arXiv:2103.10957v1 [cs.CV] 19 Mar 2021

Efficient Visual Pretraining with Contrastive Detection

Olivier J. Hénaff Skanda Koppula Jean-Baptiste Alayrac
Aaron van den Oord Oriol Vinyals João Carreira

DeepMind, London, UK

Abstract

Self-supervised pretraining has been shown to yield powerful representations for transfer learning. These performance gains come at a large computational cost however, with state-of-the-art methods requiring an order of magnitude more computation than supervised pretraining. We tackle this computational bottleneck by introducing a new self-supervised objective, contrastive detection, which tasks representations with identifying object-level features across augmentations. This objective extracts a rich learning signal per image, leading to state-of-the-art transfer performance from ImageNet to COCO, while requiring up to 5× less pretraining. In particular, our strongest ImageNet-pretrained model performs on par with SEER, one of the largest self-supervised systems to date, which uses 1000× more pretraining data. Finally, our objective seamlessly handles pretraining on more complex images such as those in COCO, closing the gap with supervised transfer learning from COCO to PASCAL.

1. Introduction

Since the AlexNet breakthrough on ImageNet, transfer learning from large labeled datasets has become the dominant paradigm in computer vision [34, 49]. While recent advances in self-supervised learning have alleviated the dependency on labels for pretraining, they have done so at a tremendous computational cost, with state-of-the-art methods requiring an order of magnitude more computation than supervised pretraining [7, 9, 20]. Yet the promise of self-supervised learning is to harness massive unlabeled datasets, making its computational cost a critical bottleneck.

In this work, we aim to alleviate the computational burden of self-supervised pretraining. To that end we introduce *contrastive detection*, a new objective which maximizes the similarity of object-level features across augmentations. The benefits of this objective are threefold. First, it extracts separate learning signals from all objects in an image, enriching the information provided by each training example for free—object-level features are simply obtained from intermediate feature arrays. Second, it provides a larger and more diverse set of *negative samples* in the contrastive loss, which also accelerates learning. Finally, this objective is well suited to learning from complex scenes with many objects, a pretraining domain that has proven challenging for self-supervised methods.

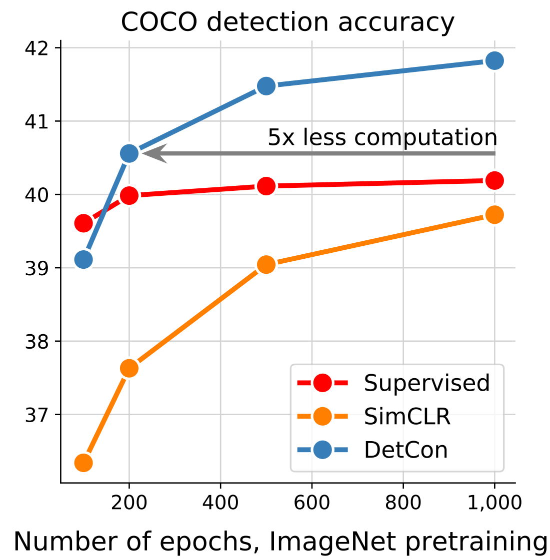

Figure 1. **Efficient self-supervised pretraining with DetCon.** Self-supervised pretraining with SimCLR [8] matches the transfer performance of supervised pretraining only when given 10× more training iterations. Our proposed DetCon objective surpasses both, while requiring 5× less computation than SimCLR. Transfer performance is measured by fine-tuning the representation on the COCO dataset for 12 epochs, using a Mask-RCNN.

We identify approximate object-based regions in the image through the use of unsupervised segmentation algorithms. Perceptual grouping [32, 40]—the idea that low and mid-level regularities in the data such as color, orientation and texture allow for approximately parsing a scene into connected surfaces or object parts—has long been theorized to be a powerful prior for vision [21, 39, 54]. We leverage these priors by grouping local feature vectors accordingly, and applying our contrastive objective to each object-level feature separately. We investigate the use of several unsupervised, image-computable masks [16, 2], and find our objective to work well despite their inaccuracies.

We test the ability of our objective to quickly learn transferable representations by applying it to the ImageNet dataset and measuring its transfer performance on challenging tasks such as COCO detection and instance segmentation. Compared to representations obtained from recent self-supervised objectives such as SimCLR and BYOL [8, 20], our representations are more accurate and can be obtained with much less training time. We also find this learning objective to better handle images of more complex scenes, bridging the gap with supervised transfer from the COCO dataset. In summary, we make the following contributions:

1. We formulate a new contrastive objective which maximizes the similarity across augmentations of all objects in a scene, where object regions are provided by a simple, unsupervised heuristic. We dissect this new objective and assess the improvements afforded by each of its elements.

2. We find this objective to alleviate the computational burden of self-supervised transfer learning, reducing by $5\times$ the computation required to match supervised transfer learning from ImageNet. Longer training schedules lead to state-of-the-art transfer to COCO detection and instance segmentation, and our best model matches the very recent state-of-the-art self-supervised system SEER [19] which is trained on $1000\times$ more—if less curated—images.

3. When transferring from complex scene datasets such as COCO, our method closes the gap with a supervised model which learns from human-annotated segmentations.

