Scale-Equivalent Distillation for Semi-Supervised Object Detection

Qiushan Guo[1], Yao Mu[1], Jianyu Chen[2], Tianqi Wang [1], Yizhou Yu[1], Ping Luo[1]
[1]The University of Hong Kong [2]Tsinghua University
{qsguo,ymu,tqwang,yzyu,pluo}@cs.hku.hk jianyuchen@tsinghua.edu.cn

A. Appendix

A.1. Implementation of SED on DETR.

Our method can be extended to DETR, a single feature map detector based on anchor-free label assignment rule. We match the predictions of input in different views according to Hungarian algorithm, where the pair-wise matching cost is defined as: $L_{\text{match}} = D_{\text{JS}}(p_1, p_2) + \lambda L_{\text{IoU}}(b_1, b_2)$, where $D_{\text{JS}}(p_1, p_2)$ is JS-Divergence between the probability vectors and L_{IoU} is GIoU loss [10]. The python-style pseudo-code of matching algorithm is provided in Alg. 1. The DETR model is trained with AdamW setting the transformer's learning rate to 10^{-4}, the backbone's learning rate to 10^{-5}, and weight decay to 10^{-4}. The model is trained with a long schedule for 300 epochs and the learning rate is multiplied by 0.1 at 200 epochs. The other settings are the same as DETR [1].

A.2. Stronger augmentations.

Geometric augmentations are common image data augmentations. Therefore, we further conduct experiments with stronger augmentations: color + geometric augmentations, to demonstrate the extendability of SED. We simply adopt the same geometric transformations in RandAug [3], including *RandRotate*, *RandTranslation* and *RandShear*. We set the rand level to 5 and select only 1 transformation to apply. The results in Tab. 1 show that additional geometric augmentations lead to incremental improvement.

A.3. Implementation and Training Details.

Our implementation is based on MMDetection framework [2]. The default detector is set as Faster-RCNN [9] with FPN [7] and ResNet-50 [5] for a fair comparison with prior works [8, 11–14]. **Code will be released.**

Training Details. The weights of the backbone are first initialized by the corresponding ImageNet-Pretrained model, which is a default setting in existing works [6, 8, 11, 14]. All the models are trained with learning rate starting at 0.01. The learning rate drops by 0.1 at the 120k and 160k iteration for 180k training schedule as default. We set the weight decay to 0.0001, batch size to 16, and the momentum is 0.9 for SGD optimizer. Like [8], we separate 5k/10k/12k/90k iterations from the whole process as the burn-in phase for 5%/10%/35k/100% data protocols. For verifying the effectiveness of our method, we simply set the λ_s and λ_d in Eq. 1 as 0.5 and 1 separately. The EMA update rate starts with 0.99 and steps to 0.9 at the 120k iteration, aligned with the learning rate decay policy.

Data Augmentation. As shown in Tab. 2, the weak data augmentation only contains random resize from (1333, 640) to (1333, 800) and random horizontal flip with a probability of 0.5. The strong data augmentation is composed of random Color Jittering, Grayscale, Gaussian Blur, and Cutout [4], without any geometric augmentation.

Algorithm 1 Matching Pseudocode, PyTorch-like

```
def hungarian_match(cls_score_1, cls_score_2,
    bbox_pred_1, bbox_pred_2, cls_weight,
    iou_weight):
    # cls_score: [bs, num_query, c]
    # bbox_pred: [bs, num_query, 4]
    cls_dist = JSCost(cls_score_1, cls_score_2)
    iou_dist = IoUCost(bbox_pred_1, bbox_pred_2)
    cost= cls_dist*cls_weight + iou_dist*
    iou_weight

    bs = cost.shape[0]
    col_inds = []
    for i in range(bs):
        col_ind = linear_sum_assignment(cost[i])
        col_inds.append(col_ind)
    return col_inds
```

Method	Data	AP	Augmentation
Supervised	VOC07	74.3	-
STAC [11]	VOC07+12	77.45	C, G
DGML [12]	VOC07+12	78.60	-
UBT [8]	VOC07+12	77.37	C
ISMT [13]	VOC07+12	77.23	C, DropBlock
IT [14]	VOC07+12	78.30	C, Mixup, Mosaic
Ours	VOC07+12	**80.60**	**C**
Ours	VOC07+12	**81.44**	**C, G**

Table 1. Results on Pascal VOC 2007 test set. AP_{50} is reported. "-" means that the training details are missing in the source paper.

Strong Augmentation			
Process	Probability	Parameters	Details
Color Jittering	0.8	brightness, contrast, saturation = 0.4, 0.4, 0.4	Brightness factor is chosen uniformly from [0.6, 1.4], Contrast factor is chosen uniformly from [0.6, 1.4], Saturation factor is chosen uniformly from [0.6, 1.4]
Grayscale	0.2	None	None
GaussianBlur	0.5	$\sigma \sim U(0.1, 2.0)$	Gaussian filter kernel size is 23
Cutout 1	0.7	scale=(0.05, 0.2), ratio=(0.3, 3.3)	Randomly selects a rectangle region in an image
Cutout 2	0.5	scale=(0.02, 0.2), ratio=(0.1, 6)	Randomly selects a rectangle region in an image
Cutout 3	0.3	scale=(0.02, 0.2), ratio=(0.05, 8)	Randomly selects a rectangle region in an image

Table 2. Details of data augmentations.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 1

[3] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 1

[4] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[6] Jisoo Jeong, Seungeui Lee, Jeesoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. *Advances in neural information processing systems*, 32:10759–10768, 2019. 1

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[8] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. *arXiv preprint arXiv:2102.09480*, 2021. 1

[9] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1

[10] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 1

[11] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020. 1

[12] Zhenyu Wang, Yali Li, Ye Guo, Lu Fang, and Shengjin Wang. Data-uncertainty guided multi-phase learning for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4568–4577, 2021. 1

[13] Qize Yang, Xihan Wei, Biao Wang, Xian-Sheng Hua, and Lei Zhang. Interactive self-training with mean teachers for semi-supervised object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5941–5950, 2021. 1

[14] Qiang Zhou, Chaohui Yu, Zhibin Wang, Qi Qian, and Hao Li. Instant-teaching: An end-to-end semi-supervised object detection framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4081–4090, 2021. 1