Self-Supervised Visual Representations Learning by Contrastive Mask Prediction
Supplementary Material

Yucheng Zhao*†1 Guangting Wang*†1 Chong Luo[2] Wenjun Zeng[2] Zheng-Jun Zha‡1
University of Science and Technology of China[1] Microsoft Research Asia[2]
{lnc, flylight}@mail.ustc.edu.cn {cluo, wezeng}@microsoft.com zhazj@ustc.edu.cn

A. Additional Implementation Details

A.1. Pseudo Code for MaskCo Training Loop

We provide the pseudo code for MaskCo training loop in Algorithm 1.

A.2. Additional Training Details for Pre-training

Negative Sampling Strategy: In the pre-training stage, our default model uses $M \times N = 16 \times 31 = 496$ negative samples from the same GPU, where $M = 16$ is the number of negative boxes per image, and $N = 31$ is the number of images per GPU minus one (excluding the same image). The MaskCo(+in) model uses additional intra-image negatives, which is additional 16 negative samples from the same image so that the number of total negative samples become 512.

Pre-training on Multiple Datasets: In all pre-training datasets, including ImageNet, CC, and COCO, we use exactly the same training hyper-parameters that is tuned on the ImageNet.

A.3. Training Details for Downstream Tasks

ImageNet Linear Classification: We adopt the ImageNet Linear Classification protocol used in [6]. The features from different residual layers, from conv1, conv2, until conv5, are extracted, and additional pooling and linear layers are added on top of the extracted features. Only the additional linear layer is trainable. We call this evaluation protocol as *multi-layer linear evaluation*. There is another linear evaluation protocol used in MoCo [8] in which only the global average pooling features were used to train the linear classifier. We call this evaluation protocol as *last-layer linear evaluation*. We observe that the last-layer protocol is biased towards the ID-based methods, and SSL methods based on other pretext tasks usually do not produce the best result at the final layer. Since we intend to develop a new pretext task, we choose the **multi-layer** protocol to evaluate the capabilities of different pre-trained layers. Moreover, the multi-layer protocol was also used by a lot of previous works [9, 1, 5]. We use exactly the same training configurations as in [1], and no hyper-parameter tuning is performed. The initial learning rate is set to 0.1 with momentum 0.9 and weight decay 1e-4. The total fine-tuning epoch is set to 90, and the learning rate is decayed by 10 at epochs 30 and 60.

Pascal VOC Object Detection: We use the exact configurations as in [2]. The Faster RCNN with ResNet-50-R4 backbone is trained on `trainval07+12` set and evaluated on `test2007` set. We train for 24K iterations using SGD optimizer with batch size 16 (2 per GPU). We use the base learning rate 0.02, perform warmup for 100 iterations, and divided it by 10 at iterations 18K and 22K.

COCO Object Detection and Instance Segmentation: We use the exact configurations as in [3], which is the standard 2x schedule in [10].

B. Additional Experimental Results

The complete results of ImageNet Linear Classification: Our main paper only reports the ImageNet linear classification results of conv4 and conv5 in Table 3. For completeness, we list the complete results from all residual layers in Table 1 of this supplementary document.

More Visualization results of MPH: We also present additional visualization results of MPH in Figure 1.

Ablation on COCO Pre-Training: We report the ablation results of the mask strategy when our model is pretrained on the COCO dataset in Table 2. The trend is almost identical to what we find in ImageNet pre-training.

* Equal contribution. † Interns at MSRA. ‡ Corresponding author.

