Supplementary: Rethinking Few-Shot Object Detection on a Multi-Domain Benchmark

Kibok Lee[1,2⋆], Hao Yang[1†], Satyaki Chakraborty[1], Zhaowei Cai[1],
Gurumurthy Swaminathan[1], Avinash Ravichandran[1], and Onkar Dabeer[1]

[1]AWS AI Labs [2]Yonsei University
{kibok,haoyng,satyaki,zhaoweic,gurumurs,ravinash,onkardab}@amazon.com
kibok@yonsei.ac.kr

A Natural K-Shot Sampling

In this section, we describe how we perform natural K-shot sampling in detail:

Step 1. Sample $C \times K$ images. C is the number of classes of the original dataset $\mathcal{S}$. In this step, without worrying about class labels, we sample $\mathcal{S}$ from the entire dataset $\mathcal{D}$. Unlike the standard K-shot sampling algorithm in recent FSOD works [11,26,21], we do not apply stratified sampling. This is because an image usually contains multiple annotations, such that stratified sampling might result in an artificial class distribution [11].

Step 2. Check missing classes. The initial sampled dataset might not contain some classes, particularly those present only in a few images in the original dataset. To compensate for this, we check if there are any missing classes and update the sampled dataset. Specifically, we manage two datasets: $\mathcal{P}$ is a set of images to be added, and $\mathcal{Q}$ is a set of images to be kept. Then, for each class, if no image in $\mathcal{S}$ contains the class, we sample an image from the $\mathcal{D}$ containing the class and put it in $\mathcal{P}$; otherwise, we sample an image from $\mathcal{S}$ containing the class and put it in $\mathcal{Q}$.

Step 3. Update the sampled dataset. As the final step, we adjust the initial dataset $\mathcal{S}$ to guarantee that all classes are present. To match the number of added and removed images, we sample a set of images to be removed $\mathcal{R}$ from $\mathcal{S} - \mathcal{Q}$ where the size of $\mathcal{R}$ is the same as $\mathcal{P}$. Here, $\mathcal{Q}$ guarantees that any class in $\mathcal{S}$ does not become empty. Finally, we add $\mathcal{P}$ and remove $\mathcal{R}$ from $\mathcal{S}$.

The complete algorithm is in Algorithm 1.

B Dataset Size Reduction

We initially collected more than 100 public detection datasets, and then selected 32 datasets based on availability, diversity of domains, annotation quality, and number of citations. After initial experiments on them, to reduce the computational burden for future research, we picked 10 datasets out of the 32, which show similar performance trends with the 32 datasets, while covering a variety of domains based on the domain distance.

Algorithm 1 Natural K-shot sampling algorithm.

Input: Dataset $\mathcal{D}$, classes $\mathcal{Y}$, number of classes $C = |\mathcal{Y}|$, average shot number K
Output: Sampled dataset $\mathcal{S}$
if $|\mathcal{D}| \leq C \times K$ **then**
  $\mathcal{S} \leftarrow \mathcal{D}$
else
  // Step 1: sample an initial dataset
  Sample $\mathcal{S} \subset \mathcal{D}$ where $|\mathcal{S}| = N \times K$
  // Step 2: check missing classes
  $\mathcal{P} = \{\}$ // images to be added
  $\mathcal{Q} = \{\}$ // images to be kept
  for $y \in \mathcal{Y}$ **do**
    if no image in $\mathcal{S}$ contains y **then**
      Sample $I \in \mathcal{D}$ where I contains y
      $\mathcal{P} \leftarrow \mathcal{P} \cup \{I\}$
    else
      Sample $I \in \mathcal{S}$ where I contains y
      $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{I\}$
    end if
  end for
  // Step 3: update the sampled dataset
  if $|\mathcal{P}| > 0$ **then**
    Sample $\mathcal{R} \subset \mathcal{S} - \mathcal{Q}$ where $|\mathcal{R}| = |\mathcal{P}|$
    $\mathcal{S} \leftarrow (\mathcal{S} \cup \mathcal{P}) - \mathcal{R}$
  end if
end if

In the proposed MoFSOD benchmark, several datasets contain a large number of classes and testing images, such as LogoDet-3K. With the proposed natural K-shot sampling, the training time is proportional to the number of classes. To address concerns on computational cost and speed up overall experiment time, we limited the number of classes to 50 and the number of test samples to 1k.

Specifically, we randomly sample 50 classes and remove images containing all the rest classes in each episode, such that the intention of the original datasets is kept, *i.e.*, all remaining logos or traffic signs should be detected. We note that all classes in these datasets are mostly isolated to certain images, such that removing images containing a class does not hurt the distribution of other classes. We confirmed that the performance differences between sampled and full test sets are less than 1.5% for all datasets.

C Additional Experimental Results

C.1 Dataset Statistics

More detailed statistics of the ten datasets of MoFSOD can be found in Table C.1.

Table C.1: Statistics of 10 datasets in the proposed benchmark. For KITTI We use the merged set of classes from the universal object detection benchmark [27].

