Supplementary Materials for Few-Shot Object Detection with Fully Cross-Transformer

Guangxing Han, Jiawei Ma, Shiyuan Huang, Long Chen, Shih-Fu Chang
Columbia University
{gh2561,jiawei.m,sh3813,cl3695,sc250}@columbia.edu

The supplementary materials are organized as follows. First, we describe the implementation details of our training framework in Section 1. Then, we show more visualization examples of our multi-level cross-attention in Section 2.

1. Implementation Details for Model Training

We have three steps for model training. In the first two steps, the model is trained over the data-abundant base-classes dataset. In the last step, the model is fine-tuned over novel classes and then used for evaluation.

(1) Pre-training the single-branch based model over base classes. In the first step, we train the single-branch based few-shot object detection model using large-scale base-class dataset. For model architecture, we use the vallina Faster R-CNN model with the vision transformer backbone (PVTv2 [3, 4] in this work). The model is initialized from the ImageNet pretrained model provided by [3]. We follow the original Faster R-CNN paper for model training. The loss function is defined as,

$$\mathcal{L}_1 = \mathcal{L}_{rpn} + \mathcal{L}_{rcnn} \tag{1}$$

where $\mathcal{L}_{rpn}$ and $\mathcal{L}_{rcnn}$ both consist of a classification loss and a bbox regression loss as follows,

$$\mathcal{L}_{rpn} = \mathcal{L}^B_{cls} + \mathcal{L}_{loc}, \quad \mathcal{L}_{rcnn} = \mathcal{L}^M_{cls} + \mathcal{L}_{loc} \tag{2}$$

where $\mathcal{L}^B_{cls}$ denotes the binary cross-entropy loss over a "foreground" class (the union of all base classes) and a "background" class, and $\mathcal{L}^M_{cls}$ denotes the multi-class cross-entropy loss over all base classes plus a "background" class. The $\mathcal{L}_{loc}$ denotes the bbox regression loss using smooth L_1 loss defined in [2].

For model training on the MSCOCO dataset, we use the AdamW optimizer with an initial learning rate of 0.0002, weight decay of 0.0001, and a batch size of 8. The learning rate is divided by 10 after 85,000 and 100,000 iterations. The total number of training iterations is 110,000.

Similarly, we use smaller training iterations for model training on the PASCAL VOC dataset. The initial learning rate is 0.0002, divided by 10 after 12,000 and 16,000 iterations. The total number of training iterations is 18,000.

(2) Training the two-branch based model over base classes. In this step, we train the proposed two-branch based model with fully cross-transformer (FCT). The model is initialized by the pretrained model in the first step. Our FCT model can reuse most of the parameters of the single-branch based model in the first step, and only need to learn the branch embedding and pairwise matching network [1]. We also show in Table 3 of the main paper the importance of the pre-trained single-branch based model. The loss function is defined as,

$$\mathcal{L}_2 = \mathcal{L}_{att\_rpn} + \mathcal{L}_{matching} \tag{3}$$

where $\mathcal{L}_{att\_rpn}$ and $\mathcal{L}_{matching}$ both consist of a binary cross-entropy loss and a bbox regression loss,

$$\mathcal{L}_{att\_rpn} = \mathcal{L}^B_{cls} + \mathcal{L}_{loc}, \quad \mathcal{L}_{matching} = \mathcal{L}^B_{cls} + \mathcal{L}_{loc} \tag{4}$$

where the Attention-RPN and the pairwise matching network both use the binary cross-entropy loss $\mathcal{L}^B_{cls}$ and bbox regression loss $\mathcal{L}_{loc}$ for training, following [1].

For model training on the MSCOCO dataset, we use the AdamW optimizer with an initial learning rate of 0.0001, weight decay of 0.0002, and a batch size of 4. The learning rate is divided by 10 after 15,000 and 20,000 iterations. The total number of training iterations is 20,000. We use much smaller iterations than in the first step thanks to the good initialization.

For model training on the PASCAL VOC dataset, we use the same hyper-parameters as on the MSCOCO dataset except using fewer training iterations. The initial learning rate is 0.0002, divided by 10 after 7,500 and 10,000 iterations. The total number of training iterations is 10,000.

(3) Fine-tuning the two-branch based model over novel classes. In this step, the model is fine-tuned using a sub-sampled K-shot dataset with both base classes and novel classes. We use the same loss function in the second step for model training. After the few-shot fine-tuning, the learned model is used for evaluation.

