Supplementary Material for Accurate Few-shot Object Detection with Support-Query Mutual Guidance and Hybrid Loss

Lu Zhang[1] Shuigeng Zhou[1*] Jihong Guan[2] Ji Zhang[3]
[1]Shanghai Key Lab of Intelligent Information Processing, and School of Computer Science, Fudan University, China
[2]Department of Computer Science & Technology, Tongji University, China
[3]Zhejiang Laboratory, China
{l_zhang19,sgzhou}@fudan.edu.cn, jhguan@tongji.edu.cn, Ji.Zhang@zhejianglab.com

1. Additional Related Work

1.1. Object Detection

Recent advances in object detection are dominated by deep learning methods with a large number of labeled training samples. These approaches can be roughly categorized into two types: proposal-based methods and proposal-free methods. Proposal-based detectors [11, 4, 1, 2] usually have a two-stage architecture. In the first stage, they select many class-agnostic candidate boxes. Then, they classify these boxes to different categories in the second stage. Proposal-free detectors directly predict the categories and positions of the objects in one stage. Such methods (*e.g.* RetinaNet [8], FCOS [14],NETNet [6]) trade localization performance for fast inference speed and directly predict the bounding boxes and class labels.

2. Details of Datasets

2.1. More Details of PASCAL VOC

Following the setting of previous works [5, 18, 19] The novel classes of three splits for PASCAL VOC are shown in Table 1. The remaining categories are the base classes.

Splits	Novel Classes				
Novel Set 1	bird	bus	cow	mbike	sofa
Novel Set 2	aero	bottle	cow	horse	sofa
Novel Set 3	boat	cat	mbike	sheep	sofa

Table 1. Novel classes of three different splits for PASCAL VOC.

*correspondence author

2.2. More Details of MS COCO

We noticed that there are some invalid annotations in COCO that label the width or height of bounding boxes with values less than or equal to 1. In order to prevent the bad influence of these invalid annotations, we filter out them. The number of invalid annotations for training/testing is 79 and 1 respectively.

3. Additional Experimental Results

Due to page limit, we put the results under 30-shot on COCO in the supplementary material. The results are shown in Table 2. Again, our methods with ResNet-50 as backbone has already achieve SOTAs. Our method with ResNet-101 as backbone brings further performance improvements with 1.1%AP and outperforms existing SOTA(MPSR[17]) by 1.8%, 2.1% and 1.3% in terms of AP, AP_{50} and AP_{75}, proving the generalization power of our approach on novel classes. Nevertheless, we believe that the evaluation under 10-shot has a higher reference value in FSOD task.

4. Implementation Details

4.1. Class-Agnostic Regressor (CAR)

CAR takes the features of proposals as input and outputs the coordinate adjustment values (dx, dy, dh, dw), where dx and dy are used to adjust the top-left coordinates of the proposals, dh and dw are used to adjust the height and width. CAR module is class-agnostic, which means it does not distinguish between categories when regresses the adjustment coordinates and all categories of objects share the same parameters.

The structure of CAR is shown in Fig. 1, where $\overline{P_j}$ denotes proposal j output from RPN. In order to speed up

Method	Backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
LSTD[3]	SSD	6.7	15.8	5.1	0.4	2.9	12.3
MetaYOLO[5]	DarkNet-19	9.1	19.0	7.6	0.8	4.9	16.8
MetaDet[16]	VGG-16	11.3	21.7	8.1	1.1	6.2	17.3
MetaRCNN[18]	ResNet-101	12.4	25.3	10.8	2.8	11.6	19.0
TFA w/cos[15]	ResNet-101	13.7	–	13.4	–	–	–
MPSR[17]	ResNet-101	14.1	25.4	14.2	4.0	12.9	23.0
Ours	ResNet-50	14.7	29.4	13.0	8.3	15.6	22.1
Ours	ResNet-101	**15.9**	**31.5**	**14.3**	**8.6**	**17.0**	**23.1**

Table 2. Results on the COCO minival set for 20 novel classes under 30-shot. '–': No reported results.