Domain	Dataset	# classes	# train images	# train anno.	# test images	# test anno.
Aerial	VisDrone	10	7019	381965	1610	75103
Agriculture	DeepFruits	7	457	2553	114	590
Animal	iWildCam	1	21065	31591	5313	7901
Cartoon	Clipart	20	500	1640	500	1527
Fashion	iMaterialist	46	45623	333402	1158	8782
Food	Oktoberfest	15	1110	2697	85	236
Logo	LogoDet-3K	352	18752	35264	8331	15945
Person	CrowdHuman	2	15000	705967	4370	206231
Security	SIXray	5	7496	15439	1310	2054
Traffic	KITTI	4	5481	38077	7481	52458

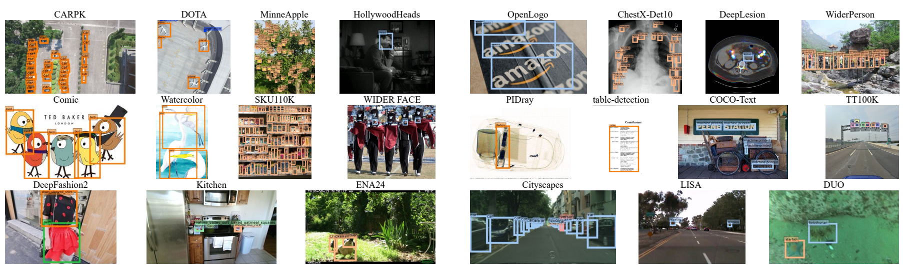

Fig. C.1: Image samples of the additional datasets.

C.2 Detailed 1-, 3- and 10-shot Results

In addition to per-dataset 5-shot results in Table 3 of the main paper, we present per-dataset 1-, 3- and 10-shot results in Table C.2, C.3, and C.4, respectively.

C.3 Extended 32 Datasets Results

We evaluate *FT* against SOTA methods in an extended 32 dataset benchmark on 17 Domains. These datasets are: CARPK [9], DOTA [28], and VisDrone [33] in aerial images, DeepFruits [17] and MinneApple [8] in agriculture, ENA24 [31] and iWildCam [1] in animal in the wild, Clipart, Comic, and Watercolor [10] in cartoon, SKU110K [6] in dense product, DeepFashion2 [3] and iMaterialist [7] in fashion, WIDER FACE [30] in face, Kitchen [5] and Oktoberfest [36] in food, HollywoodHeads [23] in head, LogoDet-3K [25] and OpenLogo [20] in logo, ChestX-Det10 [13] and DeepLesion [29] in medical imaging, CrowdHuman [19] and WiderPerson [32] in person, PIDray [24] and SIXray [14] in security, table-detection [18] in table, COCO-Text [22] in text in the wild, and Cityscapes [2], KITTI [4], LISA [15], and TT100K [35] in traffic, DUO [12] in underwater. Their statistics can be found in Table C.5. Sample images from these datasets can be found in Figure C.1.

Table C.6 compares *FT* and SOTA methods. *TFA-cos* is a variation of *TFA* where the classification head is replaced with the cosine similarity. Note that, while the comparison within tables is fair, the results are NOT directly comparable to the results in the main paper, as they are experimented in different

Table C.2: Per-dataset 1-shot performance of the effects of tuning different parameters, different architectures and pre-training datasets.

1-shot	Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Unfrozen	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Last FC layers (TFA [26])	7.5 ± 0.5	28.7 ± 5.0	55.5 ± 17.3	29.2 ± 2.5	6.7 ± 1.2	21.9 ± 3.3	12.3 ± 3.9	26.3 ± 2.1	2.6 ± 1.8	43.6 ± 5.7	23.4 ± 4.6	2.7 ± 0.3
Detection Head (FSCE-base [21])	7.8 ± 0.8	34.6 ± 6.3	62.3 ± 8.2	30.6 ± 2.6	14.1 ± 1.2	41.0 ± 4.0	24.1 ± 5.5	44.4 ± 4.8	4.5 ± 2.4	41.3 ± 5.5	30.5 ± 2.3	1.9 ± 0.1
Whole Network (Ours-FT)	8.4 ± 1.0	36.2 ± 7.7	56.1 ± 5.1	37.2 ± 3.9	14.2 ± 1.5	47.7 ± 6.7	25.0 ± 4.7	45.0 ± 4.2	6.6 ± 4.3	38.8 ± 3.6	**31.5** ± 2.0	**1.5** ± 0.3

(a) Fine-tuning different number of parameters with Faster R-CNN pre-trained on COCO.

1-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Faster R-CNN	COCO	8.4 ± 1.0	36.2 ± 7.7	56.1 ± 5.1	37.2 ± 3.9	14.2 ± 1.5	47.7 ± 6.7	25.0 ± 4.7	45.0 ± 4.2	6.6 ± 4.3	38.8 ± 3.6	31.5 ± 2.0	3.6 ± 0.2
Cascade R-CNN		7.2 ± 0.8	35.2 ± 6.4	56.3 ± 8.0	39.0 ± 3.6	13.0 ± 1.2	47.1 ± 6.4	27.6 ± 4.3	44.5 ± 4.2	6.5 ± 4.5	38.3 ± 5.6	31.5 ± 2.1	3.9 ± 0.4
CenterNet2		7.9 ± 0.8	35.6 ± 5.2	38.2 ± 20.6	33.0 ± 7.7	14.7 ± 2.2	45.2 ± 6.7	27.6 ± 4.3	43.8 ± 4.8	7.0 ± 4.1	38.4 ± 3.6	29.1 ± 5.2	4.0 ± 0.3
RetinaNet		5.5 ± 0.6	27.6 ± 7.3	50.9 ± 14.6	10.2 ± 1.7	8.7 ± 0.7	42.6 ± 5.8	24.3 ± 4.1	41.8 ± 2.8	6.3 ± 3.9	36.3 ± 2.3	25.4 ± 4.0	5.4 ± 0.6
Deformable-DETR		8.7 ± 1.1	44.0 ± 6.4	53.2 ± 12.1	23.4 ± 4.4	15.3 ± 1.3	47.8 ± 6.3	28.0 ± 4.9	47.8 ± 4.4	10.1 ± 4.6	41.9 ± 5.1	32.0 ± 2.9	2.5 ± 0.5
Cascade R-CNN-P67		9.6 ± 1.0	41.0 ± 4.9	65.5 ± 7.2	44.0 ± 2.5	16.2 ± 1.6	51.3 ± 5.9	28.3 ± 4.9	47.0 ± 3.9	8.8 ± 4.5	42.1 ± 4.5	**35.4** ± 1.8	**1.6** ± 0.3
Faster R-CNN	LVIS	8.3 ± 0.7	45.2 ± 4.4	58.7 ± 6.4	24.2 ± 3.9	20.7 ± 1.3	49.2 ± 5.2	25.6 ± 5.2	41.6 ± 3.2	8.6 ± 3.8	34.9 ± 3.5	31.7 ± 1.7	2.1 ± 0.2
CenterNet2		7.6 ± 0.7	41.5 ± 6.0	36.0 ± 12.7	20.0 ± 2.4	18.7 ± 1.4	50.4 ± 7.1	27.8 ± 5.0	38.7 ± 2.6	8.6 ± 4.2	32.3 ± 3.6	28.1 ± 3.3	2.7 ± 0.3
Cascade R-CNN-P67		9.2 ± 0.8	46.4 ± 6.1	61.2 ± 6.0	29.9 ± 2.9	23.1 ± 1.4	52.4 ± 6.9	31.0 ± 5.3	43.4 ± 2.7	9.2 ± 4.7	38.5 ± 4.8	**34.4** ± 2.0	**1.2** ± 0.2