4. Finally, we assess to what extent the existing contrastive learning paradigm could be simplified in the presence of high quality image segmentations, raising questions and opening avenues for future work.

2. Related work

Transferring the knowledge contained in one task and dataset to solve other downstream tasks (i.e. *transfer learning*) has proven very successful in a range of computer vision problems [18, 38]. While early work focused on improving the pretraining architecture [26, 50] and data [51], recent work in self-supervised learning has focused on the choice of pretraining objective and task. Early self-supervised pretraining typically involved image restoration, including denoising [58], inpainting [45], colorization [64, 35], egomotion prediction [1], and more [14, 42, 65]. Higher-level pretext tasks have also been studied, such as predicting context [12], orientation [17], spatial layouts [43], temporal ordering [41], and cluster assignments [5].

Contrastive objectives, which maximize the similarity of a representation across views, while minimizing its similarity with distracting negative samples, have recently gained considerable traction [22]. These views have been defined as local and global crops [29, 4, 55, 28] or different input channels [52]. Instance-discrimination approaches generate global, stochastic views of an image through data-augmentation, and maximize their similarity relative to marginally sampled negatives [8, 13, 15, 23, 61], although the need for negative samples has recently been questioned [11, 20]. While the benefits of instance-discrimination approaches have mostly been limited to pretraining from simple datasets such as ImageNet, clustering-based pretraining has proven very successful in leveraging large amounts of uncurated images for transfer learning [3, 6, 7, 19, 31].

While most work has focused on learning whole-image representations, there has been increasing interest in learning local descriptors that are more relevant for downstream tasks such as detection and segmentation. Examples of such work include the addition of auxiliary losses [53], architectural components [47], or both [62]. While perceptual grouping has long been used for representation learning, often relying on coherent motion in videos [36, 44, 59], it has only recently been combined with contrastive learning [30, 57, 66]. Most related to our work are [57, 66] that also leverage image segmentations for self-supervised learning, although both differ from ours in that they learn backbones that are specialized for semantic segmentation and employ different loss functions. Although these works arrive at impressive unsupervised segmentation accuracy, neither report gains in pretraining efficiency for transfer learning tasks such as COCO detection and instance segmentation, which we study next.

3. Method

We introduce a new contrastive objective which maximizes the similarity across views of local features which represent the same object (Figure 2). In order to isolate the benefit of these changes, we make the deliberate choice of re-using elements of existing contrastive learning frameworks where possible. To test the generality of our approach, we derive two variants, DetCon$_S$ and DetCon$_B$, based on two recent self-supervised baselines, SimCLR [8] and BYOL [20] respectively. We adopt the data augmentation procedure and network architecture from these methods while applying our proposed Contrastive Detection loss to each.

3.1. The contrastive detection framework

Data augmentation. Each image is randomly augmented twice, resulting in two images: $\boldsymbol{x}, \boldsymbol{x}'$. DetCon$_S$ and DetCon$_B$ adopt the augmentation pipelines of SimCLR and BYOL respectively, which roughly consist of random cropping, flipping, blurring, and point-wise color transformations. We refer the reader to appendix A.1 for more details. In all cases, images are resized to 224×224 pixel resolution.

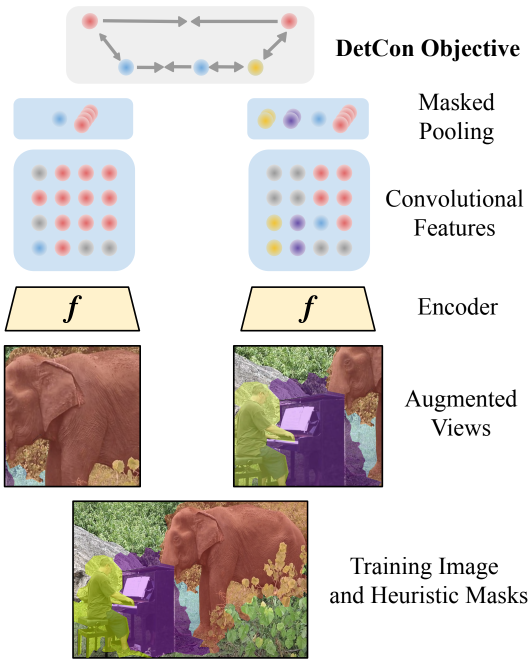

Figure 2. **The contrastive detection method.** We identify object-based regions with approximate, image-computable segmentation algorithms (bottom). These masks are carried through two stochastic data augmentations and a convolutional feature extractor, creating groups of feature vectors in each view (middle). The contrastive detection objective then pulls together pooled feature vectors from the same mask (across views) and pushes apart features from different masks and different images (top).

In addition, we compute for each image a set of masks which segment the image into different components. As described in Section 3.2, these masks can be computed using efficient, off-the-shelf, unsupervised segmentation algorithms. If available, human-annotated segmentations can also be used. In any case, we transform each mask (represented as a binary image) using the same cropping and resizing as used for the underlying RGB image, resulting in two sets of masks $\{\boldsymbol{m}\}, \{\boldsymbol{m}'\}$ which are aligned with the augmented images $\boldsymbol{x}, \boldsymbol{x}'$ (see Figure 2, *augmented views*).

