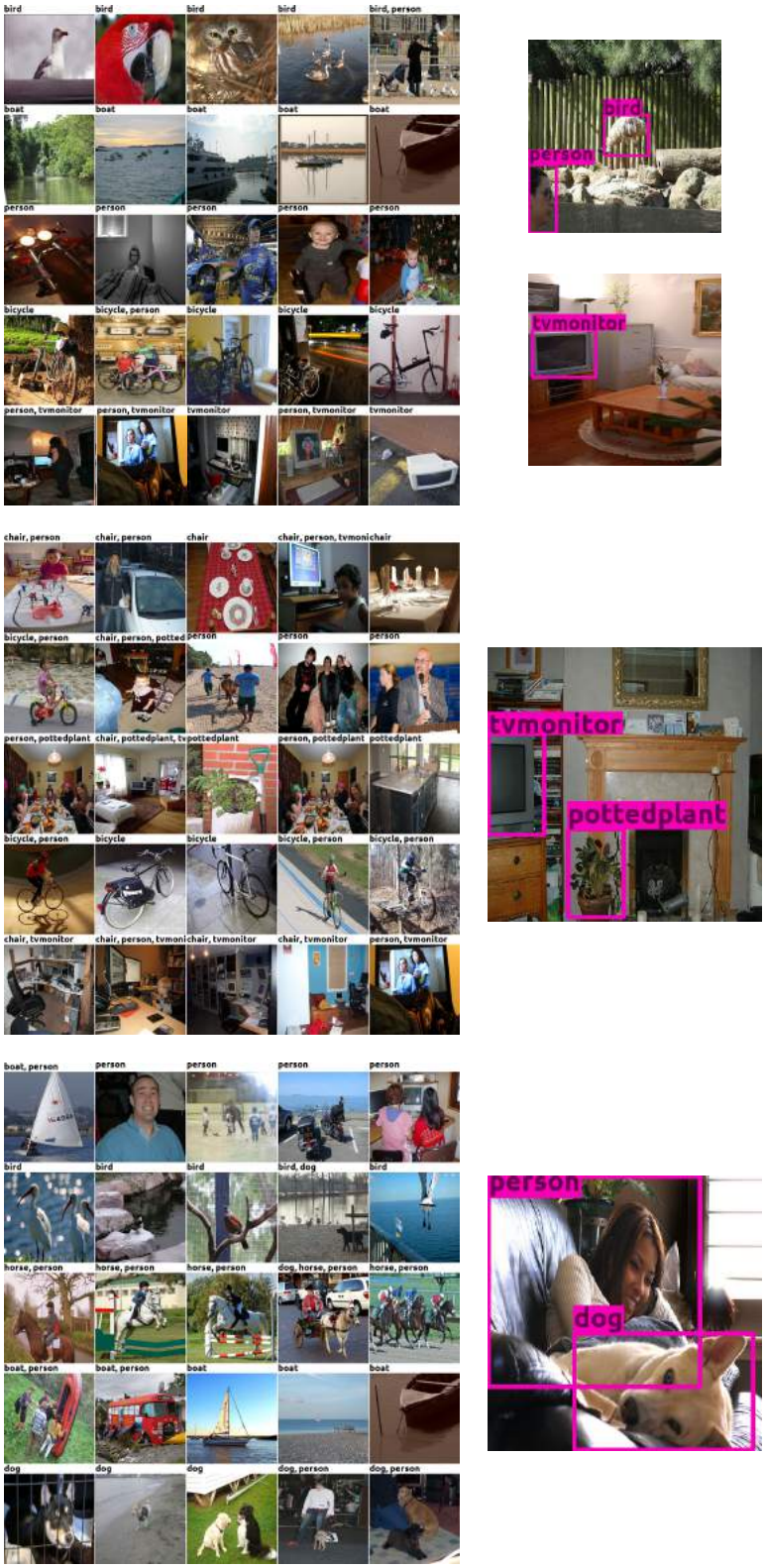

Figure 9. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object in query images on the right side.*