(b) Performance of different architectures pre-trained on COCO and LVIS.

1-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Cascade R-CNN-P67	ImageNet	4.8 ± 0.5	23.5 ± 4.5	1.0 ± 0.8	2.8 ± 1.0	9.3 ± 1.1	43.3 ± 5.2	23.1 ± 3.8	14.8 ± 2.5	2.1 ± 1.3	10.7 ± 2.7	13.5 ± 1.6	6.0 ± 0.1
	COCO	9.6 ± 1.0	41.0 ± 4.9	65.5 ± 7.2	44.0 ± 2.5	16.2 ± 1.6	51.3 ± 5.9	28.3 ± 4.9	47.0 ± 3.9	8.8 ± 4.5	42.1 ± 4.5	**35.4** ± 1.8	2.9 ± 0.4
	FSODD	5.8 ± 0.5	48.0 ± 3.9	44.1 ± 11.1	12.5 ± 2.7	14.8 ± 1.2	50.1 ± 6.5	29.3 ± 4.2	28.6 ± 1.8	8.0 ± 3.6	25.5 ± 3.0	26.7 ± 2.9	4.2 ± 0.5
	LVIS	9.2 ± 0.8	46.4 ± 6.1	61.2 ± 6.0	29.9 ± 2.9	23.1 ± 1.4	52.4 ± 6.9	31.0 ± 5.3	43.4 ± 2.7	9.2 ± 4.7	38.5 ± 4.8	34.4 ± 2.0	3.0 ± 0.2
	Unified	9.7 ± 1.0	47.2 ± 6.5	45.8 ± 8.5	31.2 ± 5.1	18.4 ± 1.3	52.8 ± 6.1	31.2 ± 5.1	46.4 ± 3.0	10.4 ± 4.8	39.5 ± 3.6	33.3 ± 2.2	2.8 ± 0.3
	LVIS+	11.7 ± 1.0	57.4 ± 6.4	30.9 ± 15.9	37.3 ± 1.9	25.5 ± 0.9	50.0 ± 8.0	36.0 ± 3.8	45.1 ± 3.1	13.3 ± 5.2	39.5 ± 4.9	34.7 ± 4.2	**2.1** ± 0.5
CenterNet2	COCO	7.9 ± 0.8	35.6 ± 5.2	38.2 ± 20.6	33.0 ± 7.7	14.7 ± 2.2	45.2 ± 6.7	27.6 ± 4.3	43.8 ± 4.8	7.0 ± 4.1	38.4 ± 3.6	29.1 ± 5.2	3.1 ± 0.4
	LVIS	7.6 ± 0.7	41.5 ± 6.0	36.0 ± 12.7	20.0 ± 2.4	18.7 ± 1.4	50.4 ± 7.1	27.8 ± 5.0	38.7 ± 2.6	8.6 ± 4.2	32.3 ± 3.6	28.1 ± 3.3	3.3 ± 0.3
	LVIS+	10.6 ± 1.1	55.7 ± 5.4	38.4 ± 12.8	35.5 ± 2.9	25.1 ± 1.1	48.4 ± 6.7	36.7 ± 4.2	46.4 ± 3.3	13.9 ± 5.7	37.9 ± 5.3	34.9 ± 3.2	1.9 ± 0.3
	LVIS++	10.7 ± 1.0	59.4 ± 5.5	41.7 ± 13.4	38.2 ± 2.3	26.7 ± 1.0	46.2 ± 6.1	35.6 ± 4.2	47.0 ± 3.5	15.7 ± 5.8	37.1 ± 4.7	**35.8** ± 3.4	**1.7** ± 0.3

(c) Performance of Cascade R-CNN-P67 and CenterNet2 pre-trained on different datasets.

settings. Specifically, the architecture uses deformable convolution v1 while the one in the main paper uses v2, and is trained with a non-standard scheduler. We can observe that *FT* is a strong baseline, outperforming all other methods.

We then present the results between different architectures and pre-training datasets in Table C.7. Although the ablation study here is not as comprehensive as the main paper, we can still see that *Cascade R-CNN-P67* outperforms *Faster R-CNN*. The margin here is smaller mainly due to the less optimal learning rate scheduler we used for *Cascade R-CNN-P67* and possibly the lack of deformable convolution in this model. Once we use the same learning rate scheduler and backbone architecture with deformable convolution v2 [34] for both *Cascade R-CNN-P67* and *Faster R-CNN* as in the main paper, the performance gap for different shots actually increases. On the other hand, the comparison between all *Cascade R-CNN-P67* experiments is fair. We can see that over a larger range of domains, **Unified** provides better results than **COCO** by a significant margin. However, this performance gap could be due to the non-optimal training of **COCO**. These suggest that besides the size/quality of the pre-training datasets, how to train for downstream tasks optimally is also an important factor.