Architecture. We use a convolutional feature extractor f to encode each image with a spatial map of hidden vectors: $\boldsymbol{h} = f(\boldsymbol{x})$ where $\boldsymbol{h} \in \mathbb{R}^{H \times W \times D}$. We use the output of a standard ResNet-50 encoder [26], before the final mean-pooling layer, such that hiddens form a 7×7 grid of 2048-dimensional vectors $\boldsymbol{h}[i,j]$. For every mask $\boldsymbol{m}$ associated with the image, we compute a mask-pooled hidden vector

$$\boldsymbol{h}_{\boldsymbol{m}} = \frac{1}{\sum_{i,j} m_{i,j}} \sum_{i,j} m_{i,j}\, \boldsymbol{h}[i,j],$$

having spatially downsampled the binary mask to a 7×7 grid with average pooling. We then transform each of these vectors with a two-layer MLP, yielding non-linear projections $\boldsymbol{z}_{\boldsymbol{m}} = g(\boldsymbol{h}_{\boldsymbol{m}}) \in \mathbb{R}^d$. Note that we can recover the architecture of SimCLR and BYOL by using a single global mask in this step.

For DetCon$_S$ we process both views with the same encoder f_θ and projection network g_θ where θ are the learned parameters. For DetCon$_B$ one view is processed with f_θ and g_θ and the other with f_ξ and g_ξ, where ξ is an exponential moving average of θ. The first view is further transformed with a prediction network q_θ. Here again we reuse the details of SimCLR and BYOL for DetCon$_S$ and DetCon$_B$ respectively in the definition of the projection and prediction networks (see appendix A.2). In summary, we represent each view and mask as latents $\boldsymbol{v}_{\boldsymbol{m}}$ and $\boldsymbol{v}'_{\boldsymbol{m}'}$ where

$$\boldsymbol{v}_{\boldsymbol{m}} = g_\theta(\boldsymbol{h}_{\boldsymbol{m}}), \quad \boldsymbol{v}'_{\boldsymbol{m}'} = g_\theta(\boldsymbol{h}'_{\boldsymbol{m}'})$$

for **DetCon**$_S$ and

$$\boldsymbol{v}_{\boldsymbol{m}} = q_\theta \circ g_\theta(\boldsymbol{h}_{\boldsymbol{m}}), \quad \boldsymbol{v}'_{\boldsymbol{m}'} = g_\xi(\boldsymbol{h}'_{\boldsymbol{m}'})$$

for **DetCon**$_B$. We rescale all latents with a temperature hyperparameter τ, such that their norm is equal to $1/\sqrt{\tau}$, with $\tau = 0.1$. Note that for downstream tasks, we retain only the feature extractor f_θ and discard all other parts of the network (the prediction and projection heads, as well as any exponential moving averages).

Objective: contrastive detection. Let $\boldsymbol{v}_{\boldsymbol{m}}, \boldsymbol{v}'_{\boldsymbol{m}'}$ be the latents representing masks $\boldsymbol{m}, \boldsymbol{m}'$ in the views $\boldsymbol{x}, \boldsymbol{x}'$. The contrastive loss function

$$\ell_{\boldsymbol{m},\boldsymbol{m}'} = -\log \frac{\exp(\boldsymbol{v}_{\boldsymbol{m}} \cdot \boldsymbol{v}'_{\boldsymbol{m}'})}{\exp(\boldsymbol{v}_{\boldsymbol{m}} \cdot \boldsymbol{v}'_{\boldsymbol{m}'}) + \sum_{\boldsymbol{n}} \exp(\boldsymbol{v}_{\boldsymbol{m}} \cdot \boldsymbol{v}_{\boldsymbol{n}})} \tag{1}$$

defines a prediction task: having observed the projection $\boldsymbol{v}_{\boldsymbol{m}}$, learn to recognize the latent $\boldsymbol{v}'_{\boldsymbol{m}'}$ in the presence of negative samples $\{\boldsymbol{v}_{\boldsymbol{n}}\}$. We include negative samples from different masks in the image and different images in the batch. Note that we make no assumptions regarding these masks, allowing negative masks to overlap with the positive one.

A natural extension of this loss would be to jointly sample paired masks $\boldsymbol{m}, \boldsymbol{m}'$ which correspond to the same region in the original image, and maximize the similarity of features representing them

$$\mathcal{L} = \mathbb{E}_{(\boldsymbol{m},\boldsymbol{m}') \sim \mathcal{M}}\, \ell_{\boldsymbol{m},\boldsymbol{m}'}. \tag{2}$$

We make a few practical changes to this objective. First, in order to facilitate batched computation we randomly sample at each iteration a set of 16 (possibly redundant) masks from the variable-sized sets of masks $\{\boldsymbol{m}\}$ and $\{\boldsymbol{m}'\}$. Second, we densely evaluate the similarity between all pairs of masks and all images, such that each image contributes 16 negative samples to the set $\{\boldsymbol{v}_{\boldsymbol{n}}\}$ in equation (1), rather than a single one. We aim to makes these negatives as diverse as possible by choosing masks that roughly match different objects in the scene (Section 3.2). Finally, we mask out the