For model training on the MSCOCO and PASCAL VOC dataset, we use the AdamW optimizer with an initial learning

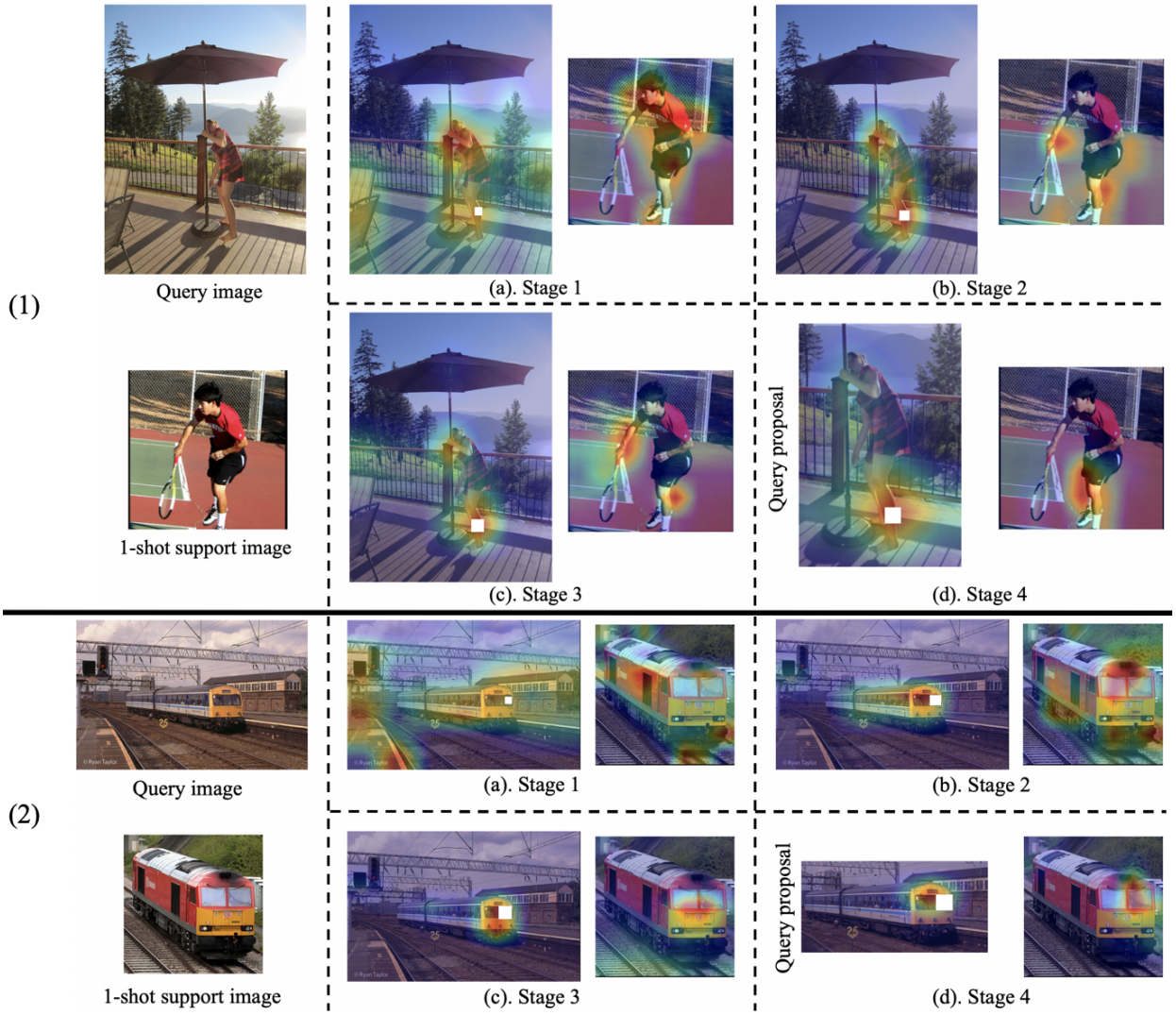

Figure S1. Visualization of the multi-level cross-attention, similar as the Figure 4 in the main paper.

rate of 0.0002, weight decay of 0.0001, and a batch size of 4. For 30-shot fine-tuning, the learning rate is divided by 10 after 3,000 iterations, and the total number of training iterations is 5,000. For 10-shot fine-tuning or fewer, the learning rate is divided by 10 after 2,000 iterations, and the total number of training iterations is 3,000.

