

Rethinking Self-Supervised Learning: Small is Beautiful

Yun-Hao Cao and Jianxin Wu

National Key Laboratory for Novel Software Technology

Nanjing University, Nanjing, China

{caoyunhao1997, wujx2001}@gmail.com

Abstract

Self-supervised learning (SSL), in particular contrastive learning, has made great progress in recent years. However, a common theme in these methods is that they inherit the learning paradigm from the supervised deep learning scenario. Current SSL methods are often pretrained for many epochs on large-scale datasets using high resolution images, which brings heavy computational cost and lacks flexibility. In this paper, we demonstrate that the learning paradigm for SSL should be different from supervised learning and the information encoded by the contrastive loss is expected to be much less than that encoded in the labels in supervised learning via the cross entropy loss. Hence, we propose scaled-down self-supervised learning (S3L), which include 3 parts: small resolution, small architecture and small data. On a diverse set of datasets, SSL methods and backbone architectures, S3L achieves higher accuracy consistently with much less training cost when compared to previous SSL learning paradigm. Furthermore, we show that even without a large pretraining dataset, S3L can achieve impressive results on small data alone. Our code has been made publicly available at <https://github.com/CupidJay/Scaled-down-self-supervised-learning>.

1. Introduction

Deep supervised learning has achieved great success in the last decade. However, its dependency on image labels has driven people to explore a better solution. Self-supervised learning (SSL) has gained popularity because of its ability to avoid the cost of annotating large-scale datasets. After the emerging of the InfoNCE loss [36] and the contrastive learning paradigm, SSL has clearly gained momentum and a large amount of research contributions have been published, such as MoCo [17] (and MoCov2 [8]), SimCLR [6] (and SimCLRv2 [7]), BYOL [16] and many more.

A common theme in all these methods, however, is that they all learn self-supervised models in a setup that is clearly inherited from the supervised learning setting. Common

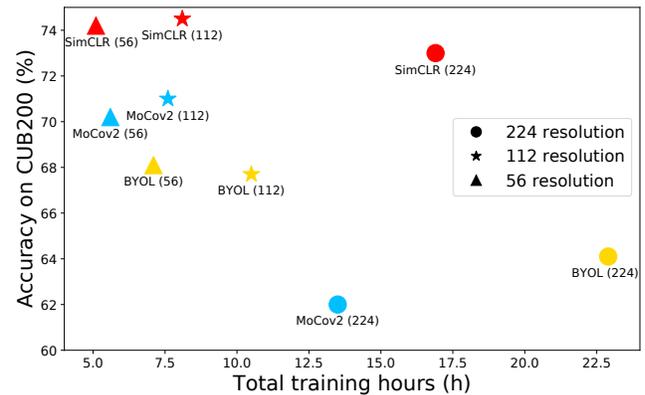


Figure 1: Accuracy of SSL methods on CUB200 [37] with ResNet-50 backbone. Only the small CUB200 dataset was used in SSL pretraining and subsequent fine-tuning. In each method (MoCov2, SimCLR and BYOL), smaller resolution not only trained much faster, but also achieved higher accuracy than the baseline 224 resolution.

characteristics of these methods include: (1) Use 224x224 as the input resolution; (2) Use large-scale training sets (e.g., ILSVRC-2012 [33]); (3) Use the entire network architecture from supervised learning tasks (mostly ResNet-50 [20]) and the entire backbone network afterwards in downstream tasks; (4) Require many training (e.g., 800 or more) epochs. The combination of these four characteristics dominates current SSL researches. This combination, however, has exhibited clear disadvantages in three important aspects:

- **Computing.** The combination of a large-scale dataset, a large input resolution, a large number of epochs and a complex backbone network means that SSL methods are computationally extremely expensive. This phenomenon makes SSL a privilege for researchers at few institutions.
- **Data.** SSL methods are often pretrained on a large-scale dataset (such as ILSVRC-2012 or even larger ones), and then fine-tuned in various downstream tasks. For a task where the total amount of available images (labeled or not) is limited (e.g., 100 categories with roughly 20 images per

category), it is unknown whether SSL will still be useful without a large pretraining dataset.

- **Flexibility.** This SSL paradigm (pretraining on big data followed by downstream fine-tuning) will sometimes become cumbersome. For instance, we need to train 10 different models for the same task, and deploy them to different hardware platforms [1], but it is impractical to pretrain 10 models on a large-scale dataset.

Self-supervised learning, however, dramatically differs from supervised learning. In SSL, why must we blindly inherit the setup from supervised learning? The most important difference is, of course, the presence or missing of image labels. As a direct consequence, we assume that *the information encoded by the contrastive loss is expected to be much less than that encoded in the labels via the cross entropy loss*, which is the fundamental assumption of this paper. To properly learn lesser information, we rethink how SSL should be carried out, and recommend the following changes to SSL:

- **Smaller resolution.** Fine-grained details in high resolution images may be unnecessary, or may even confuse the contrastive loss;
- **Directly perform SSL on the target domain**, even when there are only *a small set of training images*;
- **Partial backbone.** As will be further analyzed, removing the last residual block in the SSL pretrained model is helpful in improving accuracies for small data. We *do not* always need to train the full backbone model in SSL.

In short, in this paper we propose a new paradigm for training SSL models, which moves away from the supervised setup: use smaller resolution, fewer training data, and only part of the pretrained model. Because all these changes are scaled-down versions of the supervised learning setup, we call it *scaled-down self-supervised learning* (S3L). With S3L, obviously we can *greatly accelerate self-supervised learning*, thanks to the reduction in various dimensions. As Figure 1 shows, S3L also leads to *higher accuracy* on downstream tasks *using only small data from the target task*, which *leads to much higher flexibility*. S3L will be empirically verified by extensive experiments in this paper.

2. Related Works

To avoid time-consuming and expensive data annotations and to explore better feature representations, many self-supervised methods were proposed to learn visual representations from large-scale unlabeled images or videos. Generative approaches learn to generate or otherwise model pixels in the input space ([41, 23, 12]). Pretext-based approaches mainly explore the context features of images or videos such as context similarity [30, 11], spatial structure [15], clustering property [3], temporal structure [24], etc.