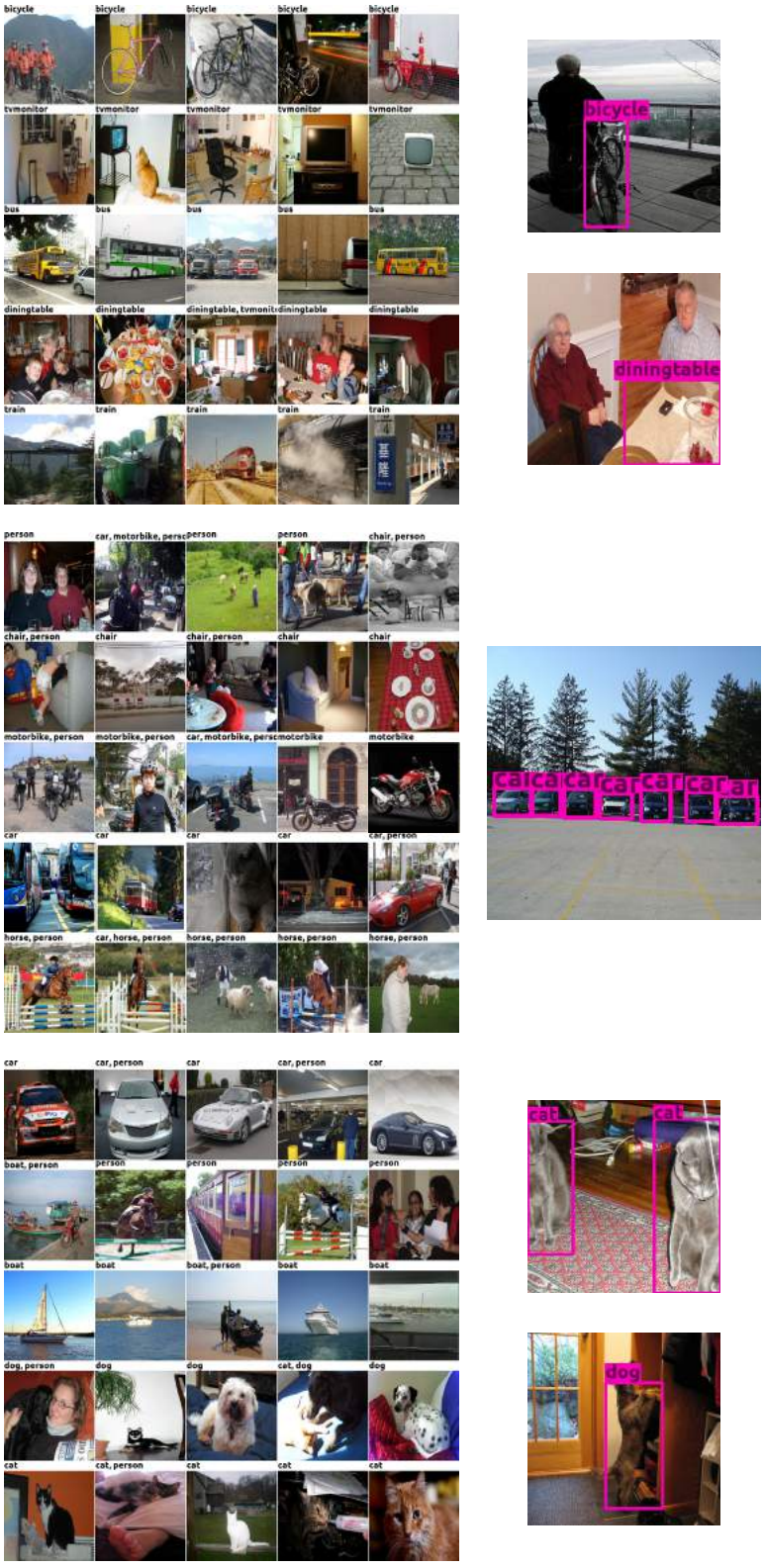

Figure 10. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object in query images on the right side.*