Table C.3: Per-dataset 3-shot performance of the effects of tuning different parameters, different architectures and pre-training datasets.

3-shot	Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Unfrozen	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Last FC layers (TFA [26])	9.4 ± 0.4	41.8 ± 3.0	70.3 ± 4.5	35.9 ± 2.3	7.7 ± 1.3	32.0 ± 6.9	13.3 ± 3.3	29.6 ± 1.0	5.4 ± 1.9	46.6 ± 4.2	29.2 ± 1.8	2.7 ± 0.2
Detection Head (FSCE-base [21])	11.4 ± 0.7	51.6 ± 4.8	70.0 ± 1.9	38.7 ± 1.7	18.4 ± 1.0	61.4 ± 5.4	38.4 ± 4.8	49.0 ± 3.2	10.7 ± 3.5	44.6 ± 4.4	39.4 ± 1.6	1.9 ± 0.2
Whole Network (Ours-FT)	12.0 ± 0.8	52.6 ± 4.6	62.9 ± 5.2	45.5 ± 3.1	19.3 ± 1.0	70.1 ± 5.9	41.7 ± 4.5	48.8 ± 2.7	15.5 ± 6.2	43.2 ± 3.6	**41.1** ± 1.8	**1.5** ± 0.2

(a) Fine-tuning different number of parameters with Faster R-CNN pre-trained on COCO.

3-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Faster R-CNN	COCO	12.0 ± 0.8	52.6 ± 4.6	62.9 ± 5.2	45.5 ± 3.1	19.3 ± 1.0	70.1 ± 5.9	41.7 ± 4.5	48.8 ± 2.7	15.5 ± 6.2	43.2 ± 3.6	41.1 ± 1.8	3.5 ± 0.3
Cascade R-CNN		11.0 ± 0.7	52.3 ± 2.9	67.7 ± 3.2	45.9 ± 2.5	18.3 ± 0.9	69.5 ± 5.1	42.3 ± 4.1	48.9 ± 2.6	14.4 ± 5.9	41.8 ± 3.1	41.2 ± 1.5	3.8 ± 0.3
CenterNet2		11.6 ± 0.7	50.9 ± 6.3	60.6 ± 4.9	44.0 ± 6.8	19.7 ± 2.4	68.2 ± 5.9	42.7 ± 2.4	48.6 ± 3.9	15.5 ± 5.7	40.5 ± 4.4	40.2 ± 1.9	3.8 ± 0.4
RetinaNet		8.2 ± 0.5	45.7 ± 3.2	59.0 ± 7.2	19.2 ± 1.8	14.5 ± 0.6	66.5 ± 6.4	39.1 ± 4.7	45.5 ± 2.0	12.0 ± 4.9	37.8 ± 2.9	34.8 ± 2.2	5.6 ± 0.6
Deformable-DETR		12.7 ± 0.7	61.1 ± 4.3	64.8 ± 3.2	35.9 ± 2.7	19.9 ± 1.2	67.9 ± 4.6	42.5 ± 4.6	53.1 ± 3.1	21.3 ± 6.7	43.4 ± 3.5	42.3 ± 1.6	2.7 ± 0.5
Cascade R-CNN-P67		13.7 ± 0.9	55.3 ± 3.0	72.8 ± 2.5	52.1 ± 2.4	21.9 ± 1.0	71.2 ± 5.4	46.6 ± 4.4	51.2 ± 2.4	17.9 ± 5.6	44.4 ± 2.8	**44.7** ± 1.6	**1.7** ± 0.5
Faster R-CNN	LVIS	11.9 ± 0.6	59.2 ± 5.2	69.4 ± 3.1	33.8 ± 3.5	25.8 ± 0.9	71.1 ± 4.4	41.5 ± 4.2	45.9 ± 2.5	18.3 ± 5.1	38.6 ± 3.6	41.6 ± 1.5	2.1 ± 0.3
CenterNet2		11.2 ± 0.6	56.9 ± 3.9	59.0 ± 5.5	27.9 ± 3.9	23.1 ± 0.6	70.7 ± 5.0	45.3 ± 4.0	43.8 ± 2.3	16.5 ± 5.3	35.8 ± 4.2	39.0 ± 1.7	2.7 ± 0.3
Cascade R-CNN-P67		13.1 ± 0.7	59.9 ± 5.3	71.1 ± 3.1	39.6 ± 3.7	28.1 ± 1.0	71.9 ± 5.5	47.7 ± 2.8	47.3 ± 2.4	19.0 ± 5.4	42.8 ± 4.1	**44.0** ± 1.6	**1.2** ± 0.3

(b) Performance of different architectures pre-trained on COCO and LVIS.