Unlike generative and pretext-based models, contrastive

learning is a discriminative approach that aims at grouping similar samples closer and diverse samples far from each other. Contrastive learning methods greatly improve the performance of representation learning, which has become the driving force of self-supervised representation learning in recent years ([17, 6, 39, 16, 4, 9]). Following MoCo [17], contrastive learning can be viewed as a dictionary lookup task. For each encoded query q , there is a set of encoded keys $\{k_0, k_1, \dots\}$, among which a single positive key k_+ matches the query q (generated from different views). A contrastive loss function InfoNCE [36] is employed to pull q close to k_+ while pushing it away from other negative keys:

$$L_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\exp(q \cdot k_+ / \tau) + \sum_{k_-} \exp(q \cdot k_- / \tau)}, \quad (1)$$

where τ denotes a temperature parameter. Both SimCLR [6] and MoCo are based on Equation (1). The main difference is that SimCLR samples negative pairs from the current batch while MoCo maintains a momentum memory bank. A more radical step is made by BYOL [16], which discards negative sampling in contrastive learning but achieves even better results in case a momentum encoder is used. Recently, follow-up work SimSiam [9] reports surprising results that simple siamese networks can learn meaningful representations even without the momentum encoder. SwAV [4] takes advantages of contrastive methods without computing pairwise comparisons by enforcing consistency between cluster assignments from different views.

However, they all suffer from heavy training costs because they train *the entire network* on *large-scale datasets* at *a large resolution* for *many epochs*, which is clearly inherited from the supervised learning settings. Recently, SEED [14] proposes to use self-supervised knowledge distillation for SSL with small models. However, it still follows this training paradigm using large resolution on large-scale datasets. In this paper, we argue that SSL should have a different learning paradigm and we aim to scale down SSL from three aspects mentioned above, i.e., resolution, model, and data.

3. Methods

Our fundamental assumption is that the information encoded inside the contrastive loss is much less than that encoded in the labels via the cross entropy loss. To adapt to the reduction in information, we make a paradigm shift from previous SSL methods, which mainly include 3 aspects:

- **Large (224) → small resolution (112 or even 56).** In the supervised setting, training with larger resolutions often yields better performance for classification [27], object detection [19] and semantic segmentation [5], in spite of much higher overhead. Also, with much more fine-grained labels, object detection and semantic segmentation often need much larger resolutions (800x600 or larger) than image classification (224x224) to get good results. However,

Table 1: Statistics of the 7 small datasets used in the paper.

Datasets	# Category	# Training	# Testing
CUB200 [37]	200	5994	5794
Cars [22]	196	8144	8041
Aircrafts [28]	100	6667	3333
Flowers [29]	102	2040	6149
Pets [31]	37	3680	3669
Dogs [21]	120	12000	8580
DTD [10]	47	3760	1880

given the very weak supervision via the contrastive loss, we expect an SSL model will *not learn image details*, and *a low-resolution input image is intuitively a better fit*. In the next section, we show that lower resolution in fact brings higher accuracy at much less training cost, which is *the opposite of the supervised learning situation*.

- **Entire** → **partial backbone**. We find that SSL fails to learn deep layers (e.g., conv5) well on small data, especially for large models. This phenomenon once again confirms our conjecture: the information encoded by the contrastive loss is limited and complex models suffer more from training on small data. Hence, we propose to only train shallow layers (e.g., conv1-conv4) during SSL pre-training and then train all layers during supervised fine-tuning. It greatly improves the accuracy with less cost.
- **Large** → **small data**. The key is to explore the power of small data with SSL. On one hand, SSL methods are often pretrained on large-scale datasets, which brings heavy training cost. On the other hand, the paradigm of pre-training followed by downstream fine-tuning will become cumbersome in some scenarios. As aforementioned, we cannot endure such a huge training cost to pretrain 10 different models on big data. Hence, how to directly perform SSL on the target small datasets is a valuable and interesting problem.

Combining the 3 aspects above, our scaled-down self-supervised learning (S3L) framework successfully explore the power of small data with existing SSL methods, which not only greatly accelerates the training process but also achieves higher accuracy, as shown in the next section.

4. Experimental Results

We used 7 small datasets for our experiments, as shown in Table 1. First, we demonstrate the effectiveness of small resolution on small datasets as well as the large-scale ImageNet [33] in Section 4.1. Then, we investigate the effect of removing the last residual block in Section 4.2. Finally, we explore the power of small data in Section 4.3. All our experiments were conducted using PyTorch and we used Titan Xp GPUs for ImageNet experiments and Tesla K80 GPUs for small datasets. Codes will be made publicly available.

4.1. Small resolution is beautiful

We first investigate the effectiveness and efficiency of small resolution on small datasets in Section 4.1.1. Then, we demonstrate that small resolution is also useful for the large-scale dataset ImageNet (IN) in Section 4.1.2.

4.1.1 Results on CUB200 and other small datasets

We carefully compare the influence of various input resolutions during SSL pretraining using 3 typical SSL methods, namely MoCov2 [8], SimCLR [6] and BYOL [16] under both ResNet-18 and ResNet-50 [20]. The full learning process contains two stages: pretraining and fine-tuning. We use the pretrained weights obtained by SSL for initialization and then fine-tune networks for classification using the cross entropy loss. Note that *SSL pretraining and fine-tuning are both performed only on the target dataset*.

For the fine-tuning stage, we fine-tune all methods for 120 epochs using SGD with a batch size of 64, a momentum of 0.9 and a weight decay of $5e-4$ for fair comparisons. For the ImageNet supervised setting, the learning rate (lr) is initialized to 0.01, which is divided by 10 every 40 epochs following [2]. For other methods, we initialize the lr to 0.1 and use the cosine learning rate decay. We also list the results using the mixup [40] strategy, where alpha is set to 1.0. For the SSL pretraining stage, we follow the same settings in the original papers and more details are included in the appendix. Experimental results are shown in Table 2 (and part of the results are visualized in Figure 1).

Notice that we use the same batch size in Table 2 and Table 4 for all input resolutions for fair comparisons. However, we know that small resolutions require much fewer GPU memories and thus enable larger batch sizes, hence we also investigate the influence of input resolution and batch size in Table 3. From these results, we have the following observations:

- **SSL pretraining is useful** and SimCLR yields the best performance among the 3 SSL methods on CUB200. MoCov2, SimCLR and BYOL all achieve much higher accuracies than random initialization when fine-tuned for 120 epochs (and similar results are obtained in Table 4 when fine-tuned for more epochs). ‘SimCLR 800ep (56)’ (800 epochs pretraining with 56x56 input resolution using SimCLR) achieves the highest accuracy (71.0%) for ResNet-18 and ‘SimCLR 800ep (112)’ achieves the highest accuracy (74.5%) for ResNet-50.
- **Small resolution achieves better performance with much fewer training cost** using various SSL methods and backbone networks. Take SimCLR 800ep as an example, 56x56 resolution achieves 5.5% relative higher accuracy (71.0 v.s. 67.3) and 58.8% relative fewer training hours (3.5 v.s. 8.5) than the baseline 224x224 resolution under ResNet-18.