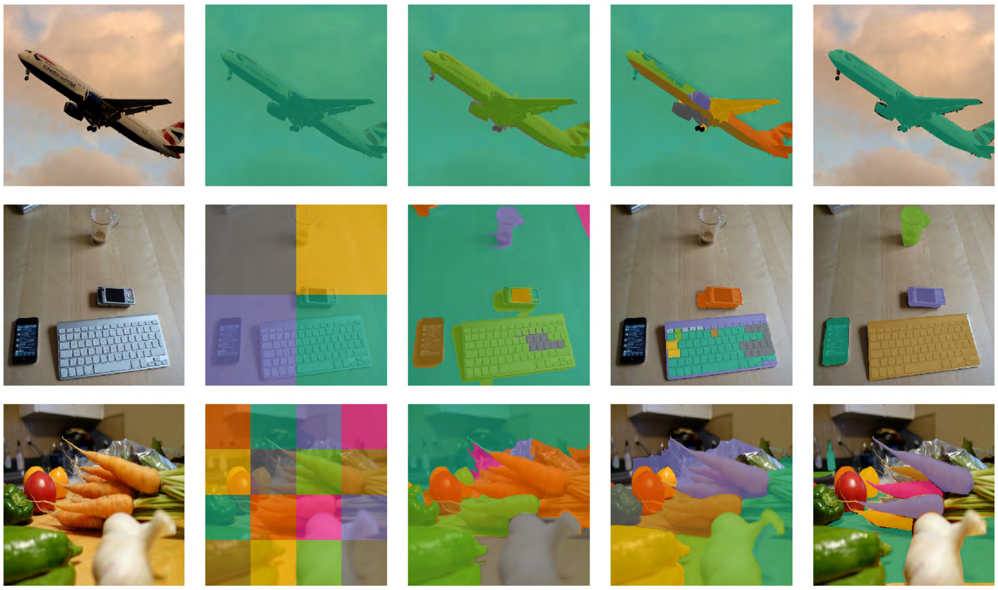

Figure 3. **Example masks used by the DetCon model.** 1st column: random images from the COCO training set. 2nd column: masks based on spatial proximity only. Global masks (top) are implicitly used by methods such as SimCLR, MoCo, and BYOL. 3rd column: image-computable masks obtained from the Felzenszwalb-Huttenlocher (FH, [16]) algorithm, with $s = 500$. 4th column: image-computable masks inferred using Multiscale Combinatorial Grouping (MCG) [2]. 5th column: "oracle" masks used to assess potential improvements from higher-quality segmentations.

loss to only maximize the similarity of paired locations, allowing us to handle cases where a mask is present in one view but not another (see Figure 2). Together, these simple modifications bring us to the DetCon objective:

$$\mathcal{L} = \sum_{\boldsymbol{m}} \sum_{\boldsymbol{m}'} \mathbb{1}_{\boldsymbol{m},\boldsymbol{m}'} \ell_{\boldsymbol{m},\boldsymbol{m}'} \tag{3}$$

where the binary variable $\mathbb{1}_{\boldsymbol{m},\boldsymbol{m}'}$ indicates whether the masks $\boldsymbol{m}, \boldsymbol{m}'$ correspond to the same underlying region.

Optimization. When pretraining on ImageNet we adopt the optimization details of SimCLR and BYOL for training DetCon_S and DetCon_B respectively. When pretraining on COCO we make minor changes to the learning schedule to alleviate overfitting (see appendix A.3).

Computational cost. The computational requirements of self-supervised learning are largely due to forward and backward passes through the convolutional backbone. For the typical ResNet-50 architecture applied to 224×224-resolution images, a single forward pass requires approximately 4B FLOPS. The additional projection head in SimCLR and DetCon_S requires an additional 4M FLOPS. Since we forward 16 hidden vectors through the projection head instead of 1, we increase the computational cost of the forward pass by 67M FLOPS, less than 2% of total. Together with the added complexity of the contrastive loss, this increase is 5.3% for DetCon_S and 11.6% for DetCon_B (see appendix A.2). Finally, the cost of computing image segmentations is negligible because they can be computed once and reused throughout training. Therefore the increase in complexity of our method relative to the baseline is sufficiently small for us to interchangeably refer to "training iterations" and "computational cost".

3.2. Unsupervised mask generation

To produce masks required by the DetCon objective, we investigate several segmentation procedures, from simple spatial heuristics to graph-based algorithms from the literature.

Spatial heuristic. The simplest segmentation we consider groups locations based on their spatial proximity only. Specifically, we divide the image into an $n \times n$ grid of non-overlapping, square sub-regions (Figure 3, 2nd column). As noted in Section 3.1, when using a single, global mask ($n = 1$), the DetCon_S objective reverts to SimCLR.

Image-computable masks: FH. We also consider the Felzenszwalb-Huttenlocher algorithm [16], a classic segmentation procedure which iteratively merges regions using pixel-based affinity (Figure 3, 3rd column). We generate a diverse set of masks by varying two hyperparameters, the scale s and minimum cluster size c, using $s \in \{500, 1000, 1500\}$ and $c = s$ when training on COCO and $s = 1000$ when training on ImageNet.