3-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Cascade R-CNN-P67	ImageNet	7.9 ± 0.5	43.1 ± 3.3	4.1 ± 2.2	7.3 ± 1.8	16.7 ± 0.7	65.8 ± 5.5	37.6 ± 4.7	25.6 ± 3.3	6.5 ± 3.4	17.9 ± 2.0	23.2 ± 1.5	6.0 ± 0.1
	COCO	13.7 ± 0.9	55.3 ± 3.0	72.8 ± 2.5	52.1 ± 2.4	21.9 ± 1.0	71.2 ± 5.4	46.6 ± 4.4	51.2 ± 2.4	17.9 ± 5.6	44.4 ± 2.8	44.7 ± 1.6	2.9 ± 0.4
	FSODD	9.0 ± 0.8	61.0 ± 3.4	62.3 ± 4.6	19.6 ± 2.5	19.5 ± 0.8	71.8 ± 5.0	46.9 ± 4.4	35.0 ± 2.5	16.1 ± 5.0	28.3 ± 3.2	36.9 ± 1.5	4.3 ± 0.4
	LVIS	13.1 ± 0.7	59.9 ± 5.3	71.1 ± 3.1	39.6 ± 3.7	28.1 ± 1.0	71.9 ± 5.5	47.7 ± 2.8	47.3 ± 2.4	19.0 ± 5.4	42.8 ± 4.1	44.0 ± 1.6	3.2 ± 0.5
	Unified	14.0 ± 0.9	62.5 ± 3.2	63.7 ± 4.2	41.9 ± 3.3	24.2 ± 0.8	73.7 ± 5.1	50.1 ± 4.5	50.3 ± 2.2	19.7 ± 4.5	43.1 ± 3.0	44.3 ± 1.4	2.7 ± 0.4
	LVIS+	16.3 ± 0.9	70.7 ± 3.5	55.0 ± 6.3	46.8 ± 2.3	29.8 ± 0.7	72.1 ± 3.5	52.2 ± 3.6	50.1 ± 2.6	26.8 ± 4.9	46.0 ± 3.4	**46.6** ± 1.6	**1.9** ± 0.5
CenterNet2	COCO	7.9 ± 0.8	35.6 ± 5.2	38.2 ± 20.6	33.0 ± 7.7	14.7 ± 2.2	45.2 ± 6.7	27.6 ± 4.3	43.8 ± 4.8	7.0 ± 4.1	38.4 ± 3.6	29.1 ± 5.2	3.1 ± 0.4
	LVIS	7.6 ± 0.7	41.5 ± 6.0	36.0 ± 12.7	20.0 ± 2.4	18.7 ± 1.4	50.4 ± 7.1	27.8 ± 5.0	38.7 ± 2.6	8.6 ± 4.2	32.3 ± 3.6	28.1 ± 3.3	3.3 ± 0.3
	LVIS+	10.6 ± 1.1	55.7 ± 5.4	38.4 ± 12.8	35.5 ± 2.9	25.1 ± 1.1	48.4 ± 6.7	36.7 ± 4.2	46.4 ± 3.3	13.9 ± 5.7	37.9 ± 5.3	34.9 ± 3.2	1.9 ± 0.3
	LVIS++	10.7 ± 1.0	59.4 ± 5.5	41.7 ± 13.4	38.2 ± 2.3	26.7 ± 1.0	46.2 ± 6.1	35.6 ± 4.2	47.0 ± 3.5	15.7 ± 5.8	37.1 ± 4.7	**35.8** ± 3.4	**1.7** ± 0.3

(c) Performance of Cascade R-CNN-P67 and CenterNet2 pre-trained on different datasets.

Table C.4: Per-dataset 10-shot performance of the effects of tuning different parameters, different architectures and pre-training datasets.

10-shot	Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Unfrozen	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Last FC layers (TFA [26])	10.8 ± 0.5	55.9 ± 1.8	73.8 ± 2.9	44.5 ± 1.0	8.1 ± 1.2	48.0 ± 2.7	15.3 ± 4.4	31.4 ± 0.7	11.0 ± 1.2	53.2 ± 2.4	35.2 ± 1.2	2.7 ± 0.2
Detection Head (FSCE-base [21])	15.5 ± 0.6	69.7 ± 2.8	72.2 ± 1.8	50.2 ± 1.1	23.2 ± 1.2	83.0 ± 3.3	55.7 ± 4.3	54.4 ± 1.8	23.8 ± 1.9	54.0 ± 2.5	50.2 ± 1.1	1.8 ± 0.2
Whole Network (Ours-FT)	17.5 ± 0.7	71.4 ± 1.9	65.3 ± 6.9	57.4 ± 1.1	24.8 ± 1.0	90.6 ± 1.9	59.2 ± 4.4	53.3 ± 1.7	36.1 ± 3.0	50.6 ± 3.1	**52.6** ± 1.8	**1.5** ± 0.2

(a) Fine-tuning different number of parameters with Faster R-CNN pre-trained on COCO.