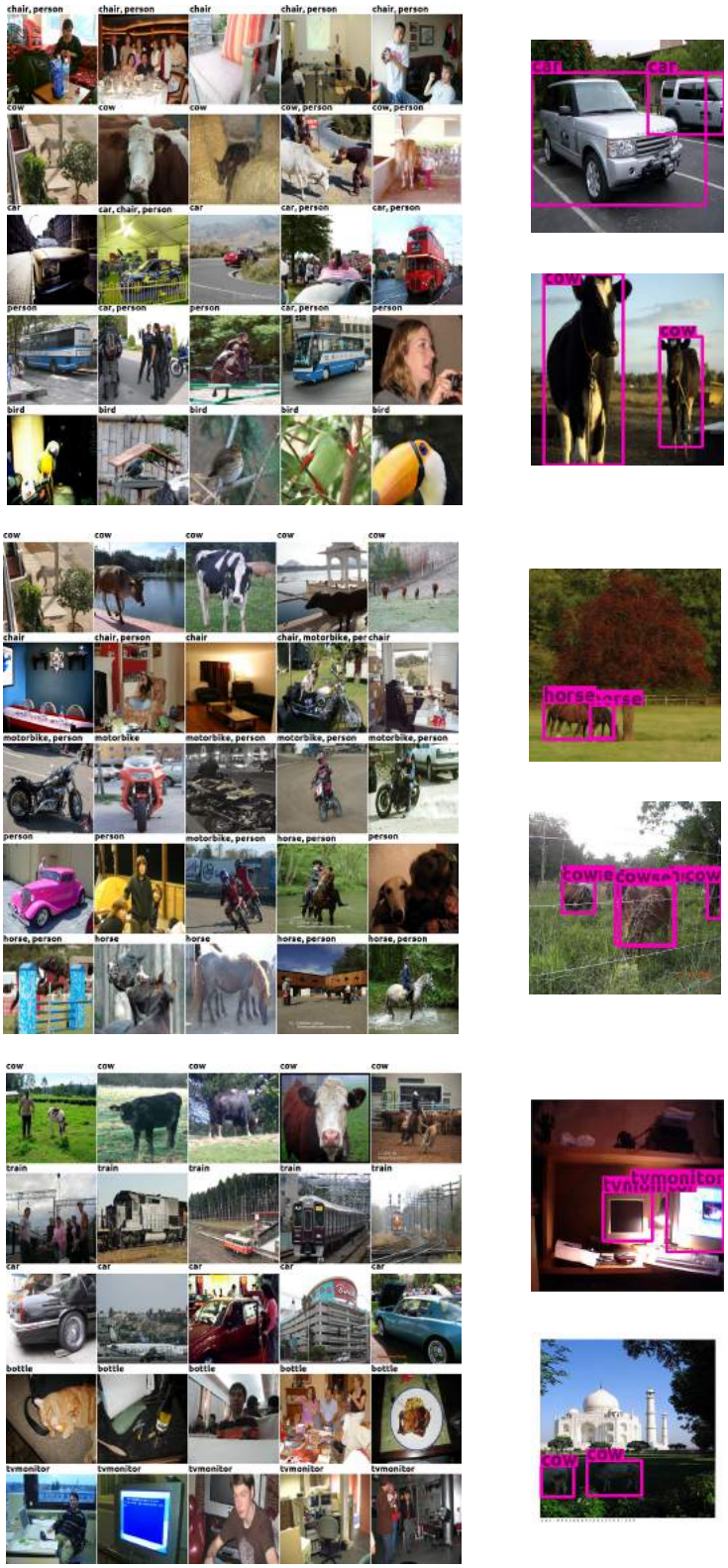

Figure 11. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object in query images on the right side.*

