Supplementary Material: Meta-tuning Loss Functions and Data Augmentation for Few-shot Object Detection

Berkan Demirel[1,2] Orhun Buğra Baran[1] Ramazan Gokberk Cinbis[1]
[1]Middle East Technical University [2]HAVELSAN Inc.
bdemirel@havelsan.com.tr bugra@ceng.metu.edu.tr gcinbis@ceng.metu.edu.tr

A. Proxy task class splits

We use proxy tasks to apply the meta-tuning ideas, so we generate sub-splits in the base classes. In this context, we select some base classes to mimic novel classes to conduct the proxy task. We summarize the list of proxy Pascal VOC classes on Table 1. The list of selected proxy novel classes for the MS-COCO dataset is as follows: {*"skis", "tennis racket", "scissors", "truck", "baseball bat", "handbag", "carrot", "mouse", "parking meter", "apple", "knife", "microwave", ""refrigerator", "cake", "zebra"*}.

B. Algorithm

We summarize the main meta-tuning procedure in Algorithm 1. We can divide this algorithm into three parts: (i) model initialization and parameter sampling, (ii) instance sampling and mAP calculation, (iii) mAP normalization and RL steps.

1) Model initialization and parameter sampling. This algorithm firstly initializes the base proxy detection model weights for the proxy task and sample ρ value from normal distributions. The base proxy detection model represents the object detection model trained using the $D_{\text{p-pretrain}}$ dataset.

2) Instance sampling and mAP calculation. The proposed algorithm samples new instances from the proxy fine-tuning dataset $D_{\text{p-support}}$, and calculates the mean average precision scores on proxy validation dataset $D_{\text{p-query}}$ after a certain number of iterations. The algorithm repeats this process for N times.

3) mAP normalization and RL steps. The proposed algorithm normalizes the mAP scores, selects the maximum score as the reward value among the normalized APs, and applies a single REINFORCE step.

Algorithm 1 Meta-tuning Loss Function Learning

Input: Pre-trained model m_{init}, proxy fine-tuning dataset $D_{\text{p-support}}$, proxy validation dataset $D_{\text{p-query}}$, number of rho trials N, maximum iteration number M

```
iteration_index = 1
repeat
    Initialize m_init and sample new ρ
    for rho_index = 1 to N do
        Sample new fine-tuning images from D_p-support
        Take m_init, run all iter. using current samples
        Calculate mAP[rho_index] on D_p-query
    end for
    Normalize mAP scores
    Get max normalized AP as a reward
    Make a single REINFORCE step
    iteration_index += 1
until iteration_index = M
```

C. Additional Experimental Results

In this section, we share detailed experimental comparison results for Pascal VOC and MS COCO datasets.

Comparison to fine-tuning based FSOD and G-FSOD methods on Pascal VOC. We first present the detailed Pascal VOC comparisons for each split and shot with only novel classes in Table 2, and the detailed comparisons with all classes in Table 3. The experimental results show that the meta-tuning approach significantly improves the strong fine-tuning based few-shot detection baselines on the Pascal VOC benchmark. We provide complementary visual results of the MPSR+Meta-ScaledDynamic+Aug method using the Pascal VOC split-3/10-shot setting in Figure 1. We also present examples from the visual results of the DeFRCN+Meta-ScaledDynamic+Aug method using the Pascal VOC split-2/10-shot setting in Figure 2.

Comparisons to meta-learning based FSOD and G-FSOD on Pascal VOC. We present the detailed Pascal VOC comparisons with meta-learning based methods in Table 4 and Table 5 for novel-only and all-classes settings, respectively. Since the most of the meta-learning methods do not share G-FSOD results, we are able to compare against a more

Proxy-base classes ($C_{\textbf{p-base}}$)			**Proxy-novel classes** ($C_{\textbf{p-novel}}$)		
Split-1	Split-2	Split-3	Split-1	Split-2	Split-3
aeroplane	bicycle	aeroplane	person	motorbike	horse
bicycle	bird	bicycle	pottedplant	person	person
boat	boat	bird	sheep	sheep	pottedplant
bottle	bus	bottle	train	train	train
car	car	bus	tvmonitor	tvmonitor	tvmonitor
cat	cat	car			
chair	chair	chair			
diningtable	diningtable	cow			
dog	dog	diningtable			
horse	pottedplant	dog			

Table 1. Proxy task class splits for Pascal VOC.