Image-computable masks: MCG. Multiscale Combinatorial Grouping [2] is a more sophisticated algorithm which groups superpixels into many overlapping object proposal regions, guided by mid-level classifiers (Figure 3, 4th column). For each image we use 16 MCG masks with the highest scores. Note that the fact that the masks can overlap is supported by our formulation.

Human annotated masks. Throughout this work we consider the benefits afforded by the use of the *unsupervised* masks detailed above. In the final section, we ask whether higher quality masks (provided by human annotators; Figure 3, 5th column) can improve our results.

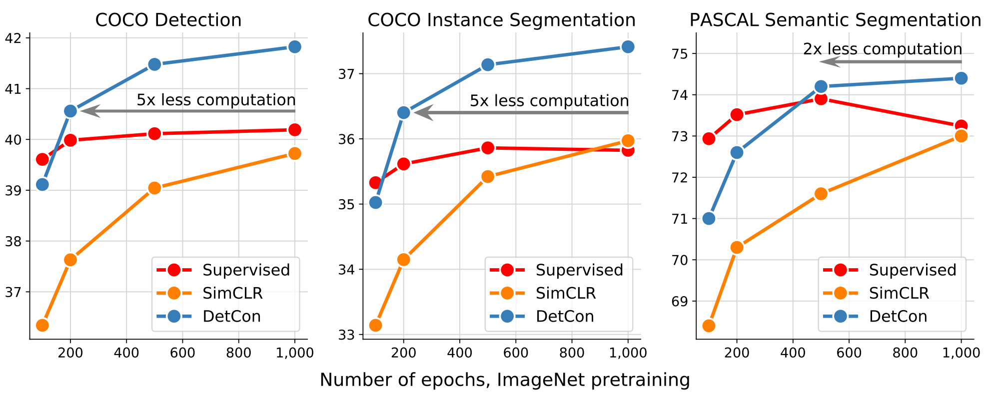

Figure 4. **Efficient ImageNet pretraining.** We pretrain models with SimCLR, DetCon$_S$, or supervised learning on ImageNet for different numbers of epochs, and evaluate their accuracy on COCO detection (left), COCO instance segmentation (middle), or PASCAL semantic segmentation (right) by fine-tuning them for 12 epochs on COCO or 45 epochs on PASCAL. DetCon outperforms SimCLR, with up to 5× less computation.

	Detection (AP^{bb})		Segmentation (AP^{mk})	
Pretrain epochs	200	1000	200	1000
SimCLR	37.6	39.7	34.1	36.0
DetCon$_S$	40.6	41.8	36.4	37.4
Efficiency Gain		$> 5\times$		$> 5\times$
Pretrain epochs	300	1000	300	1000
BYOL	41.2	41.6	37.1	37.2
DetCon$_B$	42.0	42.7	37.8	38.2
Efficiency Gain		$> 3\times$		$> 3\times$

Table 1. **Efficient ImageNet pretraining.** We pretrain models on ImageNet with SimCLR, BYOL, DetCon$_S$ or DetCon$_B$ and fine-tune them on COCO for 12 epochs, reporting bounding-box AP (AP^{bb}) and mask AP (AP^{mk}). Colors highlight comparisons between models which illustrate gains in pretraining efficiency.

3.3. Evaluation protocol

Having trained a feature extractor in an unsupervised manner, we evaluate the quality of the representation by transferring it to COCO detection, COCO instance segmentation and PASCAL semantic segmentation tasks.

Transfer to COCO. We use the representation to initialize the feature extractor of a Mask-RCNN [25] equipped with feature pyramid networks [37] and cross-replica batch-norm [46]. We adopt the publicly available Cloud TPU implementation[1] and use it without modification. We fine-tune the whole model end-to-end, and report the bounding-box AP (AP^{bb}) and mask AP (AP^{mk}) evaluated on the `val2017` set. We use two standard training schedules: 12 epochs (a "1×schedule") and 24 epochs (a "2×schedule") [23].

Transfer to PASCAL. Following [23] we stack a fully-convolutional network [38] on top of the representation and train the network end-to-end for semantic segmentation. We train on the `train_aug2012` set for 45 epochs and we report the mean intersection over union (mIoU) on the `val2012` set.

[1] https://github.com/tensorflow/tpu/tree/master/models/official/detection

4. Experiments

Our main self-supervised learning experiments employ FH masks, because as we will show, with DetCon they outperform simple spatial heuristics and approach the performance of MCG masks while being fast and easy to apply to large datasets such as ImageNet, given their availability in `scikit-image` [56].

4.1. Transfer learning from ImageNet

We first study whether the DetCon objective improves the pretraining efficiency of transfer learning from ImageNet.

Training efficiency. We train SimCLR and DetCon$_S$ models on ImageNet for 100, 200, 500 and 1000 epochs, and transfer them to COCO and PASCAL. When fine-tuning on COCO, DetCon$_S$ substantially outperforms SimCLR across all training regimes (Figure 4, left, middle). Equivalently, the maximum performance attained by SimCLR is reached by DetCon$_S$ with 5× less training iterations (Table 1, top). We also transferred these same models to semantic segmentation on PASCAL, and found similar results (DetCon$_S$ yields a 2× gain in pretraining efficiency; Figure 4, right).