10-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Faster R-CNN	COCO	17.5 ± 0.7	71.4 ± 1.9	65.3 ± 6.9	57.4 ± 1.1	24.8 ± 1.0	90.6 ± 1.9	59.2 ± 4.4	53.3 ± 1.7	36.1 ± 3.0	50.6 ± 3.1	52.6 ± 1.8	3.8 ± 0.4
Cascade R-CNN		16.6 ± 0.7	71.6 ± 2.2	69.7 ± 3.5	58.5 ± 1.5	23.8 ± 0.9	90.0 ± 2.2	58.8 ± 4.6	53.5 ± 1.6	33.7 ± 2.7	50.8 ± 2.9	52.7 ± 1.2	3.9 ± 0.4
CenterNet2		16.4 ± 0.7	71.3 ± 2.0	65.2 ± 5.1	57.6 ± 7.0	25.4 ± 1.7	89.9 ± 2.7	62.0 ± 5.2	54.1 ± 2.8	33.1 ± 2.1	50.0 ± 5.0	52.5 ± 1.9	3.8 ± 0.4
RetinaNet		12.6 ± 0.6	69.2 ± 2.5	64.3 ± 4.5	40.9 ± 1.6	21.5 ± 0.5	90.2 ± 1.7	60.4 ± 3.5	50.0 ± 1.5	32.7 ± 1.4	46.3 ± 2.4	48.8 ± 1.2	5.2 ± 0.5
Deformable-DETR		18.3 ± 0.9	78.6 ± 2.0	70.3 ± 3.4	54.5 ± 1.5	24.2 ± 1.1	86.7 ± 2.2	61.1 ± 3.9	59.6 ± 1.7	39.8 ± 3.3	54.3 ± 2.8	54.7 ± 1.0	2.7 ± 0.5
Cascade R-CNN-P67		19.1 ± 0.8	73.3 ± 1.6	75.4 ± 2.5	63.4 ± 1.0	27.4 ± 1.2	91.2 ± 2.1	65.7 ± 4.7	56.2 ± 1.6	38.3 ± 2.4	53.7 ± 2.9	**56.4** ± 1.1	**1.7** ± 0.4
Faster R-CNN	LVIS	17.7 ± 0.7	74.8 ± 1.9	71.9 ± 2.5	51.1 ± 1.0	31.0 ± 0.8	90.5 ± 1.6	63.1 ± 4.2	51.1 ± 1.7	36.6 ± 2.0	47.7 ± 2.6	53.6 ± 1.0	2.1 ± 0.3
CenterNet2		16.6 ± 0.8	74.6 ± 2.0	68.1 ± 3.1	43.9 ± 1.3	28.3 ± 0.9	90.8 ± 2.0	64.1 ± 4.1	50.0 ± 1.6	34.1 ± 1.7	45.7 ± 2.9	51.6 ± 1.0	2.7 ± 0.3
Cascade R-CNN-P67		18.6 ± 0.7	76.1 ± 1.4	72.8 ± 2.7	55.7 ± 1.1	33.1 ± 0.6	91.4 ± 2.0	66.9 ± 4.0	52.5 ± 1.5	37.7 ± 2.6	51.1 ± 2.8	**55.6** ± 1.0	**1.2** ± 0.3

(b) Performance of different architectures pre-trained on COCO and LVIS.

10-shot		Aerial	Agriculture	Animal	Cartoon	Fashion	Food	Logo	Person	Security	Traffic	Mean	Rank
Architecture	Pre-training	VisDrone	DeepFruits	iWildCam	Clipart	iMaterialist	Oktoberfest	LogoDet-3K	CrowdHuman	SIXray	KITTI		
Cascade R-CNN-P67	ImageNet	13.5 ± 0.5	66.5 ± 2.5	14.6 ± 4.4	25.8 ± 1.6	21.9 ± 0.6	86.2 ± 3.1	54.7 ± 3.7	38.9 ± 1.3	22.1 ± 2.3	32.6 ± 3.0	37.7 ± 1.2	5.9 ± 0.2
	COCO	19.1 ± 0.8	73.3 ± 1.6	75.4 ± 2.5	63.4 ± 1.0	27.4 ± 1.2	91.2 ± 2.1	65.7 ± 4.7	56.2 ± 1.6	38.3 ± 2.4	53.7 ± 2.9	56.4 ± 1.1	2.9 ± 0.4
	FSODD	13.7 ± 0.6	74.9 ± 2.1	67.8 ± 3.7	35.9 ± 1.9	24.6 ± 0.9	90.8 ± 1.5	66.5 ± 3.6	42.9 ± 1.8	34.3 ± 1.6	39.5 ± 2.6	49.1 ± 1.0	4.5 ± 0.4
	LVIS	18.6 ± 0.7	76.1 ± 1.4	72.8 ± 2.7	55.7 ± 1.1	33.1 ± 0.6	91.4 ± 2.0	66.9 ± 4.0	52.5 ± 1.5	37.7 ± 2.6	51.1 ± 2.8	55.6 ± 1.0	3.2 ± 0.5
	Unified	19.7 ± 0.7	76.8 ± 1.1	69.9 ± 3.4	58.6 ± 1.5	29.5 ± 1.1	92.8 ± 1.0	69.5 ± 4.0	55.6 ± 1.5	39.9 ± 2.9	53.6 ± 2.9	56.6 ± 1.1	2.4 ± 0.2
	LVIS+	21.4 ± 0.7	84.4 ± 1.5	67.1 ± 3.5	60.4 ± 0.8	34.4 ± 0.5	89.9 ± 1.8	70.7 ± 3.2	55.1 ± 1.4	50.2 ± 2.6	55.9 ± 2.9	**59.0** ± 1.0	**2.0** ± 0.4
CenterNet2	COCO	7.9 ± 0.8	35.6 ± 5.2	38.2 ± 20.6	33.0 ± 7.7	14.7 ± 2.2	45.2 ± 6.7	27.6 ± 4.3	43.8 ± 4.8	7.0 ± 4.1	38.4 ± 3.6	29.1 ± 5.2	3.1 ± 0.4
	LVIS	7.6 ± 0.7	41.5 ± 6.0	36.0 ± 12.7	20.0 ± 2.4	18.7 ± 1.4	50.4 ± 7.1	27.8 ± 5.0	38.7 ± 2.6	8.6 ± 4.2	32.3 ± 3.6	28.1 ± 3.3	3.3 ± 0.3
	LVIS+	10.6 ± 1.1	55.7 ± 5.4	38.4 ± 12.8	35.5 ± 2.9	25.1 ± 1.1	48.4 ± 6.7	36.7 ± 4.2	46.4 ± 3.3	13.9 ± 5.7	37.9 ± 5.3	34.9 ± 3.2	1.9 ± 0.3
	LVIS++	10.7 ± 1.0	59.4 ± 5.5	41.7 ± 13.4	38.2 ± 2.3	26.7 ± 1.0	46.2 ± 6.1	35.6 ± 4.2	47.0 ± 3.5	15.7 ± 5.8	37.1 ± 4.7	**35.8** ± 3.4	**1.7** ± 0.3

(c) Performance of Cascade R-CNN-P67 and CenterNet2 pre-trained on different datasets.

Table C.5: Statistics of 32 datasets in the extended benchmark. The 10 datasets used in MoFSOD are shown in **bold**.