Method/Shot	Split 1					Split 2					Split 3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FRCN [20] (ICCV'19)	15.2	20.3	29.0	25.5	28.7	13.4	20.6	28.6	32.4	38.8	19.6	20.8	28.7	42.2	42.1
TFA-fc [15] (ICML'20)	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA-cos [15] (ICML'20)	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
MPSR [17] (ECCV'20)	37.2	43.6	50.9	53.7	60.2	24.8	28.1	38.0	39.8	45.9	37.3	40.0	43.9	47.8	50.1
Ret. R-CNN [4] (CVPR'21)	42.4	45.8	45.9	53.7	56.1	21.7	27.8	35.2	37.0	40.3	30.2	37.6	43.0	49.7	50.1
TFA+H [23] (CVPR'21)	45.1	44.0	44.7	55.0	55.9	23.2	27.5	35.1	34.9	39.0	30.5	35.1	41.4	49.0	49.3
FSCE [14] (CVPR'21)	37.6	44.7	46.9	52.2	60.3	24.5	30.1	38.2	40.4	45.9	25.4	34.2	42.3	48.7	50.3
FADI [1] (NeurIPS'21)	50.3	54.8	54.2	59.3	63.2	30.6	35.0	40.3	42.8	48.0	45.7	49.7	49.1	55.0	59.6
LVC [9] (CVPR'22)	36.0	40.1	48.6	57.0	59.9	22.3	22.8	39.2	44.2	47.8	34.3	43.4	42.9	52.0	54.5
LVC-PL [9] (CVPR'22)	**54.5**	53.2	58.8	63.2	65.7	**32.8**	29.2	**50.7**	49.8	50.6	48.4	52.7	55.0	59.6	59.6
DeFRCN [13] (CVPR'21)	53.7	**59.5**	**61.2**	**65.7**	**66.6**	32.3	**42.0**	49.5	**52.4**	**53.4**	**53.6**	**56.2**	**56.9**	**61.9**	**62.3**
MPSR+Meta-Static	36.7	47.0	52.1	53.8	60.8	25.3	31.6	38.4	40.8	46.9	38.3	39.7	44.8	47.2	50.1
MPSR+Meta-Dynamic	40.4	47.5	51.9	54.9	60.5	25.6	31.7	38.5	40.6	46.7	37.6	40.2	44.7	49.1	50.3
MPSR+Meta-ScaledDynamic	41.5	47.9	52.7	55.4	60.9	25.7	32.2	38.9	40.8	46.8	38.5	40.9	45.9	49.0	51.0
MPSR+Aug	39.5	47.1	53.2	54.9	59.5	26.2	31.0	39.7	41.8	47.8	38.0	37.8	45.2	48.4	50.9
MPSR+Meta-Static+Aug	40.9	47.6	53.6	54.7	60.2	26.5	31.6	38.9	42.2	47.3	38.7	38.1	45.8	48.2	50.8
MPSR+Meta-Dynamic+Aug	41.0	47.5	53.8	55.2	60.2	26.4	32.2	39.8	42.7	48.5	38.9	39.1	46.0	48.8	51.3
MPSR+Meta-ScaledDynamic+Aug	41.8	48.7	54.2	55.7	61.1	26.5	32.7	40.0	42.5	48.7	39.0	40.4	46.2	49.6	51.2
DeFRCN+Meta-ScaledDynamic+Aug	**58.4**	**62.4**	**63.2**	**67.6**	**67.7**	**34.0**	**43.1**	**51.0**	**53.6**	**54.0**	**55.1**	**56.6**	**57.3**	**62.6**	**63.7**

Table 2. Comparison to fine-tuning based FSOD methods on the Pascal VOC dataset, with only novel classes. The best and the second-best results are marked with red and blue. MPSR+Meta-Static, MPSR+Meta-Dynamic, and MPSR+Meta-ScaledDynamic represent meta-tuning results.

limited number of meta-learning methods than FSOD. The experimental results (Table 4) show that our DeFRCN+Meta-ScaledDynamic+Aug method obtains the best results in all of the FSOD cases, except for the Split-2/1-shot setting. In the G-FSOD experiments (Table 5), it is observed that the proposed meta-tuning approach obtains the state-of-the-art results with a clear margin against existing meta-learning based methods.