We also evaluated the transfer performance of a supervised ResNet-50 trained on ImageNet (Figure 4, red). Note that while SimCLR reaches the accuracy of supervised transfer, consistently with prior work [8, 23], it does so at a greatly increased computational cost, requiring 10× more pretraining iterations to arrive at this representation quality. In contrast, DetCon$_S$ converges much faster to useful representations: 200 epochs are sufficient to surpass supervised transfer to COCO, and 500 to PASCAL.

From BYOL to DetCon$_B$. How general is DetCon? We tested this by comparing DetCon$_B$ to the BYOL framework, upon which it is based. Here too, we adopt the underlying framework details (regarding architecture, optimization, etc) *without modification*, possibly putting the DetCon objective at a disadvantage. Despite this, DetCon$_B$ outperforms BYOL across pretaining budgets, yielding a 3× gain in pretraining efficiency (Table 1, bottom).

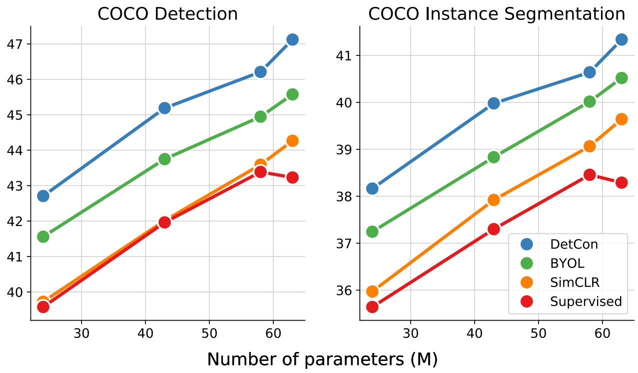

Figure 5. **Scaling DetCon to larger models.** We pretrain ResNet-50, ResNet-101, ResNet-152, and ResNet-200 feature extractors on ImageNet using supervised learning, SimCLR, BYOL, or DetCon$_B$ and fine-tune them on COCO for 12 epochs.

Comparison with prior art. We now compare to prior work on self-supervised transfer learning, and use fully-trained DetCon$_S$ and DetCon$_B$ models for the comparison. Note that other methods use a slightly different implementation of the Mask-RCNN [60] however their results for supervised ImageNet pretraining and SimCLR match our own [53, 62], enabling a fair comparison. Table 2a shows that DetCon outperforms all other methods for supervised and self-supervised transfer learning.

Scaling model capacity. Prior works in self-supervised learning have been shown to scale very well with model capacity [13, 33, 8]. Could the gains afforded by DetCon disappear with larger models? We trained SimCLR, BYOL, and DetCon$_B$ models on ImageNet, using ResNet-101, ResNet-152, and ResNet-200 feature extractors instead of ResNet-50. Tables 2b and 2c, and Figure 5 show that DetCon continues to outperform other methods in this higher-capacity regime.

We went a step further and trained a ResNet-200 with a 2× width multiplier [33], containing 250M parameters. Surprisingly, despite only being trained on ImageNet, this model's transfer performance matches that of a very recently proposed large-scale self-supervised model, SEER [19], having 693M parameters and trained on 1000× more data (Table 3). While the comparison is imperfect (large-scale data is necessarily more noisy), it highlights the potential of improvements from the self-supervised learning objective alone.

4.2. Transfer learning from COCO

We next investigate the ability of the DetCon objective to handle complex scenes with multiple objects. For this we pretrain on the COCO dataset and compare to SimCLR.

Training efficiency. We train SimCLR and DetCon$_S$ for a range of schedules (324-5184 epochs), and transfer all models to semantic segmentation on PASCAL. We find DetCon$_S$ to outperform SimCLR across training budgets (Figure 6). Equivalently, the maximum accuracy attained by SimCLR is reached with 4× less pretraining time.

method	Fine-tune 1×		Fine-tune 2×	
	AP^{bb}	AP^{mk}	AP^{bb}	AP^{mk}
Supervised	39.6	35.6	41.6	37.6
VADeR [47]	39.2	35.6	-	-
MoCo [23]	39.4	35.6	41.7	37.5
SimCLR [8]	39.7	35.8	41.6	37.4
MoCo v2 [10]	40.1	36.3	41.7	37.6
InfoMin [53]	40.6	36.7	42.5	38.4
PixPro [62]	41.4	-	-	-
BYOL [20]	41.6	37.2	42.4	38.0
SwAV [7]	41.6	37.8	-	-
DetCon$_S$	41.8	37.4	42.9	38.1
DetCon$_B$	**42.7**	**38.2**	**43.4**	**38.7**

(a) **ResNet-50** feature extractor

method	Fine-tune 1×		Fine-tune 2×	
	AP^{bb}	AP^{mk}	AP^{bb}	AP^{mk}
Supervised	42.0	37.3	43.4	38.4
SimCLR [8]	42.0	37.9	43.8	39.3
InfoMin [53]	42.9	38.6	44.5	39.9
BYOL [20]	43.7	38.8	44.3	39.4
DetCon$_B$	**45.2**	**40.0**	**45.7**	**40.4**