Table 2: Comparisons of pretraining details, total time (GPU hours using 2 Tesla K80s) and accuracy (%) on CUB200. All are fine-tuned for 120 epochs for fair comparisons. ‘N/A’ means that we didn’t conduct ImageNet pretraining on K80 GPUs.

Backbone	pretraining				Accuracy		Total time	
	method	resolution	#FLOPS	epochs	time	Normal		Mixup
ResNet-18	ImageNet supervised				N/A	76.2	75.0	N/A
	random initialize				0.0	62.0	63.4	1.1
	MoCov2	224	1824.54M	200	1.6	63.7	65.8	2.7
				800	6.4	65.0	66.3	7.5
		112	488.40M	200	0.9	64.2	65.4	2.0
				800	3.6	66.2	67.4	4.7
		112→224	755.63M	800→200	5.2	66.4	68.4	6.3
		56	130.75M	200	0.7	63.2	64.6	1.8
	800			2.8	66.1	67.5	3.7	
	56→112	202.28M	800→200	3.7	66.0	68.8	4.8	
	56→112→224	295.95M	800→200→100	4.5	66.2	69.3	5.6	
	SimCLR	224	1824.54M	200	1.8	63.6	64.5	2.9
				800	6.4	66.0	67.3	8.5
		112	488.40M	200	0.8	64.8	67.9	1.9
				800	3.2	67.9	69.2	4.3
		56	130.75M	200	0.6	65.7	68.9	1.7
				800	2.4	68.1	71.0	3.5
	BYOL	224	1824.54M	200	2.0	63.2	66.0	3.1
				800	8.0	65.3	68.6	9.1
		112	488.40M	200	0.9	64.9	65.0	2.0
800				3.7	66.3	70.3	4.8	
56		130.75M	200	0.6	64.0	67.5	1.7	
			800	2.4	67.2	70.0	3.5	
ResNet-50	ImageNet supervised				N/A	81.3	82.1	N/A
	MoCov2 IN 800ep				N/A	77.7	77.9	N/A
	random initialize				0.0	58.6	56.3	2.1
	MoCov2	224	4135.79M	800	11.4	66.5	62.0	13.5
				1200	17.2	69.0	72.4	19.3
		112	1091.26M	800	5.5	67.0	71.0	7.6
				1200	8.3	68.9	74.0	10.4
		112→224	1700.17M	800→200	8.4	68.4	72.3	10.5
		56	304.06M	800	3.5	66.2	70.2	5.6
	1200			5.3	68.0	72.3	7.4	
	56→112	461.50M	800→200	8.4	69.1	72.6	10.5	
	56→112→224	673.14M	800→200→100	11.3	69.8	72.7	13.4	
	SimCLR	224	4135.79M	200	3.7	68.0	66.5	5.8
				800	14.8	69.2	73.0	16.9
		112	1091.26M	200	1.5	65.3	69.8	3.6
				800	6.0	71.2	74.5	8.1
		56	304.06M	200	0.8	68.0	70.9	2.9
				800	3.0	71.5	74.2	5.1
	BYOL	224	4135.79M	200	5.2	59.4	57.7	7.3
				800	20.8	62.4	64.1	22.9
112		1091.26M	200	2.1	60.4	62.7	4.2	
			800	8.4	63.3	67.7	10.5	
56		304.06M	200	1.3	63.0	65.8	3.4	
			800	5.0	64.1	68.1	7.1	

- **Gradual transition from small to large resolution is effective** to train with large resolution for SSL methods. In Table 2, we design a multi-stage pretraining strategy (gradually from small to large resolution). For example, in the ‘56→112→224’ setting, the SSL pretraining process contains 3 stages: (1) train the network for 800 epochs with 56x56 input resolution; (2) then, train for 200 epochs with 112x112 resolution initialized with the weights obtained in the first stage; (3) finally, train for 100 epochs with 224x224 resolution initialized with the weights obtained in the second stage. This multi-stage training strategy achieves higher accuracy than the baseline ‘224’ setting with fewer training hours, although the final pretraining resolution are both 224x224. It indicates that directly train-

ing with a large resolution is harmful on small datasets while a gradual transition from small to large resolution is promising to train with large resolution for SSL methods.

- As shown in Table 3, **a smaller resolution enables a larger batch size**, which will further improve the performance when compared to the results in Table 2 where we used the same batch size for all resolutions. Take ResNet-50 as an example, the maximum batch size is 128 with 224x224 resolution using 2 K80 GPUs and a batch size of 512 will incur the ‘out of memory’ problem. This problem will not appear when small resolutions are used. It has already been shown that larger batch size brings higher accuracy for SimCLR in [6], hence we use BYOL here

Table 3: Smaller resolution enables larger batch sizes (bs). All methods are trained using BYOL and ResNet-50 for 800 epochs on 2 K80 GPUs on CUB200. ‘N/A’ indicates the 224 resolution incurs *out of memory* with batch size being 512.

bs	resolution	time	Accuracy	
			Normal	Mixup
128	224	20.8	62.4	64.1
	112	8.3	63.3	67.7
	56	5.0	64.1	68.1
512	224	N/A	N/A	N/A
	112	7.0	67.0	71.8
	56	3.7	66.8	70.5

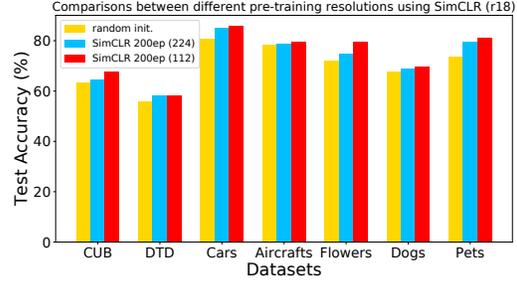
Table 4: Results on CUB200 with more fine-tuning (FT) epochs. ‘(112 FT)’ means fine-tuning with 112x112 resolution. ‘SimCLR 800ep (112)’ means 800 epochs pretraining with 112x112 resolution under SimCLR.