Comparisons to meta-learning based FSOD and G-FSOD on MS-COCO. We compare our results with meta-learning based methods on the MS-COCO dataset and share the obtained results in Table 6. In this table, we are able to report a rather limited number of meta-learning methods to compare the G-FSOD results since most meta-learning based methods do not share G-FSOD results on the MS-COCO dataset. In FSOD experiments, we also observe that our DeFRCN+Meta-ScaledDynamic+Aug method obtains higher results than several recently published meta-learning based methods. We additionally observe major improvements in terms of HM scores in the G-FSOD setting, similar to the improvements obtained on the Pascal VOC dataset.

D. Implementation and runtime

We run our MPSR and DeFRCN experiments on a server with 4 Nvidia Tesla V100 32GB GPUs. The base MPSR model training to be used during fine-tuning takes 0.25 days for Pascal VOC and 0.45 days for MS COCO datasets. Since the base models used for the proxy tasks contain fewer

Method/Shot	Split-1					Split-2					Split-3				
	1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
FRCN [20] (ICCV'19)	24.9	31.4	40.3	37.6	41.0	22.1	31.3	39.1	43.0	47.5	30.8	32.3	40.5	52.2	51.7
TFA-fc [15] (ICML'20)	50.4	42.6	56.2	65.4	66.1	29.7	42.4	47.0	49.0	52.1	41.3	47.4	55.6	60.6	61.6
TFA-cos [15] (ICML'20)	53.1	49.5	57.1	65.4	65.3	36.3	40.0	47.6	48.6	52.2	44.5	48.5	55.9	61.2	61.4
MPSR [17] (ECCV'20)	45.8	52.5	59.3	61.8	65.5	36.0	39.7	49.8	51.7	56.9	47.6	49.9	54.5	58.1	60.0
FSCE [14] (CVPR'21)	50.7	56.5	58.1	61.6	66.1	36.5	42.4	49.8	51.5	55.8	38.2	47.4	54.6	59.9	61.1
Ret. R-CNN [4] (CVPR'21)	55.6	58.5	58.6	64.5	66.2	34.3	41.5	49.2	51.0	54.0	44.1	51.6	56.4	61.9	62.2
DeFRCN [13] (CVPR'21)	**63.3**	**67.3**	**68.1**	**71.1**	**71.2**	**45.9**	**54.7**	**60.3**	**62.8**	**63.1**	**63.7**	**65.4**	**65.5**	**68.8**	**69.2**
MPSR+Meta-Static	45.7	56.4	60.3	62.1	66.1	36.7	43.7	50.3	52.7	57.9	48.6	51.2	55.5	57.8	60.1
MPSR+Meta-Dynamic	50.2	57.2	60.6	63.3	67.0	37.0	43.9	50.4	52.5	57.8	47.9	51.8	55.4	59.1	60.2
MPSR+Meta-ScaledDynamic	51.0	57.3	60.9	63.3	67.1	37.1	44.1	50.7	52.5	57.7	48.7	52.1	56.1	59.0	60.5
MPSR+Aug	49.9	56.2	61.5	63.0	66.5	37.4	43.0	51.4	53.6	58.6	48.1	49.3	55.7	58.7	60.8
MPSR+Meta-Static+Aug	51.3	56.9	62.0	62.8	66.9	37.7	43.5	50.7	53.7	58.1	48.6	49.5	55.9	58.5	60.3
MPSR+Meta-Dynamic+Aug	51.3	56.8	62.1	63.3	67.0	37.8	44.2	51.7	54.3	59.3	48.9	50.5	56.5	59.0	61.2
MPSR+Meta-ScaledDynamic+Aug	51.9	57.6	62.4	63.7	67.6	37.8	44.9	51.9	54.2	59.4	49.2	51.9	56.7	59.7	61.1
DeFRCN+Meta-ScaledDynamic+Aug	**66.7**	**69.3**	**69.8**	**72.2**	**72.1**	**47.7**	**55.8**	**61.8**	**63.9**	**63.7**	**64.9**	**65.8**	**66.2**	**69.7**	**70.2**

