Dense Contrastive Learning for Self-Supervised Visual Pre-Training Supplementary

A. Implementation Details

Dense projection head. In our implementation, the dense projection head consists of adaptive average pooling (optional), 1×1 convolution, ReLU, and 1×1 convolution. Following [1, 2], the hidden layer's dimension is 2048, and the final output dimension is 128.

COCO learning rate. For COCO pre-training including both baseline and ours, we use an initial learning rate of 0.3 instead of the original 0.03, as the former shows better performance in MoCo-v2 baseline when pre-training on COCO. The results are reported in Table 1.

	Detection			Classification
lr	AP	AP_{50}	AP_{75}	mAP
0.03	56.4	81.3	62.6	79.8
0.3	56.7	81.7	63.0	82.9

Table 1 – Learning rate comparison. The results are from 800-epoch COCO pre-trained MoCo-v2. The detection performance is evaluated by fine-tuning the pre-trained models on VOC0712. We also provide results of VOC2007 SVM Classification.

Fine-tuning details. We provide more details about evaluation by fine-tuning. For COCO object detection and segmentation with Mask R-CNN, we follow the settings in [8]. Synchronized batch normalization is used in backbone, FPN [5] and prediction heads during the training. For semantic segmentation, we evaluate the pre-trained models by fine-tuning an FCN-8s [6]. We follow the settings in `mmsegmentation` [7], except that the first 7×7 convolution is kept to be consistent with the pre-trained models. Batch size is set to 16. Synchronized batch normalization is used. Crop size is 512 for VOC [4] and 769 for Cityscapes [3].

B. Semi-Supervised Object Detection

In Table 2, we evaluate the pre-trained models on semi-supervised object detection. In this semi-supervised setting, only 10% training data is used during the fine-tuning. We evaluate by fine-tuning a Mask R-CNN (FPN backbone) for 90k iterations on COCO `train2017` and tested on COCO `val2017`. DenseCL outperforms MoCo-v2 by 1.3% AP^b and 1.0% AP^b when pre-training on COCO and ImageNet respectively. It should be noted that the gains are more significant than that of the fully-supervised setting which uses all of ~118k images during the fine-tuning. For example, when pre-training on ImageNet, DenseCL surpasses MoCo-v2 by 1.0% AP^b and 0.5% AP^b for semi-supervised setting and fully-supervised setting respectively.

pre-train	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
semi-supervised						
random init.	20.6	34.0	21.5	18.9	31.7	19.8
super. IN	23.6	37.7	25.4	21.8	35.4	23.2
MoCo-v2 CC	22.8	36.4	24.2	20.9	34.6	21.9
DenseCL CC	24.1	38.1	25.6	21.9	36.0	23.0
MoCo-v2 IN	23.8	37.5	25.6	21.8	35.4	23.2
DenseCL IN	24.8	38.8	26.8	22.6	36.8	23.9
fully-supervised						
MoCo-v2 CC	38.5	58.1	42.1	34.8	55.3	37.3
DenseCL CC	39.6	59.3	43.3	35.7	56.5	38.4
MoCo-v2 IN	39.8	59.8	43.6	36.1	56.9	38.7
DenseCL IN	40.3	59.9	44.3	36.4	57.0	39.2

Table 2 – Semi-supervised object detection and instance segmentation fine-tuned on COCO. During the fine-tuning, only 10% training data is used. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet respectively. All the detectors are trained on `train2017` for 90k iterations and evaluated on `val2017`. The metrics include bounding box AP (AP^b) and mask AP (AP^m).

C. Visualization

Given two views of the same image, we use the pre-trained backbone to extract the features $\mathbf{F}_1$ and $\mathbf{F}_2$. For each feature vector in $\mathbf{F}_1$, we find the corresponding feature vector in $\mathbf{F}_2$ which has the highest cosine similarity. The match is kept if the same match holds from $\mathbf{F}_2$ to $\mathbf{F}_1$. Each match is assigned an averaged similarity. In Figure 1, we visualize the high-similarity matches (*i.e.*, similarity ≥ 0.9). DenseCL extracts many more high-similarity matches than its baseline. It is in accordance with our intention that the local features extracted from the two views of the same image should be similar.

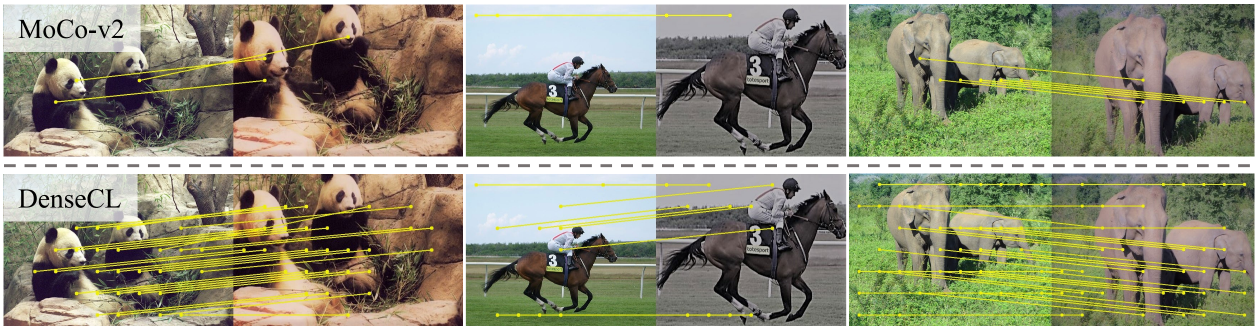

Figure 1 – Visualization of dense correspondence. The correspondence is extracted between two views of the same image, using the 200-epoch ImageNet pre-trained model. DenseCL extracts more high-similarity matches compared with MoCo-v2. Best viewed on screen.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, 2020. 1

[2] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv: Comp. Res. Repository*, 2020. 1

[3] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3213–3223, 2016. 1

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vision*, 88(2):303–338, 2010. 1

[5] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017. 1

[6] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015. 1

[7] OpenMMLab. mmsegmentation. `https://github.com/open-mmlab/mmsegmentation`, 2020. 1

[8] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeurIPS*, 2020. 1