(b) **ResNet-101** feature extractor

method	Fine-tune 1×		Fine-tune 2×	
	AP^{bb}	AP^{mk}	AP^{bb}	AP^{mk}
Supervised	43.2	38.3	43.5	38.5
SimCLR [8]	44.3	39.6	45.3	40.3
BYOL [20]	45.6	40.5	45.9	40.5
DetCon$_B$	**47.1**	**41.3**	**47.2**	**41.5**

(c) **ResNet-200** feature extractor

Table 2. **Comparison to prior art:** all methods are pretrained on ImageNet then fined-tuned on COCO for 12 epochs (1× schedule) or 24 epochs (2× schedule). Bounding-box AP (AP^{bb}) and mask AP (AP^{mk}) are evaluated on the COCO `val2017` set.

pretrain	Data	Params (M)	AP^{bb}	AP^{mk}
Supervised [19]	IN-1M	250	45.9	41.0
SEER [19]	**IG-1B**	**693**	48.5	**43.2**
DetCon$_B$	IN-1M	250	**48.9**	43.0

Table 3. **Comparison to large-scale transfer learning:** all methods pretrain a backbone and transfer to COCO detection and instance segmentation using a Mask-RCNN. SEER trains on a billion Instagram images whereas DetCon$_S$ trains on ImageNet (1.3 million images). SEER and the supervised baseline use the recent RegNet architecture [48], whereas DetCon$_S$ uses a ResNet-200 with 2× width. Despite this, DetCon pretraining matches the performance of large-scale SEER pretraining.

Surpassing supervised transfer from COCO. We also evaluated the transfer performance of representations trained on COCO in a supervised manner. Specifically, we trained a Mask-RCNN with a long schedule (108 epochs, a "9×" schedule), and use the learned feature extractor (a ResNet-50, as for SimCLR and DetCon pretraining) as a representation for PASCAL segmentation. While DetCon pretraining surpasses the performance of the supervised baseline, SimCLR continues to lag behind (Figure 6).

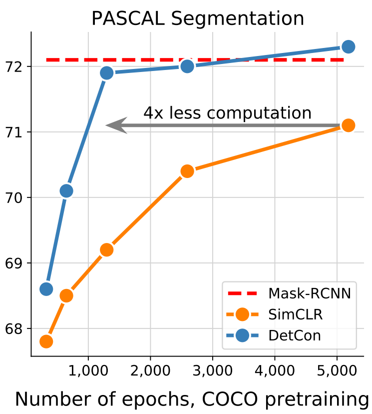

Figure 6. **Efficient transfer from COCO.** We pretrain representations using SimCLR or DetCon$_S$ on COCO for different numbers of epochs, and transfer to PASCAL semantic segmentation by finetuning them for 45 epochs.

4.3. Ablations and analysis

We now dissect the components of the DetCon objective and assess the benefits of each. For this we pretrain on COCO as it contains complex scenes with many objects and associated ground-truth masks, allowing us to measure the impact of segmenting them accurately. We evaluate learned representations with a *frozen-feature analysis*, in which the feature extractor is kept fixed while we train the other layers of a Mask-RCNN also on COCO. This controlled setting is analogous[2] to the linear classification protocol used to evaluate the quality of self-supervised representations for image recognition [12, 14, 44, 64]

What makes good masks? The DetCon objective can be used with a variety of different image segmentations, which ones lead to the best representation? We first consider spatial heuristics which partition the image into 2×2, 5×5, or 10×10 grids, a 1×1 grid being equivalent to using the SimCLR objective. We find downstream performance increases with finer grids, a 5×5 grid being optimal (Figure 7).

Next we consider image-computable FH and MCG masks, both of which outperform the spatial heuristic masks, MCG masks leading to slightly better representations. Interestingly, the quality of the representation *correlates very well with the overlap between pretraining masks and ground-truth*—the better each ground truth object is covered by some mask, the better DetCon performs.

[2] But note that the Mask-RCNN contains several non-linear layers due to the additional complexity of the output space relative to classification.

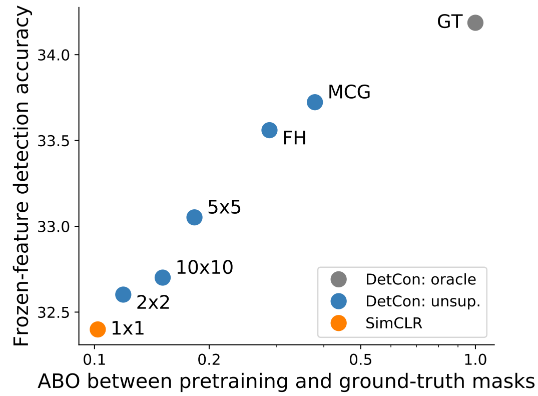

Figure 7. **Effect of type of masks used in DetCon objective.** We train DetCon models on COCO using unsupervised masks (blue), or the ground-truth COCO masks (grey). Using a single, global mask (i.e. a "1×1" grid) is equivalent to SimCLR (orange). We compute the Average Best Overlap (ABO) by measuring the IoU between each ground-truth mask and the closest pretraining mask, and averaging over all ground truth instances and images (x-axis). We evaluate the accuracy of each model on COCO detection using the *frozen-feature* paradigm (y-axis).

model	masks	#latents	AP^{bb}	AP^{mk}
SimCLR	global	1	31.6	29.2
(a)	global	16	31.5 (−0.1)	29.3 (+0.1)
(b)	FH	1	31.2 (−0.4)	28.8 (−0.4)
DetCon$_S$	FH	16	33.4 (+1.8)	30.6 (+1.4)

Table 4. **Ablation: from SimCLR to DetCon$_S$.** We pretrain on COCO and evaluate *frozen feature* accuracy also on COCO. **masks:** specifies whether hidden vectors are pooled globally, or within individual FH masks. **#latents:** number of masks.