Table 3. Comparison to fine-tuning based G-FSOD methods on the Pascal VOC dataset, with both base and novel classes. The best and the second-best results are marked with red and blue. The harmonic mean (HM) of the base and novel class mAPs is used for the calculation.

	Method/Shot	Novel Set 1					Novel Set 2					Novel Set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
ML	M. R-CNN [20] (ICCV'19)	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
	M. R-CNN* [20] (ICCV'19)	16.8	20.1	20.3	38.2	43.7	7.7	12.0	14.9	21.9	31.1	9.2	13.9	26.2	29.2	36.2
	FSRW [8] (ICCV'19)	14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	39.2	19.2	21.7	25.7	40.6	41.3
	MetaDet [16] (ICCV'19)	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
	FsDet [19] (ECCV'20)	25.4	20.4	37.4	36.1	42.3	22.9	21.7	22.6	25.6	29.2	32.4	19.0	29.8	33.2	39.8
	TIP [10] (CVPR'21)	27.7	36.5	43.3	50.2	59.6	22.7	30.1	33.8	40.9	46.9	21.7	30.6	38.1	44.5	50.9
	DCNet [7] (CVPR'21)	33.9	37.4	43.7	51.1	59.6	23.2	24.8	30.6	36.7	46.6	32.3	34.9	39.7	42.6	50.7
	CME [11] (CVPR'21)	41.5	47.5	50.4	58.2	60.9	27.2	30.2	41.4	42.5	46.8	34.3	39.6	45.1	48.3	51.5
	QA-FewDet [5] (ICCV'21)	41.0	33.2	35.3	47.5	52.0	23.5	29.4	37.9	35.9	37.1	33.2	29.4	37.6	39.8	41.5
	KFSOD [22] (CVPR'22)	44.6	-	54.4	60.9	65.8	**37.8**	-	43.1	48.1	50.4	34.8	-	44.1	52.7	53.9
	FCT [6] (CVPR'22)	**49.9**	57.1	57.9	**63.2**	**67.1**	27.6	34.5	**43.7**	**49.2**	51.2	39.5	**54.7**	52.3	57.0	58.7
	Meta-DETR [21] (TPAMI'22)	40.6	51.4	**58.0**	59.2	63.6	**37.0**	**36.6**	**43.7**	49.1	**54.6**	**41.6**	45.9	**52.7**	**58.9**	**60.6**
Ours	DeFRCN+Meta-ScaledDynamic+Aug	**58.4**	**62.4**	**63.2**	**67.6**	**67.7**	34.0	**43.1**	**51.0**	**53.6**	**54.0**	**55.1**	**56.6**	**57.3**	**62.6**	**63.7**

Table 4. Comparison to meta-learning based FSOD methods on the Pascal VOC dataset, with only novel classes. The best and the second-best results are marked with red and blue. MPSR+Meta-Static, MPSR+Meta-Dynamic, and MPSR+Meta-ScaledDynamic represent meta-tuning results. ML represents the meta learning based methods.

classes and demand fewer iterations, the training of the MPSR model takes 0.1 days in Pascal VOC and 0.6 days in MS COCO datasets for the proxy-base classes. RL training for meta-tuning using the final setting takes 0.05 days for Pascal VOC splits and 0.5 days for the MS COCO dataset. Finally, we note that meta-tuning operations do not incur any overhead during the fine-tuning for novel classes.

References

[1] Yuhang Cao, Jiaqi Wang, Ying Jin, Tong Wu, Kai Chen, Ziwei Liu, and Dahua Lin. Few-shot object detection via association and discrimination. *Proc. Adv. Neural Inf. Process. Syst.*, 34:16570–16581, 2021.