Domain	Dataset	# classes	# train images	# train anno.	# test images	# test anno.
Aerial	CARPK	1	989	42275	459	47501
	DOTA	15	8949	116515	8949	116515
	VisDrone	10	7019	381965	1610	75103
Agriculture	**DeepFruits**	7	457	2553	114	590
	MinneApple	1	403	19373	267	8811
Animal	ENA24	22	7031	7811	1758	1963
	iWildCam	1	21065	31591	5313	7901
Cartoon	**Clipart**	20	500	1640	500	1527
	Comic	6	1000	3215	1000	3176
	Watercolor	6	1000	1662	1000	1655
Dense Product	SKU110K	1	8804	1298968	2935	431420
Face	WIDER FACE	1	12880	159423	3222	39698
Fashion	DeepFashion2	13	191961	312187	32153	52491
	iMaterialist	46	45623	333402	1158	8782
Food	Kitchen	11	4711	24730	2016	13430
	Oktoberfest	15	1110	2697	85	236
Head	HollywoodHeads	1	10834	17754	3984	7080
Logo	**LogoDet-3K**	2993	126891	155286	31727	38981
	OpenLogo	352	18752	35264	8331	15945
Medical	ChestX-Det10	10	2320	6864	459	1477
	DeepLesion	1	27289	28871	4831	5122
Person	**CrowdHuman**	2	15000	705967	4370	206231
	WiderPerson	1	8000	245053	1000	28424
Security	PIDray	12	29454	39709	9482	9483
	SIXray	5	7496	15439	1310	2054
Table	table-detection	1	212	244	191	276
Text	COCO-Text	2	19039	163477	4446	37651
Traffic	Cityscapes	8	2965	50348	492	9793
	KITTI	4	5481	38077	7481	52458
	LISA	5	7937	9246	1987	2283
	TT100K	151	6105	16528	3071	8175
Underwater	DUO	4	6617	63999	1100	10518

Table C.6: Performance on the proposed benchmark in AP50 (top) and the average rank (bottom) of different methods on the extended benchmark with 32 datasets. Note that the pre-trained model used for this table is different from the main paper, such that the comparison is fair within these tables only.

Method	1-shot	3-shot	5-shot	10-shot	Mean
TFA [26]	20.6 ± 3.7	25.6 ± 2.0	27.9 ± 1.7	30.9 ± 1.3	26.3 ± 2.4
TFA-cos [26]	20.8 ± 3.6	25.6 ± 2.0	27.7 ± 1.7	30.5 ± 1.3	26.1 ± 2.4
FSCE-base [21]	25.3 ± 4.2	33.6 ± 3.9	37.9 ± 2.0	43.4 ± 1.6	35.1 ± 3.3
FSCE-con [21]	25.8 ± 4.4	34.1 ± 3.7	38.1 ± 1.8	43.3 ± 1.5	35.3 ± 3.3
DeFRCN [16]	25.4 ± 4.0	32.9 ± 3.1	36.7 ± 1.9	41.2 ± 1.9	34.1 ± 3.0
FT	**26.2** ± 3.3	**35.2** ± 3.5	**39.6** ± 2.3	**45.8** ± 2.1	**36.7** ± 2.9

Method	1-shot	3-shot	5-shot	10-shot	Mean
TFA [26]	4.7 ± 0.5	4.8 ± 0.4	4.7 ± 0.3	4.7 ± 0.4	4.7 ± 0.4
TFA-cos [26]	4.3 ± 0.4	4.5 ± 0.4	4.7 ± 0.4	4.8 ± 0.4	4.6 ± 0.4
FSCE-base [21]	3.3 ± 0.4	3.0 ± 0.4	2.9 ± 0.3	2.8 ± 0.3	3.0 ± 0.4
FSCE-con [21]	2.8 ± 0.4	2.7 ± 0.3	2.7 ± 0.3	2.7 ± 0.3	2.7 ± 0.4
DeFRCN [16]	3.3 ± 0.5	3.5 ± 0.5	3.5 ± 0.4	3.7 ± 0.3	3.5 ± 0.5
FT	**2.7** ± 0.4	**2.5** ± 0.4	**2.4** ± 0.5	**2.2** ± 0.5	**2.5** ± 0.5

Table C.7: Performance of FT on the proposed benchmark with different model architectures and pre-training datasets in AP50 (top) and the average rank (bottom) on the 32 datasets extended benchmark. Note that the pre-trained model used for this table is different from the main paper, such that the comparison is fair within these tables only.

Arch	Pre-train	1-shot	3-shot	5-shot	10-shot	Mean
Faster R-CNN	COCO	26.2 ± 3.3	35.2 ± 3.5	39.6 ± 2.3	45.8 ± 2.1	36.7 ± 2.9
Cascade R-CNN-P67	COCO	26.4 ± 3.3	35.7 ± 2.6	40.3 ± 2.0	46.7 ± 1.6	37.3 ± 2.5
	FSODD	23.3 ± 3.2	32.4 ± 2.5	36.8 ± 1.8	42.9 ± 1.7	33.8 ± 2.5
	Unified	**29.2** ± 2.9	**38.7** ± 2.6	**43.4** ± 1.6	**49.7** ± 1.9	**40.3** ± 2.4

Arch	Pre-train	1-shot	3-shot	5-shot	10-shot	Mean
Faster R-CNN	COCO	2.8 ± 0.4	2.9 ± 0.4	3.0 ± 0.3	3.0 ± 0.3	2.9 ± 0.3
Cascade R-CNN-P67	COCO	2.5 ± 0.3	2.5 ± 0.2	2.5 ± 0.3	2.4 ± 0.3	2.5 ± 0.3
	FSODD	3.0 ± 0.3	3.1 ± 0.3	3.3 ± 0.3	3.4 ± 0.3	3.2 ± 0.3
	Unified	**1.6** ± 0.4	**1.4** ± 0.3	**1.3** ± 0.3	**1.2** ± 0.3	**1.4** ± 0.3