2. Visualization of Multi-level Cross-attention

We show more visualization examples of the proposed multi-level cross-attention in Figure S1 and S2.

References

[1] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4013–4022, 2020. 1

[2] Ross Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1

[3] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 1

[4] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 568–578, October 2021. 1

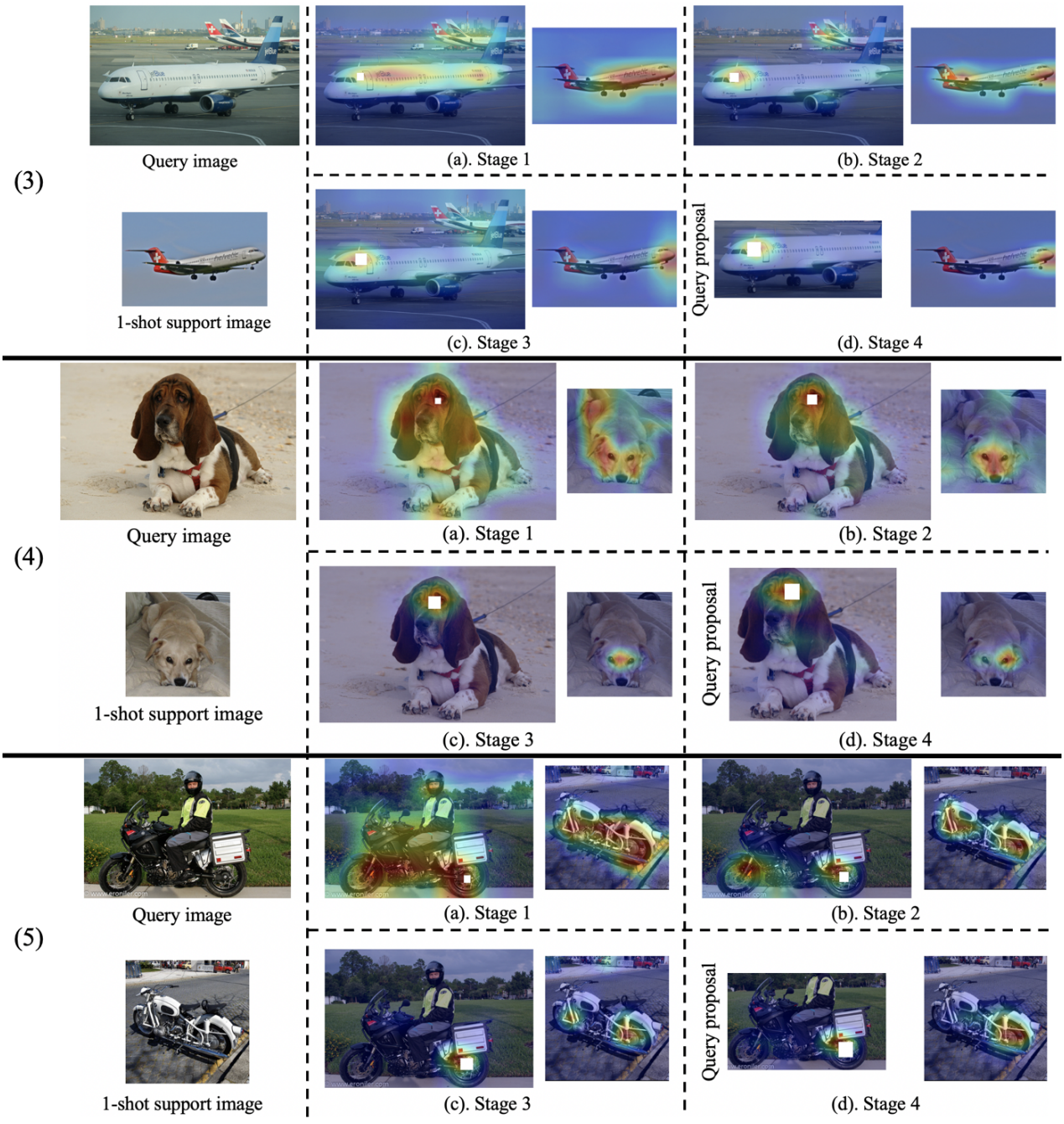

Figure S2. Visualization of the multi-level cross-attention, similar as the Figure 4 in the main paper.