[2] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, Wen-Chin Chen, and Winston Hsu. Dual-awareness attention for few-shot object detection. *IEEE Transactions on Multimedia*, 2021.

[3] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 6569–6578, 2019.

[4] Zhibo Fan, Yuchen Ma, Zeming Li, and Jian Sun. Generalized few-shot object detection without forgetting. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4527–4536, 2021.

[5] Guangxing Han, Yicheng He, Shiyuan Huang, Jiawei Ma, and Shih-Fu Chang. Query adaptive few-shot object detection with heterogeneous graph convolutional networks. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 3263–3272, 2021.

[6] Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, and Shih-Fu Chang. Few-shot object detection with fully cross-transformer. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5321–5330, 2022.

[7] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10185–10194, 2021.

	Method/Shot	Split-1					Split-2					Split-3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
ML	M. R-CNN [20] (ICCV'19)	17.3	25.3	27.3	44.4	50.4	11.6	18.5	21.9	30.8	41.3	13.3	20.2	33.4	38.0	45.5
	FSRW [8] (ICCV'19)	24.2	24.8	37.8	44.2	54.2	25.5	24.9	33.8	41.5	49.0	29.7	32.4	36.7	49.9	49.9
	FsDet [19] (ECCV'20)	31.1	28.4	39.1	43.5	49.5	29.3	30.5	30.7	34.4	39.8	35.2	26.9	35.6	41.8	47.8
Ours	DeFRCN+Meta-ScaledDynamic+Aug	**58.4**	**62.4**	**63.2**	**67.6**	**67.7**	**34.0**	**43.1**	**51.0**	**53.6**	**54.0**	**55.1**	**56.6**	**57.3**	**62.6**	**63.7**

Table 5. Comparison to meta-learning based G-FSOD methods on the Pascal VOC dataset, with both base and novel classes. The best results are marked with red. The harmonic mean (HM) of the base and novel class mAPs is used for the calculation.

	Method/Shot	Novel Classes		All Classes (HM)	
		10-shot	30-shot	10-shot	30-shot
ML	ONCE [12]	1.2	-	2.2	-
	Meta R-CNN [20]	6.1	9.9	5.6	8.3
	FSRW [8]	5.6	9.1	-	-
	FsDetView [18]	7.6	12.0	6.9	10.5
	TIP [10]	16.3	18.3	-	-
	DCNET [3]	12.8	18.6	-	-
	CME [11]	15.1	16.9	-	-
	QA-FewDet [5]	10.2	11.5	-	-
	FCT [6]	17.1	21.4	-	-
	DAnA [2]	18.6	21.6	-	-
	Meta-DETR [21]	**19.0**	22.2	-	-
Ours	DeFRCN+Meta-ScaledDynamic+Aug	18.8	**23.4**	**24.4**	**28.0**

Table 6. FSOD and G-FSOD results on the MS COCO dataset with novel classes. The best results are marked with red. The harmonic mean (HM) of the base and novel class mAPs is used for the calculation.

[8] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 8420–8429, 2019.

[9] Prannay Kaul, Weidi Xie, and Andrew Zisserman. Label, verify, correct: A simple few shot object detection method. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 14237–14247, 2022.

[10] Aoxue Li and Zhenguo Li. Transformation invariant few-shot object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3094–3102, 2021.

[11] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7363–7372, 2021.

[12] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13846–13855, 2020.

[13] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 8681–8690, 2021.

[14] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7352–7362, 2021.

[15] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.

[16] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proc. IEEE Int. Conf. on Computer Vision*, pages 9925–9934, 2019.

[17] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *Proc. European Conf. on Computer Vision*, pages 456–472. Springer, 2020.

[18] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4582–4591, 2017.

[19] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *Proc. European Conf. on Computer Vision*, pages 192–210. Springer, 2020.

[20] 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 *Proc. IEEE Int. Conf. on Computer Vision*, pages 9577–9586, 2019.

[21] Gongjie Zhang, Zhipeng Luo, Kaiwen Cui, Shijian Lu, and Eric P Xing. Meta-detr: Image-level few-shot detection with inter-class correlation exploitation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[22] Shan Zhang, Lei Wang, Naila Murray, and Piotr Koniusz. Kernelized few-shot object detection with efficient integral aggregation. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 19207–19216, 2022.