References

1. Beery, S., Agarwal, A., Cole, E., Birodkar, V.: The iwildcam 2021 competition dataset. arXiv preprint arXiv:2105.03494 (2021) 3
2. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: CVPR (2016) 3
3. Ge, Y., Zhang, R., Wang, X., Tang, X., Luo, P.: DeepFashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images. In: CVPR (2019) 3
4. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012) 3
5. Georgakis, G., Reza, M.A., Mousavian, A., Le, P.H., Košecká, J.: Multiview rgb-d dataset for object instance detection. In: 3DV (2016) 3
6. Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T.: Precise detection in densely packed scenes. In: CVPR (2019) 3
7. Guo, S., Huang, W., Zhang, X., Srikhanta, P., Cui, Y., Li, Y., R.Scott, M., Adam, H., Belongie, S.: The imaterialist fashion attribute dataset. arXiv preprint arXiv:1906.05750 (2019) 3
8. Häni, N., Roy, P., Isler, V.: MinneApple: a benchmark dataset for apple detection and segmentation. IEEE Robotics and Automation Letters **5**(2), 852–858 (2020) 3
9. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-based object counting by spatially regularized regional proposal network. In: ICCV (2017) 3
10. Inoue, N., Furuta, R., Yamasaki, T., Aizawa, K.: Cross-domain weakly-supervised object detection through progressive domain adaptation. In: CVPR (2018) 3
11. Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., Darrell, T.: Few-shot object detection via feature reweighting. In: ICCV (2019) 1
12. Liu, C., Li, H., Wang, S., Zhu, M., Wang, D., Fan, X., Wang, Z.: A dataset and benchmark of underwater object detection for robot picking. In: 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (2021) 3
13. Liu, J., Lian, J., Yu, Y.: Chestx-det10: Chest x-ray dataset on detection of thoracic abnormalities. arXiv preprint arXiv:2006.10550 (2020) 3
14. Miao, C., Xie, L., Wan, F., Su, C., Liu, H., Jiao, J., Ye, Q.: Sixray: A large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. In: CVPR (2019) 3
15. Mogelmose, A., Trivedi, M.M., Moeslund, T.B.: Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems **13**(4), 1484–1497 (2012) 3
16. Qiao, L., Zhao, Y., Li, Z., Qiu, X., Wu, J., Zhang, C.: DeFRCN: Decoupled faster r-cnn for few-shot object detection. In: ICCV (2021) 8
17. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., McCool, C.: DeepFruits: A fruit detection system using deep neural networks. sensors **16**(8), 1222 (2016) 3
18. sgrpanchal31: table-detection-dataset. https://github.com/sgrpanchal31/table-detection-dataset (2018) 3
19. Shao, S., Zhao, Z., Li, B., Xiao, T., Yu, G., Zhang, X., Sun, J.: CrowdHuman: A benchmark for detecting human in a crowd. arXiv preprint arXiv:1805.00123 (2018) 3
20. Su, H., Zhu, X., Gong, S.: Open logo detection challenge. In: BMVC (2018) 3
21. Sun, B., Li, B., Cai, S., Yuan, Y., Zhang, C.: Fsce: Few-shot object detection via contrastive proposal encoding. In: CVPR (2021) 1, 4, 5, 6, 8

22. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016) 3
23. Vu, T.H., Osokin, A., Laptev, I.: Context-aware cnns for person head detection. In: ICCV (2015) 3
24. Wang, B., Zhang, L., Wen, L., Liu, X., Wu, Y.: Towards real-world prohibited item detection: A large-scale x-ray benchmark. In: CVPR (2021) 3
25. Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., Jiang, S.: LogoDet-3K: A large-scale image dataset for logo detection. arXiv preprint arXiv:2008.05359 (2020) 3
26. Wang, X., Huang, T.E., Darrell, T., Gonzalez, J.E., Yu, F.: Frustratingly simple few-shot object detection. In: ICML (2020) 1, 4, 5, 6, 8
27. Wang, X., Cai, Z., Gao, D., Vasconcelos, N.: Towards universal object detection by domain attention. In: CVPR (2019) 3
28. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: DOTA: A large-scale dataset for object detection in aerial images. In: CVPR (2018) 3
29. Yan, K., Wang, X., Lu, L., Zhang, L., Harrison, A.P., Bagheri, M., Summers, R.M.: Deep lesion graphs in the wild: relationship learning and organization of significant radiology image findings in a diverse large-scale lesion database. In: CVPR (2018) 3
30. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: CVPR (2016) 3
31. Yousif, H., Kays, R., He, Z.: Dynamic programming selection of object proposals for sequence-level animal species classification in the wild. IEEE Transactions on Circuits and Systems for Video Technology (2019) 3
32. Zhang, S., Xie, Y., Wan, J., Xia, H., Li, S.Z., Guo, G.: WiderPerson: A diverse dataset for dense pedestrian detection in the wild. IEEE Transactions on Multimedia **22**(2), 380–393 (2019) 3
33. Zhu, P., Wen, L., Du, D., Bian, X., Fan, H., Hu, Q., Ling, H.: Detection and tracking meet drones challenge. IEEE TPAMI (2021). https://doi.org/10.1109/TPAMI.2021.3119563 3
34. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets V2: more deformable, better results. In: CVPR (2019) 4
35. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: CVPR (2016) 3
36. Ziller, A., Hansjakob, J., Rusinov, V., Zügner, D., Vogel, P., Günnemann, S.: Oktoberfest food dataset. arXiv preprint arXiv:1912.05007 (2019) 3