[1] https://github.com/open-mmlab/OpenSelfSup/blob/master/configs/benchmarks/linear_classification/imagenet/r50_multihead.py

[2] https://github.com/open-mmlab/OpenSelfSup/blob/master/benchmarks/detection/configs/pascal_voc_R_50_C4_24k_moco.yaml

[3] https://github.com/open-mmlab/OpenSelfSup/blob/master/benchmarks/detection/configs/coco_R_50_C4_2x_moco.yaml

Algorithm 1: Pseudocode for MaskCo training loop.

```
# net_q: encoder for query image, including the backbone and MPH
# net_k: momentum encoder for key image
# head_q: projection head for query image
# head_k: projection head for key image
# m: momentum
# t: temperature
# x: input image

# generate two views, masked box, and key boxes
x_q, x_k, masked_box, key_boxes = transform(x)

q = head_q(roi_align(net_q(x_q), masked_box) # queries: Nx1xC
k = head_k(roi_align(net_k(x_k), key_boxes) # keys: NxKxC
k = k.detach() # no gradient to keys

pos_k = k[:, 0:1]
neg_k_inter = sample_inter(k) # inter-image negatives: NxK1xC
neg_k_intra = sample_intra(k) # intra-image negatives: NxK2xC
neg_k = cat([neg_k_inter, neg_k_intra], dim=1)


l_pos = bmm(q, pos_k.transpose(1, 2)) # positive logits: Nx1
l_neg = bmm(q, neg_k.transpose(1, 2)) # negative logits: Nx(K1+K2)
logits = cat([l_pos, l_neg], dim=1)

# MoCo contrastive loss (Positive labels at index 0).
labels = zeros(N)
loss = CrossEntropyLoss(logits/t, labels)

# SGD update: query network
loss.backward()
update(net_q.params)
update(head_q.params)

# momentum update: key network
net_k.params = m*net_k.params+(1-m)*net_q.params
head_k.params = m*head_k.params+(1-m)*head_q.params
```

Method	ImageNet				
	conv1	conv2	conv3	conv4	conv5
Rand Init	11.4	16.2	13.5	9.1	6.5
Supervised	15.2	34.0	47.9	67.6	76.2
Relative-Pos [4]	14.8	31.3	45.8	49.3	40.2
Rotation-Pred [5]	12.9	34.3	44.9	55.0	49.1
NPID [11]	14.3	31.2	40.7	54.5	56.6
MoCo v2 [3]	14.7	32.8	45.0	<u>61.6</u>	<u>66.7</u>
SimCLR [2]	17.1	31.4	41.4	54.4	61.6
BYOL [7]	15.5	34.5	47.2	**62.8**	**71.6**
MaskCo	15.4	33.6	45.8	59.6	65.1

Table 1. The complete ImageNet linear classification results of Table 3 of the main paper.

Mask	Pascal VOC			ImageNet	
	AP	AP_{50}	AP_{75}	conv4	conv5
✓	56.2	81.4	62.6	55.3	38.5
	54.1	79.5	59.5	51.8	51.1

Table 2. Ablation studies on the mask strategy when our model is pre-trained on the COCO dataset. MPH is not used in these experiments.

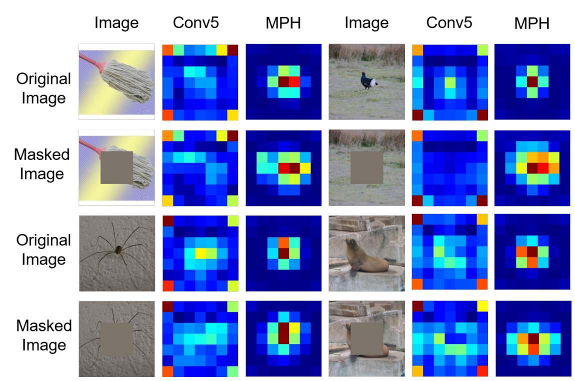

Figure 1. Additional visualization results of MPH.

References

[1] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV (14)*, volume 11218 of *Lecture Notes in Computer Science*, pages 139–156. Springer, 2018. 1

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020. 2

[3] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *CoRR*, abs/2003.04297, 2020. 2

[4] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction.

In *ICCV*, pages 1422–1430. IEEE Computer Society, 2015. 2

[5] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR (Poster)*. OpenReview.net, 2018. 1, 2

[6] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, pages 6390–6399. IEEE, 2019. 1

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent - A new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[8] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9726–9735. IEEE, 2020. 1

[9] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV (6)*, volume 9910 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2016. 1

[10] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 1

[11] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742. IEEE Computer Society, 2018. 2