pretraining method	FT epochs	ResNet-18		ResNet-50	
		acc.	time	acc.	time
IN supervised	120	75.0	N/A	82.1	N/A
random init. (112 FT)	120	46.0	0.7	44.2	1.0
	480	54.3	2.8	58.3	4.0
random init.	120	63.4	1.1	56.3	2.1
	480	69.2	4.4	72.4	8.4
	1200	72.5	11.0	76.5	21.0
	1600	72.1	14.7	77.2	28.8
MoCov2 800ep (224)	480	72.5	10.8	76.5	19.8
MoCov2 800ep (112)	480	73.6	8.0	77.9	13.9
MoCov2 800ep (56)→200ep (112)→100ep (224)	480	74.2	9.9	78.4	19.7
SimCLR 800ep (224)	480	73.2	10.8	77.4	23.2
SimCLR 800ep (112)	480	74.1	7.6	79.5	14.4
SimCLR 800ep (56)	480	75.8	6.8	79.4	11.4
BYOL 800ep (224)	480	73.1	12.4	73.0	23.2
BYOL 800ep (112)	480	73.8	8.1	75.6	16.8
BYOL 800ep (56)	480	74.7	6.8	76.0	13.4

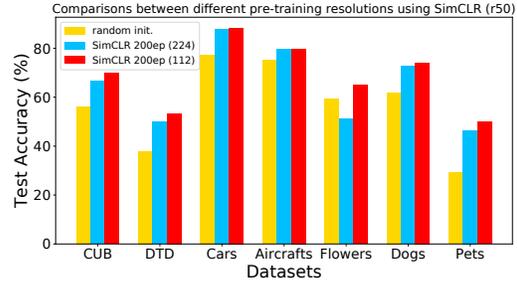
for illustration in Table 3. We can observe that 112x112 resolution have 12% relative higher accuracy (71.8 v.s. 64.1) than the baseline 224x224 resolution, using only one-third of the time (7.0 v.s. 20.8).

As known in [18], more training epochs is essential when training from scratch for object detection. Hence, we also investigate the effect of more fine-tuning epochs in Table 4 and we can have the following conclusions:

- **More epochs is essential when training from scratch on small datasets** for image classification. When randomly initialized, ResNet-18 achieves the highest accuracy of 75.8% (1200 epochs) and the accuracy will not continue to improve with more epochs, and ResNet-50 achieves the highest accuracy of 77.2% (1600 epochs).
- **Small resolution still has consistent improvements** compared to large resolution when fine-tuning for more epochs. When compared with the best performance of random initialization, ‘SimCLR 800ep (56)’ achieves 3.3%



(a) ResNet-18



(b) ResNet-50

Figure 2: Comparisons between different resolutions during pretraining under SimCLR on 7 small datasets.

higher accuracy for ResNet-18 and 2.2% higher accuracy for ResNet-50 with less than half of the training time.

- **Small resolution is useful for SSL but not for supervised learning.** When using the 112x112 resolution for supervised fine-tuning, the accuracy is much lower than using the 224x224 resolution (e.g., 54.3 v.s. 69.2 for ResNet-18 when fine-tuned for 480 epochs). In contrast, small resolution achieves much higher accuracy than large resolution for SSL. It indicates that SSL has limited information and hence fine-grained details in high resolution images may be unnecessary, or may even confuse the contrastive loss. In short, small resolution makes SSL easier to learn and it is not the case when we have supervised information (i.e., the labels).
- ‘SimCLR 800ep (112)’ achieves higher accuracy than ‘MoCov2 IN 800ep’ (79.5 v.s. 77.9) when fine-tuned for 480 epochs under ResNet-50. It indicates that **directly performing SSL on the target domain is promising**, even when there are only a small set of training images. Note that the latter is pretrained on ImageNet.

We experimented with all the datasets in Table 1 and more results are shown in Figure 2 (c.f. appendix for precise values). We can find consistent improvements of small resolution on all these datasets with much less training costs.

Moreover, we visualize the feature maps using Grad-CAM [34] to better understand the learned representations with different pretraining resolutions in the appendix. Small resolution also achieves better localization performance.

Table 5: Object detection and instance segmentation fine-tuned on COCO: bounding-box AP (AP^{bb}) and mask AP (AP^{mk}) evaluated on val2017. Pretraining time (GPU hours) was counted using 8 Titan XP GPUs.

pretraining		AP^{bb}	AP_{50}^{bb}	AP_{75}^{bb}	AP^{mk}	AP_{50}^{mk}	AP_{75}^{mk}	AP^{bb}	AP_{50}^{bb}	AP_{75}^{bb}	AP^{mk}	AP_{50}^{mk}	AP_{75}^{mk}
method	time												
random init.	0	31.0	49.5	33.2	28.5	46.8	30.4	36.7	56.7	40.0	33.7	53.8	35.9
IN supervised	27.8	38.4	59.2	41.6	35.0	55.9	37.1	40.6	61.3	44.4	36.8	58.1	39.5
MoCov2 200ep (224)	83.6	39.0	59.5	42.4	35.6	56.6	38.0	41.2	61.8	45.1	37.6	58.9	40.3
MoCov2 200ep (112)	47.3	38.9	59.3	42.6	35.2	56.4	37.7	41.0	61.7	45.0	37.2	58.8	39.9
200ep (112)→50ep (224)	68.2	38.9	59.8	42.9	35.8	56.9	38.3	41.2	61.9	45.3	37.6	59.2	40.3
MoCov2 800ep (224)	334.4	39.5	59.8	43.2	36.0	56.9	38.6	41.5	62.0	45.7	37.6	58.9	40.3
MoCov2 800ep (112)	189.2	39.3	59.8	42.9	35.8	56.9	38.3	41.4	62.2	45.3	37.6	59.2	40.3
800ep (112)→200ep (224)	272.8	39.5	60.1	43.3	35.9	57.1	38.6	41.5	62.1	45.2	37.7	59.3	40.4

(a) R50-FPN (1x)

(b) R50-FPN (2x)

Table 6: Transfer learning results from ImageNet with standard ResNet-50 architecture.