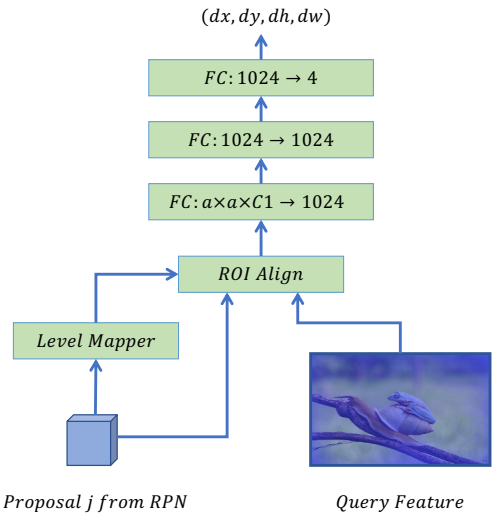

Figure 1. Illustration of CAR to adjust the coordinates of proposal j from RPN .

the calculation, feature of proposal j is extracted from one level in CAR. Firstly, the level mapper chooses the optimal level to carry out ROI Align operation for proposal j, whose implementation details follow [7]. Then, we carry out ROI Align in optimal level and get size-fixed feature. Then, feature from ROI Align is flattened, input into a fully connected network, whose output are the coordinate adjustment values for proposal j.

4.2. Implementation of Training and Evaluating

Our implementations are based on maskrcnn-benchmark [9] and PyTorch [10]. In each meta-training episode, we sample 8 query images containing the objects of support class, with each GPU 2 images. We denote the larger width and height of 2 images in one GPU as w_l and h_l respectively, and pad these 2 images with zero to make their sizes uniform to (w_l,h_l). The padding is performed on the right and bottom of images. When inference, we keep the top-1000 proposals from RPN, all of which are fed into CAR to refine locations. Before output the final detection results, the predicted bounding boxes are filtered with score threshold of 0.05. After that, we perform NMS operation with a IoU (Intersection over Union) threshold of 0.5. When training, the proposals having IoU larger than 0.6 with a annotated bounding boxes will be considered as matching that bounding box. Proposals having IoU less than 0.3 with any annotated bounding boxes will be considered as background proposals. Others will be ignored. For MPS, $\mathcal{D}_\varphi$ is not shared across all levels. In our experiments, we set a independent $\mathcal{D}_\varphi$ in each level. For k-shot setting, we sample k objects from z images, where $z \leq k$.

4.3. Details of Ablation Study

4.3.1 Hyperparameter Analysis

Actually, hyperparameter optimization is a multi-dimensional search problem. We convert it into a one-dimensional search by fixing other hyperparameter. For example, when we search the optimal γ, we fix $\alpha = 1.5, \beta = 1.5, b = 1$. However, with the help of some search algorithms, better results may be achieved.

4.3.2 Will Kernel Generator and CAR Improve Recall of Proposals?

In the ablation, we calculate the recall of proposals with/without kernel generator and CAR. It is worth noting that the recall in this section denotes the recall of proposals containing objects of same class with supports in query image, install of recall of all objects in query image. Although other objects of different classes are foreground, but they are negative.

5. Additional Ablation Study

5.1. Visualization of Proposals Distribution

To further understand the effects of *SPG* module, we visualize the distribution of generated proposals as a heatmap, and compare the heatmaps in the cases with and without the guidance of supports. As Fig. 2 shows, *SPG* generates more proposals distributed on the correct bounding boxes, which