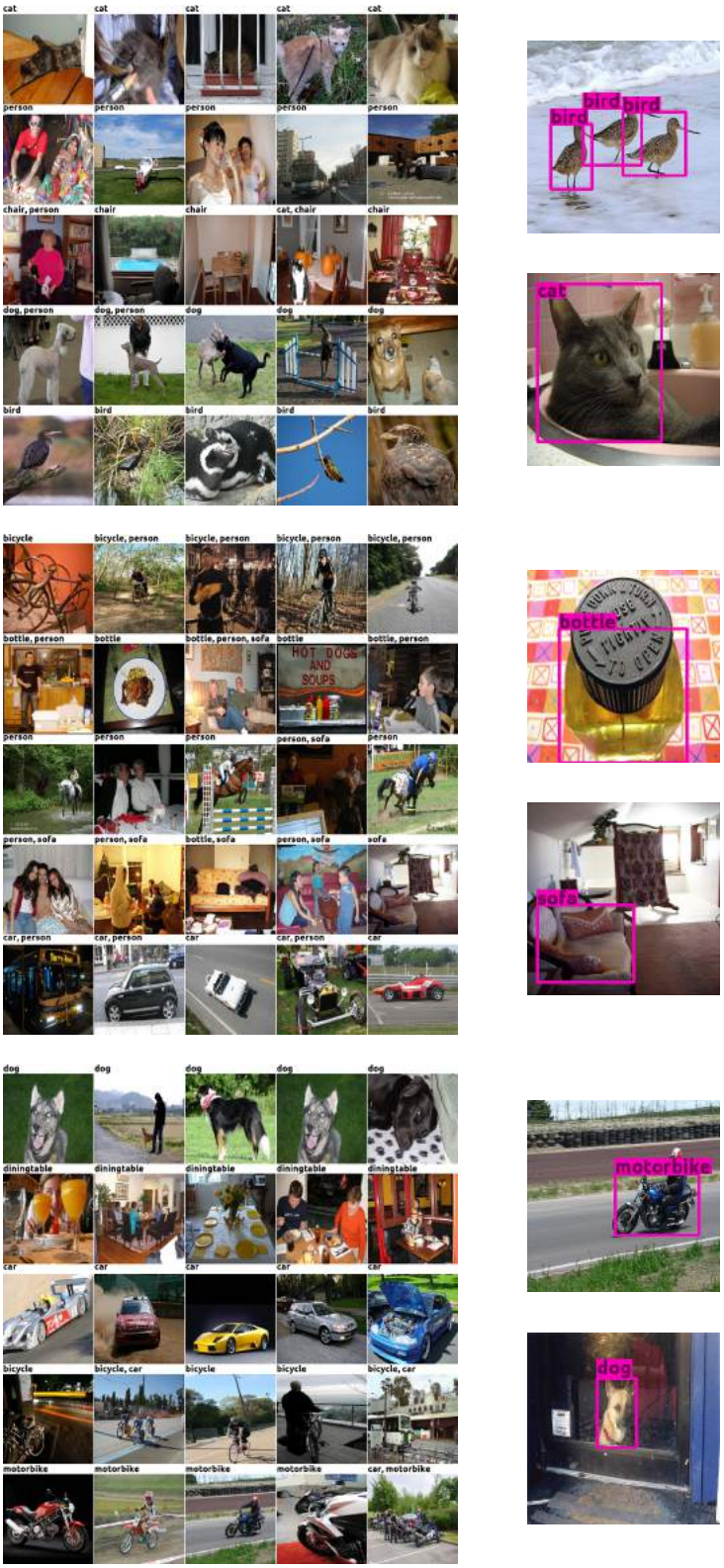

Figure 12. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object in query images on the right side.*