Method	ImageNet	VOC2007	CUB200	Cars	Aircrafts	CIFAR10	CIFAR100	Caltech-101	Flowers	Dogs	DTD
<i>Linear evaluation:</i>											
MoCov2 200ep (224)	67.7	80.6	17.8	14.1	12.3	56.4	26.0	80.8	68.5	42.1	64.9
MoCov2 200ep (112)	65.3	76.4	11.5	10.8	9.5	50.1	18.4	73.4	64.6	28.3	60.1
MoCov2 200ep (112)→50ep (224)	66.7	81.2	18.8	16.2	14.7	57.8	25.5	81.8	73.5	42.9	65.1
MoCov2 800ep (224)	71.1	82.8	17.5	13.4	11.8	56.8	23.6	82.1	67.8	46.4	65.2
MoCov2 800ep (112)	68.4	79.0	11.0	9.8	8.2	53.1	21.4	72.9	56.9	36.4	61.4
MoCov2 800ep (112)→200ep (224)	69.8	82.9	18.6	14.6	12.6	59.2	27.8	82.4	72.4	46.7	65.4
IN supervised	-	73.9	61.7	47.1	23.7	58.0	27.3	89.1	86.9	82.2	68.2
<i>Fine-tuned:</i>											
MoCov2 200ep (224)	73.9	85.6	75.5	89.2	86.5	89.7	65.0	89.2	95.7	76.6	68.6
MoCov2 200ep (112)	73.6	86.0	76.4	89.1	86.3	90.3	67.4	89.7	94.8	76.5	68.2
MoCov2 200ep (112)→50ep (224)	73.9	86.2	77.5	88.7	86.7	89.4	65.2	91.0	95.9	77.6	70.2
MoCov2 800ep (224)	75.3	87.4	77.7	89.9	87.5	89.2	65.3	89.5	95.4	77.7	67.7
MoCov2 800ep (112)	75.0	87.6	77.7	88.3	86.7	91.1	67.7	90.2	95.7	77.2	68.2
MoCov2 800ep (112)→200ep (224)	75.3	86.2	78.8	89.1	87.7	90.1	66.7	91.4	96.0	77.7	69.6
IN supervised	76.1	89.0	81.3	90.6	86.7	90.0	67.0	94.1	96.7	80.1	74.7

Table 7: Object detection fine-tuned on PASCAL VOC train-val07+12. Evaluation is on test2007: AP_{50} (default VOC metric), AP (COCO-style), and AP_{75} . All are fine-tuned for 24k iterations (~23 epochs).

pretraining		R-50 FPN			R-50 C4		
method	time	AP_{50}	AP	AP_{75}	AP_{50}	AP	AP_{75}
random init.	0.0	63.0	36.7	36.9	60.2	33.8	33.1
IN supervised	27.8	80.8	53.5	58.4	81.3	53.5	58.8
MoCov2 200ep (224)	83.6	81.8	55.0	60.5	82.2	57.1	64.5
MoCov2 200ep (112)	47.3	81.3	54.1	59.5	82.1	56.8	63.1
200ep (112)→50ep (224)	68.2	81.7	54.8	60.4	82.2	56.6	63.5
MoCov2 800ep (224)	334.4	81.5	55.0	61.0	82.6	57.7	64.5
MoCov2 800ep (112)	189.2	81.2	54.3	61.2	82.4	57.2	63.9
800ep (112)→200ep (224)	272.8	81.7	55.4	61.7	82.5	57.7	64.4

4.1.2 Results on ImageNet

Now we have shown that small resolution is beautiful for small datasets, and move on to investigating the effect of small resolution for SSL on the large-scale dataset ImageNet. We use MoCov2 for illustration following the official training and evaluation protocol in [8]. We carefully investigate the downstream object detection performance on COCO2017 [26] in Table 5 and Pascal VOC07&12 [13] in Table 7, as well as downstream classification performance on 10 datasets in Table 6. The detector is Faster R-CNN [32]

with a backbone of R50-FPN [25] or R50-C4 [19] for Pascal VOC object detection and Mask R-CNN [19] with R50-FPN backbone for COCO, implemented in [38]. For ImageNet linear evaluation, we follow the same settings in [8]. For ImageNet fine-tuning, we train for 30 epochs with the learning rate initialized to 0.01, which is divided by 10 every 10 epochs. For other classification benchmarks, we train the network for 120 epochs with a batch size of 64 and a weight decay of $5e-4$. The learning rate starts from 10.0 for linear evaluation and 0.01 for fine-tuning and is decreased every 40 epochs.

For detection, our ‘800ep (112)→200ep (224)’ strategy (c.f. Sec. 4.1.1) has comparable accuracy as the baseline ‘800ep (224)’ setting on both Pascal VOC and COCO2017, but using 61.6 fewer training hours. Notice that the 112 resolution reduces the training time by nearly a half although it does not get as much improvements as before in small datasets when compared to the 224 resolution.

For image classification, our method achieves lower accuracy than the baseline method for ImageNet linear evaluation, which is a popular benchmark in previous works. However, *it may not be an appropriate indicator* for our method, because our ‘800ep (112)→200ep (224)’ achieves higher linear evaluation accuracies than baseline ‘800ep (224)’ on all the 10 downstream classification datasets. Moreover, ‘800ep

Table 8: The effect of removing conv5 on 3 small datasets. We count the extra training time of warmup epochs in total time.

Backbone	pretraining				Accuracy			Total time	
	resolution	setting	#FLOPs	epochs	time	CUB200	Pets		Flowers
ResNet-18	224	baseline	1824.54M	200	1.6	65.4	74.2	76.1	2.7
				800	6.4	66.3	76.9	82.7	7.5
		drop conv5 weights	1824.54M	200	1.6	66.2	77.3	79.6	2.8
				800	6.4	68.9	77.4	83.2	7.6
		remove conv5	1412.51M	200	1.4	68.7	76.9	76.9	2.6
				800	5.4	68.7	78.5	82.9	6.6
	112	baseline	488.40M	200	0.9	65.8	75.2	77.2	2.0
				800	3.6	67.4	77.7	82.8	4.7
		drop conv5 weights	488.40M	200	0.9	66.8	76.6	79.6	2.1
800	3.6	68.1	77.3	83.2	4.8				
remove conv5	353.51M	200	0.8	69.6	77.1	77.6	2.0		
800	3.4	70.3	79.5	84.0	4.6				
ResNet-50	224	baseline	4135.79M	200	2.9	53.0	48.0	58.2	5.0
				800	11.4	62.0	61.1	65.6	13.5
		drop conv5 weights	4135.79M	200	2.9	71.1	80.2	73.9	5.2
				800	11.4	71.2	80.9	78.2	13.7
		remove conv5	3332.09M	200	2.4	70.7	78.6	76.9	4.7
				800	9.6	72.7	81.1	82.3	11.9
	112	baseline	1091.26M	200	1.4	54.0	48.6	64.0	3.5
				800	5.5	71.0	61.3	67.4	7.6
		drop conv5 weights	1091.26M	200	1.4	69.9	78.3	74.4	3.7
				800	5.5	72.2	81.1	81.6	7.8
		remove conv5	832.06M	200	1.2	71.3	80.1	78.1	3.4
				800	4.8	72.2	81.8	83.0	7.1

(112)→200ep (224)’ achieves higher fine-tuning accuracies than baseline ‘800ep (224)’ on 7 out of the 10 downstream datasets and the same accuracy when fine-tuning on ImageNet. *Linear evaluation may not be a good SSL evaluator.*

4.2. Small architecture is useful

Furthermore, now we show that small architecture is useful and removing the last residual block in the SSL pretrained model is in fact helpful in improving accuracies. In Table 8, we compare 3 strategies on 3 datasets to investigate the effect of removing the last residual block during SSL pretraining:

(a) baseline: we pretrain the whole network using MoCov2 and then fine-tune for 120 epochs as before.