[23] Weilin Zhang and Yu-Xiong Wang. Hallucination improves few-shot object detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 13008–13017, 2021.

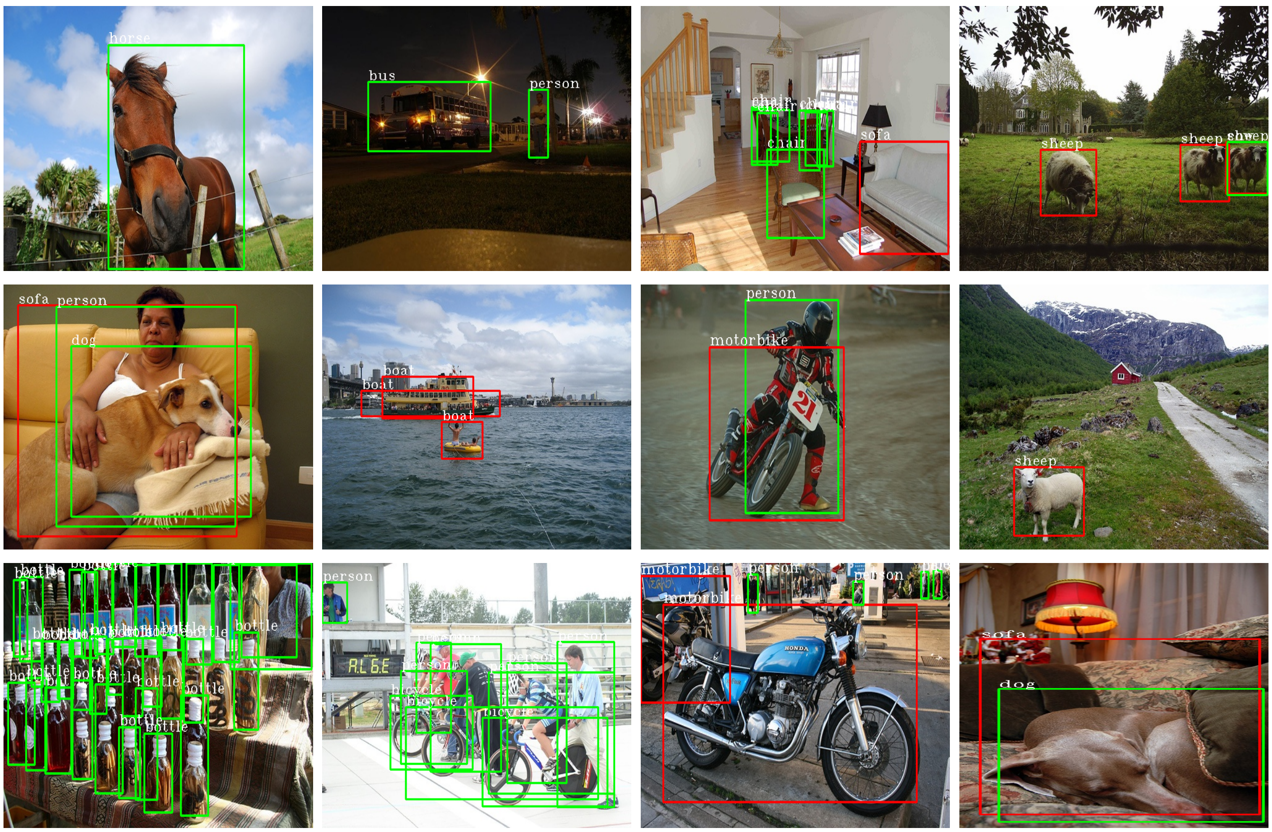

Figure 1. Randomly sampled *MPSR+Meta-ScaledDynamic+Aug* object detection results for the Pascal VOC dataset Split-3/10-shot experiment. Base class instance candidates are marked with green, and novel class instance candidates are marked with red color. (Best viewed in color.)