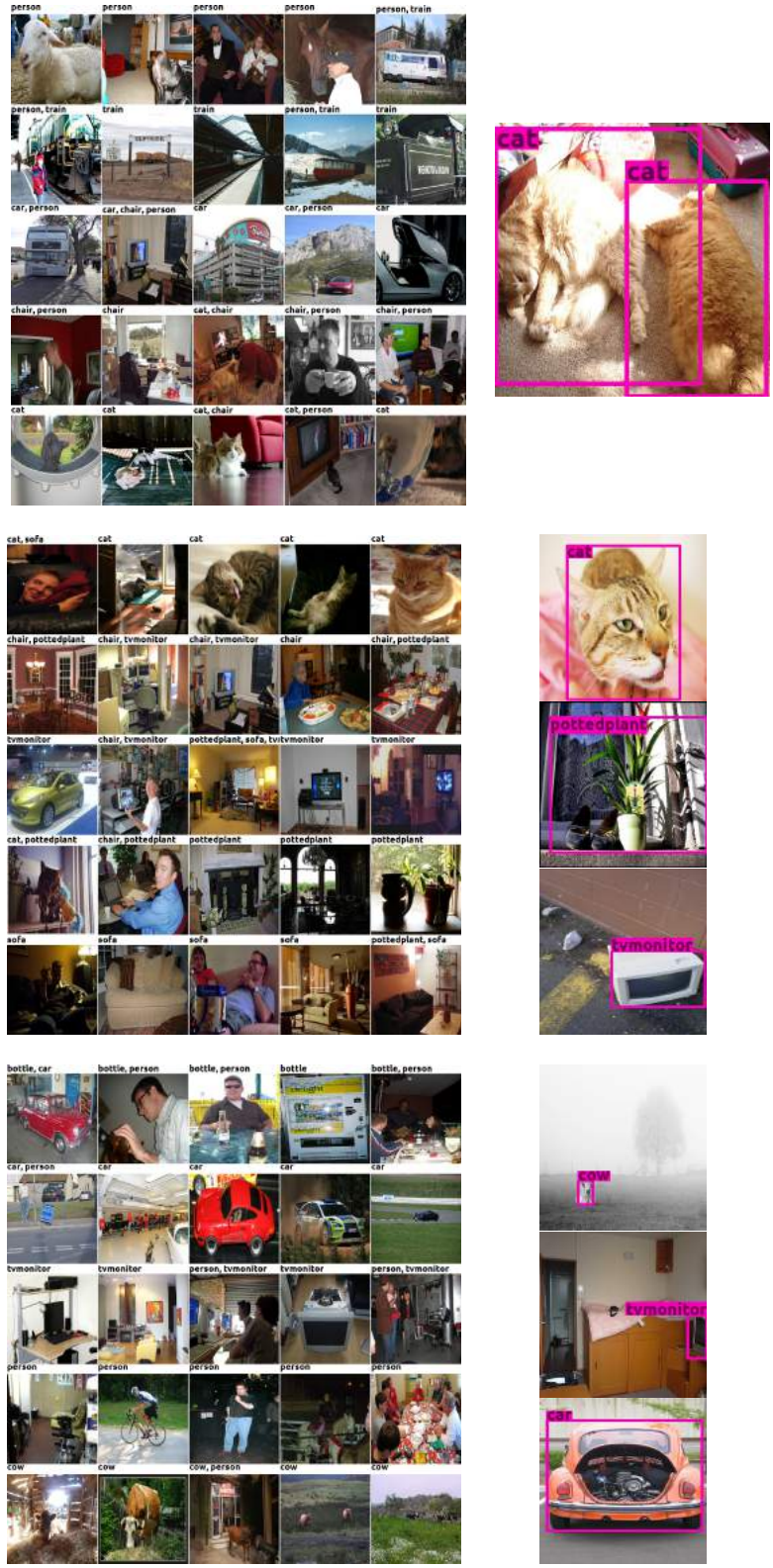

Figure 13. *Few-shot WSOD on PASCAL VOC with $N = K = 5$. Given the support set shown on the left side the algorithm detects the object in query images on the right side.*

Figure 14. *5-shot common object localization ($N = 1$) on MS COCO. Each row shows one common object localization problem. Ground-truth annotations (shown in green) are just for visualization and are not used in the algorithm. Top query bounding box prediction for each problem is shown in pink.*