(b) remove conv5: During the SSL pretraining process, we remove the last residual block in the ResNet (namely conv5) and only keep conv1-conv4 for pretraining. Then, during the fine-tuning process, following the previous practices for newly added modules in [27], we freeze the conv1-conv4 blocks and warmup the conv5 block for 10 epochs with learning rate 0.1 using the supervised cross entropy loss. After that, we fine-tune all the network for 120 epochs as before. We list the total training time in Table 8, which includes the minor extra training time for the warmup epochs.

(c) drop conv5 weights: we pretrain the whole network, and then drop the conv5 weights (i.e., randomly re-initialize these weights) during the fine-tuning process. We warmup the conv5 block for 10 epochs before fine-tuning the whole network for 120 epochs as in (b) for fair comparisons.

From Table 8, we have the following observations:

- **Large models suffer more from training on small data** than small models. When comparing the baseline setting,

ResNet-18 achieves much higher accuracy than ResNet-50 under the same pretraining and fine-tuning epochs. It indicates that large models are much easier to overfit on small data and they suffer from learning complex parameters with limited data. Moreover, MoCov2 even achieves lower accuracy than random initialization when pretrained for 200 epochs under ResNet-50 and the situation is alleviated when pretrained for more epochs, which further demonstrates the difficulty of learning on small data with large models.

- **The SSL pretrained model fails to learn deep layers well on small data**, but ‘drop conv5 weights’ and ‘remove conv5’ are both useful, especially for the large model ResNet-50. Take ResNet-50 800ep (224) as an example, our ‘remove conv5’ strategy achieve 17.3%, 32.7% and 25.5% relative higher accuracies than the baseline strategy on CUB200, pets and flowers, respectively.
- As is well known in previous works [6, 8], the MLP head is essential to separate representation learning from learning specific properties of the contrastive loss. Note that the ‘drop conv5 weights’ strategy works well and hence there is a possibility that conv5 is also fitting to certain properties of the contrastive loss rather than generic image properties. But, the ‘remove conv5’ strategy also works well and achieves even better performance than the ‘drop conv5 weights’ strategy with less training cost. Hence, it indicates that the underlying reason is **the lack of capability to learn complex conv5 layers well with small data for SSL models.**

Table 9: Classification results on small datasets. ‘448 fine-tune’ means fine-tuning with 448x448 input resolution.

Backbone	Method	Extra data	CUB200	Cars	Aircrafts	Flowers	Pets	DTD	Dogs
ResNet-18	random init.	×	72.1	88.1	82.9	86.9	84.5	60.1	68.3
	IN super.	✓	76.2	88.3	81.2	95.6	90.8	68.9	76.5
	S3L (ours)	×	75.8	90.1	89.0	91.4	86.4	63.1	70.7
	IN super. 448 fine-tune	✓	81.3	92.0	86.9	95.6	92.0	70.8	79.8
	S3L 448 fine-tune (ours)	×	80.1	92.9	91.0	92.7	88.0	67.4	76.1
ResNet-50	random init.	×	77.2	88.9	87.5	86.6	81.8	55.6	69.1
	IN super.	✓	81.3	90.6	86.7	96.7	91.5	74.7	80.1
	MoCov2 IN 800ep	✓	77.7	89.9	87.5	95.4	88.8	67.7	77.7
	S3L (ours)	×	79.5	91.9	90.6	91.7	88.7	63.4	73.8
	IN super. 448 fine-tune	✓	84.5	93.2	91.0	97.0	93.3	75.5	83.4
S3L 448 fine-tune (ours)	×	83.8	93.4	92.3	93.4	89.3	66.9	78.1	

Table 10: Downstream object detection performance when pretrained on small ImageNet under ResNet-50 backbone. Other metrics on COCO are included in the appendix.

pretraining			VOC 07&12			COCO 2017	
method	#images	epochs	AP ₅₀	AP	AP ₇₅	AP ^{bb}	AP ^{mk}
random init.	0	0	63.0	36.7	36.9	31.0	28.5
supervised	1.28M	100	80.8	53.5	58.4	38.4	35.0
		10000	55.9	31.2	30.8	28.2	26.2
	50000	2000	55.7	29.5	27.0	27.5	25.7
		800	68.4	39.3	39.3	31.6	29.1
MoCov2 (112)	1.28M	200	81.3	54.1	59.5	38.9	35.2
		10000	75.1	47.1	50.3	35.2	32.3
	50000	8000	76.5	48.4	51.9	35.6	32.5
		800	78.5	51.0	55.4	36.7	33.5
		4000	79.0	51.4	56.1	37.3	34.1

4.3. Small data is powerful

Finally, we show that small data alone is also powerful: we can achieve impressive results by directly training on small datasets without the need of pretraining on large-scale datasets (e.g., ImageNet), by combining the two strategies mentioned above.

We conducted experiments on the 7 small datasets in Table 9. For the ImageNet supervised setting, we follow the training protocols as before. For our S3L method, we pretrain SimCLR for 800~1600 epochs by combining both small resolution and small architecture strategies and fine-tune the whole model for 480 epochs with lr initialized to 0.1. For random initialization, we train the network for longer epochs (over 800 epochs) and report the best results.

Table 9 shows that our S3L method has achieved impressive results on these small datasets without using any extra data. S3L even achieves higher accuracy on Cars and Aircraft than the models pretrained on ImageNet (supervised or SSL), which indicates that our method is very effective to handle small data and directly training with small data is a very promising direction. Also, we achieve the state-of-the-art results on CUB200, Cars, Aircraft and pets when training from scratch to the best of our knowledge.

However, the results on DTD is still far from the ImageNet supervised baseline, especially for ResNet-50. Note

that DTD is a texture dataset, and textures have an important property called the self-similarity [35], which means that images of the same category have highly similar internal structures and textures. This property contradicts the contrastive loss which requires an image to be similar to itself and dissimilar to others.