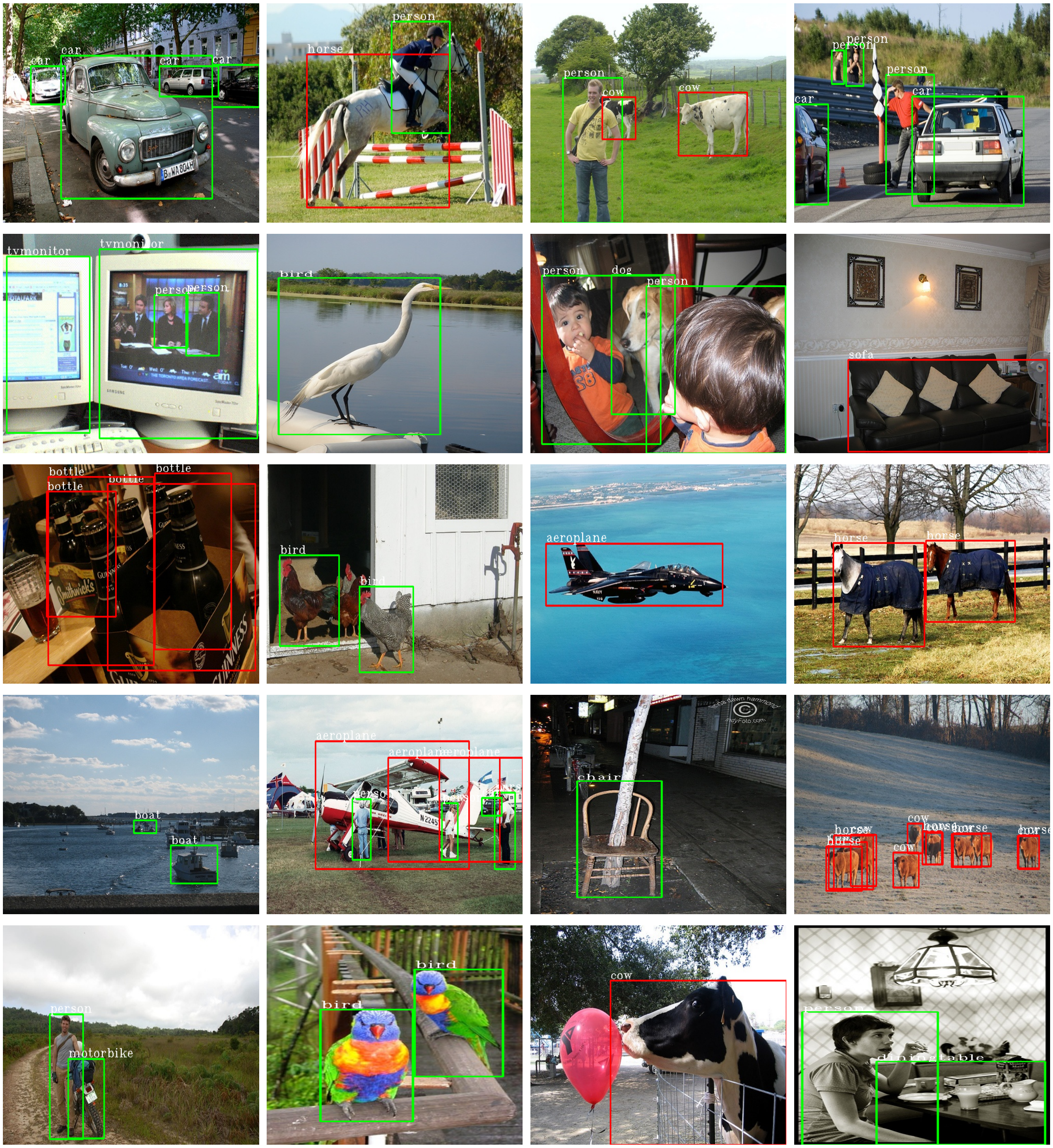

Figure 2. Randomly sampled *DeFRCN+Meta-ScaledDynamic+Aug* object detection results for the Pascal VOC dataset Split-2/10-shot experiment. Base class instance candidates are marked with green, and novel class instance candidates are marked with red color. (Best viewed in color.)