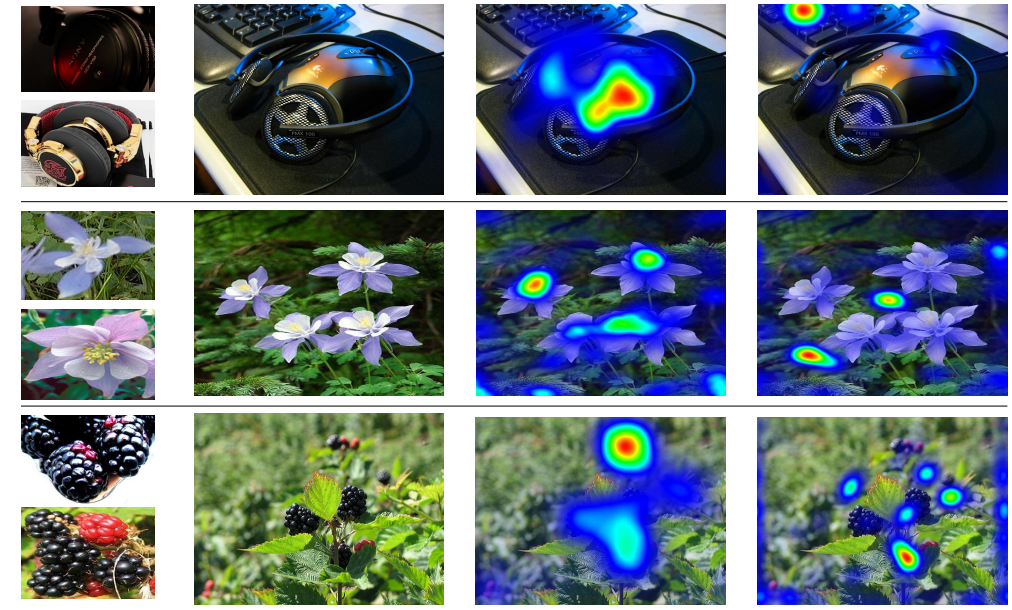

Figure 2. Comparison of proposal distributions: with *SPG* vs. without *SPG*

means with SPG our method can more correctly generate proposals.

5.2. Different Implementation of Distance Metric Function

We tried different implementation of distance metric function $\mathcal{D}_\varphi$ to perform few-shot classification for proposals, including relation network [13], prototypical network [12]. The results are summarized in Table 3. As we can see, our proposed method with relation network and prototypical network have similar performance. The difference is not significant.

$\mathcal{D}_\varphi$	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
prototypical network	12.3	26.5	11.1	7.0	13.2	17.9
relation network	12.6	27.0	10.9	7.3	13.4	17.8

Table 3. Different implementation of $\mathcal{D}_\varphi$. The experiments are carried out in COCO novel set under 10-shot.

5.3. Different Attention Mechanism to Weight Supports

CANet [20] proposed an attention mechanism to weight different supports in few-shot segmentation. In CANet, the attention scores are directly obtained from the convolution output of each support, without the guidance of query. Different from CANet, we exploit the similarity between supports and query to mine the contribution of each support. With the guidance of query, the attention scores to weight supports are generated. We introduce the attention mechanism in CANet to our framework and compare with our QSW module. The results are shown in Table 4. Our design of attention mechanism in QSW outperforms the attention in CANet, which demonstrates the significance of the guidance from query.

Figure 3. Some bad cases on the first split of VOC.

Attention Mechanism	AP	AP_{50}	AP_{75}
Attention in CANet	10.9	23.6	9.8
QSW	11.6	25.2	10.3

Table 4. Comparison of different attention mechanism to weight supports. The experiments are carried out in COCO under 3-shot.

6. Some Failure Case

We provide some failure cases for better understanding of our model, as shown in Fig. 3.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 1

[2] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. In *CVPR*, 2020. 1

[3] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *AAAI*, 2018. 2

[4] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 1

[5] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *ICCV*, 2019. 1, 2

[6] Yazhao Li, Yanwei Pang, Jianbing Shen, Jiale Cao, and Ling Shao. Netnet: Neighbor erasing and transferring network for better single shot object detection. In *CVPR*, 2020. 1

[7] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 2

[8] Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:2999–3007, 2018. 1

[9] Francisco Massa and Ross Girshick. maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch. https://github.com/facebookresearch/maskrcnn-benchmark, 2018. 2

[10] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 2

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 1

[12] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NIPS*, 2017. 3

[13] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H.S. Torr, and Timothy M. Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018. 3

[14] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019. 1

[15] Xin Wang, Thomas Huang, Joseph Gonzalez, Trevor Darrell, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020. 2

[16] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019. 2

[17] Jiaxi Wu, Songtao Liu, Di Huang, and Yunhong Wang. Multi-scale positive sample refinement for few-shot object detection. In *ECCV*, 2020. 1, 2

[18] 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 *ICCV*, 2019. 1, 2

[19] Ze Yang, Yali Wang, Xianyu Chen, Jianzhuang Liu, and Yu Qiao. Context-transformer: Tackling object confusion for few-shot detection. In *AAAI*, 2020. 1

[20] Chi Zhang, Guosheng Lin, Fayao Liu, Rui Yao, and Chunhua Shen. Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In *CVPR*, 2019. 3