Another interesting thing is to investigate the performance of supervised learning and SSL when using small number of images for ImageNet (small ImageNet). We randomly sample 10000 and 50000 images to construct small ImageNet. As shown in Table 10, we surprisingly find that:

- MoCov2 (112) can learn representations well and get impressive results on downstream object detection tasks even with only 10000 images. In comparison, supervised learning fails to learn meaningful representations because they are much easier to overfit on small ImageNet. It shows that SSL is useful and essential for small data, and we can get a good starting point with SSL on small data.
- More epochs pretraining are beneficial for SSL on small ImageNet but not for supervised learning. When training on 10000 images, the performance on downstream tasks for supervised learning even decreases when training for more epochs, which indicates that supervised learning suffers more from the overfit problem on small data.
- SSL may not need as much images as supervised learning because it gets much better results than supervised baseline with only a small number of training data on ImageNet. It once again verifies our motivation: we need to scale-down from various aspects for SSL.

5. Conclusion

In this paper, we proposed a shift from the existing self-supervised learning paradigm to scaled-down self-supervised learning (S3L) from 3 aspects: small resolution, small architecture and small data. Various experiments show that our method obtained a significant edge over the previous learning paradigm with much less training cost, especially on small datasets. Moreover, we achieved impressive results by directly learning on small data without any extra datasets using our S3L method, which shows that our S3L is both

effective and efficient, and that learning with only small data is a promising and valuable direction.

In the future, we will explore two directions. First, we will dive into deep learning with only small data. Second, we will investigate more effective and efficient methods for self-supervised learning.

References

- [1] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *The International Conference on Learning Representations*, pages 1–14, 2020. 2
- [2] Yun-Hao Cao, Jianxin Wu, Hanchen Wang, and Joan Lasenby. Neural random subspace. *Pattern Recognition*, 112:Article 107801, 2021. 3
- [3] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *The European Conference on Computer Vision*, volume 11218 of *LNCS*, pages 132–149. Springer, 2018. 2
- [4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Advances in neural information processing systems*, pages 9912–9924, 2020. 2
- [5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 2
- [6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *The International Conference on Machine Learning*, pages 1597–1607, 2020. 1, 2, 3, 4, 7
- [7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. In *Advances in neural information processing systems*, pages 22243–22255, 2020. 1
- [8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 3, 6, 7
- [9] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 2
- [10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014. 3
- [11] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representations learning by context prediction. In *The IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2
- [12] Jeff Donahue, Philipp Krahenbuhl, and Trevor Darrell. Adversarial feature learning. In *The International Conference on Learning Representations*, pages 1–12, 2017. 2
- [13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 6
- [14] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. SEED: Self-supervised distillation for visual representation. In *The International Conference on Learning Representations*, pages 1–12, 2021. 2
- [15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *The International Conference on Learning Representations*, pages 1–14, 2015. 2
- [16] Jean-Bastien Grill, Florian Strub, Florent Altche, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bial Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in neural information processing systems*, pages 21271–21284, 2020. 1, 2, 3
- [17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2
- [18] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet pre-training. In *The IEEE International Conference on Computer Vision*, pages 4918–4927, 2019. 5
- [19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *The IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 2, 6
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 3
- [21] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011. 3
- [22] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013. 3
- [23] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 4681–4690, 2017. 2
- [24] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *The IEEE International Conference on Computer Vision*, pages 667–676, 2017. 2
- [25] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 2177–2125, 2017. 6

- [26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *The European Conference on Computer Vision*, volume 8693 of *LNCS*, pages 740–755. Springer, 2014. 6
- [27] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *The IEEE International Conference on Computer Vision*, pages 1449–1457, 2015. 2, 7
- [28] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 3
- [29] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 1447–1454, 2006. 3
- [30] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *The European Conference on Computer Vision*, volume 9910 of *LNCS*, pages 69–84. Springer, 2016. 2
- [31] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. 3
- [32] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 6
- [33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 3
- [34] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision*, pages 618–626, 2017. 5
- [35] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *The IEEE International Conference on Computer Vision*, pages 1–8, 2007. 8
- [36] Aarin van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2
- [37] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 1, 3
- [38] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019. 6
- [39] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *The IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2
- [40] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *The International Conference on Learning Representations*, pages 1–13, 2018. 3
- [41] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *The European Conference on Computer Vision*, volume 9907 of *LNCS*, pages 649–666. Springer, 2016. 2

Table 11: Training details for MoCov2, SimCLR and BYOL on CUB200 for experiments presented in Table 2. τ denotes the temperature parameter and k denotes the size of the memory bank in MoCov2.

Method	backbone	Settings				
		bs	lr	lr schedule	τ	k
MoCov2	ResNet-18	128	0.03	cosine	0.2	4096
	ResNet-50	128	0.03	cosine	0.2	4096
SimCLR	ResNet-18	512	0.5	cosine	0.1	-
	ResNet-50	128	0.125	cosine	0.1	-
BYOL	ResNet-18	512	0.5	cosine	-	-
	ResNet-50	128	0.125	cosine	-	-

A. Training details

A.1. SSL settings

The training details for MoCov2, SimCLR and BYOL on CUB200 for those experimental results presented in Table 2 in the main paper are shown in Table 11.

A.2. Data augmentations

For SSL pre-training, we follow the data augmentation setting in SimCLR for all the 3 SSL methods, including Gaussian blur, color distortion, random horizontal flip, random resized crop, etc. For the ‘224’ setting, we crop 224x224 patches following previous works. For the ‘112’ (or ‘56’) setting, we crop 112x112 (or 56x56) patches, respectively, and other transformations remain the same.

For supervised fine-tuning, the images are resized with shorter side=256, then a 224 × 224 crop is randomly sampled from the resized image with horizontal flip and mean-std normalization.

B. Localization results and visualization

To better understand the difference of the final learned feature representations between different input resolutions during SSL pretraining, we visualize the feature maps using Grad-CAM and evaluate the localization performance on CUB200. Following previous works, we evaluate the fine-tuned models because Grad-CAM depends on the classification head. *GT-Known* Loc is correct when given the ground truth class label to the model, the intersection over union (IoU) between the ground truth bounding box and the predicted box is 50% or more. The localization results are shown in Table 12 and visualization results are shown in Figure 3.

As can be seen in Table 12, small resolution not only achieves higher classification accuracy but also achieves better localization performance. Note that we use *GT known* Loc for evaluation, which is irrelevant to classification performance, hence the higher localization accuracy directly comes from better feature representations.

Table 12: Localization results using Grad-CAM on CUB200 under ResNet-50 with different SSL pretraining resolutions.

Methods	Cls acc.	<i>GT known</i> Loc acc.	time
SimCLR 800ep (224)	69.2	55.9	16.9
SimCLR 800ep (112)	71.2	57.5	8.1
SimCLR 800ep (56)	71.5	57.7	5.1

From Figure 3, we can find that SimCLR (224) is sometimes confused by complex backgrounds (column (a) and (l)) and often localizes only the most discriminative part of an object in an image (column (c), (g), (h) and (i)). In contrast, SimCLR (112) achieves better localization results and interestingly in column (e), it successfully localizes two wings as well as the head.

C. More results

The precise results of the influence of pretraining resolutions on 7 small datasets (Figure 2 in the paper) are shown in Table 13. As can be seen, small resolution achieves consistent improvements on all the 7 small datasets under both ResNet-18 and ResNet-50.

Also, we didn’t list all metrics for COCO in Table 10 in Section 4.3 in the paper due to limited space, and the detailed results are shown in Table 14.

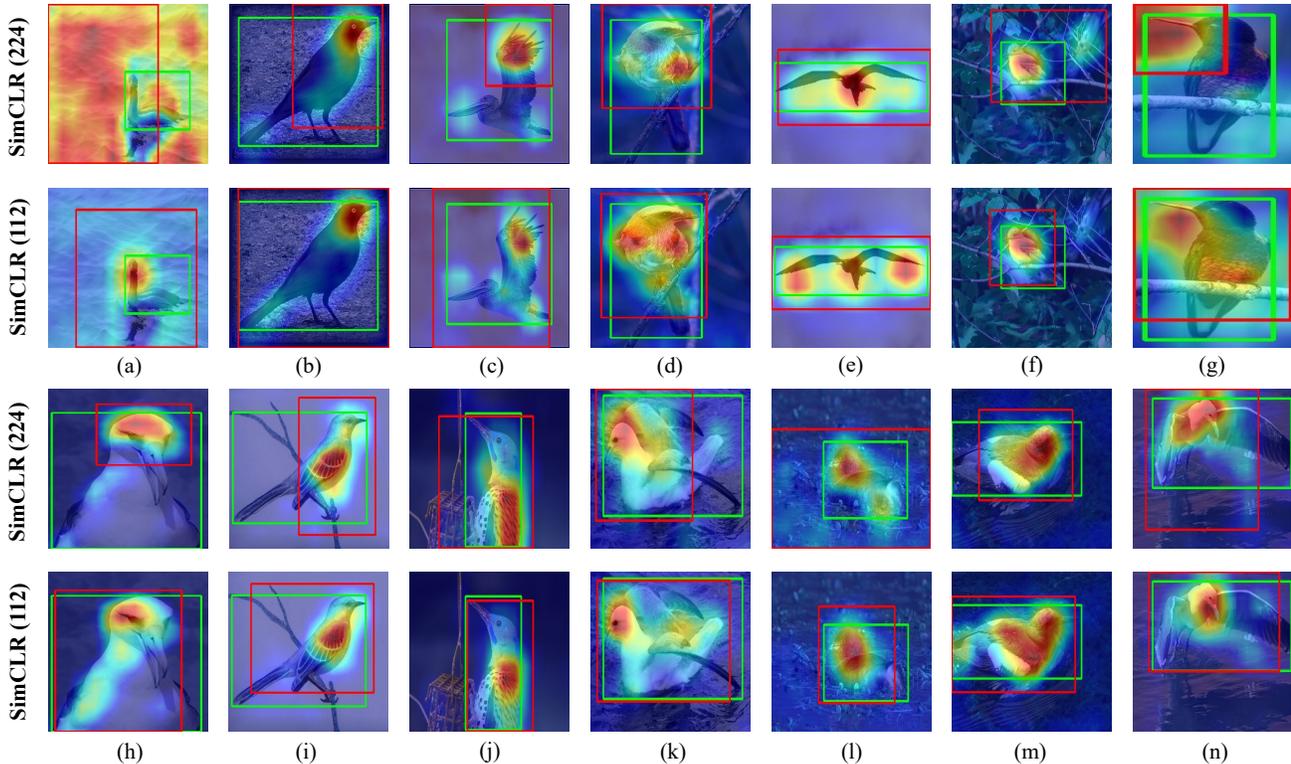


Figure 3: Comparisons of activation maps visualization between SimCLR (224) and SimCLR (112) using Grad-CAM on CUB200. Note that red boxes are Grad-CAM predicted boxes and green boxes are groundtruth boxes. This figure is best viewed in color and zoomed in.

Table 13: Ablation studies of different input resolutions on the 7 small datasets. All fine-tuned for 120 epochs.

Backbone	Method	CUB	Cars	Aircraft	Flowers	Pets	DTD	Dogs
ResNet-18	random init.	63.4	80.9	78.6	72.2	73.7	56.0	67.7
	SimCLR 200ep (224)	64.5	85.0	79.0	74.9	79.8	58.1	69.0
	SimCLR 200ep (112)	67.9	85.8	79.7	79.6	81.1	58.4	69.6
ResNet-50	random init.	56.3	77.3	75.2	59.3	29.1	37.9	61.9
	SimCLR 200ep (224)	66.5	87.9	79.6	51.2	46.4	50.0	72.6
	SimCLR 200ep (112)	69.8	88.3	79.8	65.0	50.0	53.4	74.1

Table 14: Downstream object detection performance when pretrained on small ImageNet under ResNet-50 backbone.

pretraining			VOC 07&12			COCO 2017					
method	#images	epochs	AP ₅₀	AP	AP ₇₅	AP ^{bb}	AP ₅₀ ^{bb}	AP ₇₅ ^{bb}	AP ^{mk}	AP ₅₀ ^{mk}	AP ₇₅ ^{mk}
random init.	0	0	63.0	36.7	36.9	31.0	49.5	33.2	28.5	46.8	30.4
supervised	1.28M	100	80.8	53.5	58.4	38.4	59.2	41.6	35.0	55.9	37.1
		10000	55.9	31.2	30.8	28.2	46.1	29.8	26.2	43.4	27.3
	50000	2000	55.7	29.5	27.0	27.5	45.8	28.9	25.7	42.9	26.8
		800	68.4	39.3	39.3	31.6	51.0	33.5	29.1	47.9	31.0
MoCov2 (112)	1.28M	200	81.3	54.1	59.5	38.9	59.3	42.6	35.2	56.4	37.7
		2000	75.1	47.1	50.3	35.2	55.1	38.5	32.3	51.9	34.7
	50000	8000	76.5	48.4	51.9	35.6	55.2	38.7	32.5	52.2	34.7
		4000	78.5	51.0	55.4	36.7	56.6	39.9	33.5	53.8	35.8