Contrastive detection vs contrastive recognition. How does the DetCon objective benefit from these image segmentations? We assess the impact of each of its components by incrementally adding them to the SimCLR framework. As mentioned in Section 3.1, we recover SimCLR when using a single, global mask in the DetCon objective. As a sanity check, we verify that duplicating this mask several times and including the resulting (identical) features in the DetCon objective makes no difference in the quality of the representation (Table 4, row **a**). Interestingly, using FH masks but only sampling a single mask per image slightly deteriorates performance, presumably because the model only learns from part of the image at every iteration (Table 4, row **b**). By densely sampling object regions DetCon$_S$ learns from the entire image, while also benefiting from a diverse set of positive and negative samples, resulting in increased detection and segmentation accuracy (Table 4, final row).

4.4. What if segmentation were solved?

The DetCon objective function leads to fast transfer learning and strong performance despite using fairly approximate segmentation masks. In Section 4.3 we found higher

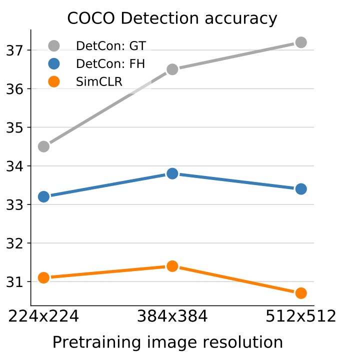

Figure 8. **Better segmentations benefit from higher resolutions.** We pretrain backbone networks on COCO using SimCLR and DetCon$_S$ (with FH or GT masks) at various resolutions. We report *frozen-feature* performance with a fixed resolution of 1024×1024.

quality segmentations (such as those computed using MCG, or obtained from human annotators) to improve representational quality. How might we improve the learning objective given more accurate segmentations? We assessed this question by revisiting our design choices for the contrastive objective, when given ground-truth masks from the COCO dataset as opposed to the approximate FH masks.

Scaling image resolution. We hypothesized that higher image resolutions might enable the network to benefit more from these more informative segmentations. To preserve fine-grained information we sample local features from within each mask and optimize them using the DetCon objective. We pretrain SimCLR and DetCon$_S$ models equipped with FH or ground-truth (GT) masks, given 384×384 or 512×512 resolution images. While DetCon with FH masks only modestly benefited and SimCLR's performance deteriorated with high-resolution images, DetCon with GT masks improves substantially (Figure 8). Note that this is solely due to an improved representation quality; the image resolution used for downstream evaluation is maintained at 1024×1024 for all models.

Revisiting the contrastive framework. Finally, we asked whether the current contrastive learning paradigm—which utilizes large numbers of negatives and predictions across stochastic augmentations—remains optimal in the context of the DetCon objective with high-quality segmentations.

Are large numbers of negative samples necessary? Not with high-quality masks. When dividing the total number of negative samples by 128 (by only gathering negatives from within a worker) the performance of DetCon with FH masks drops (Table 5, row **a**), consistently with current contrastive learning frameworks [8, 23]. In contrast, DetCon$_S$ using GT masks improves despite this limitation.

Are positive pairs sampled across augmented views necessary? Not with high-quality masks. We run DetCon models while sampling a single augmentation for each image and maximizing the similarity of mask-based features *within* this view. Here again, the DetCon objective suffers from this handicap when using approximate FH masks, but not with high-quality segmentations (Table 5, row **b**).

model	global neg	two views	Masks FH		Masks GT	
DetCon	✓	✓	33.6		37.0	
(a)		✓	32.2	(−1.4)	38.5	(+1.5)
(b)			27.7	(−5.9)	38.8	(+1.8)

Table 5. **Simplifying the contrastive framework.** We train DetCon$_S$ models on COCO using approximate FH masks or higher-quality ground-truth (GT) masks, and evaluate them in the *frozen-feature* setting. **"global neg":** Negative samples are collected from the entire batch as opposed to only within a worker (out of 128 workers). **"two views":** Contrastive predictions are made across augmentations, as opposed to within a view.

How can this be? One interpretation is that other images give us clean negative examples because all images in COCO depict different scenes, but this does not mean these are higher quality than negative examples from the same image. In fact it looks like negatives from the same image provide even better signal as long as we can make sure they are clean—i.e., we are not pushing features from the same object apart. Positives from the same image are also at least as good as those across augmentation if again they are clean—i.e., we are not pulling together features from different objects.

5. Discussion

We have proposed DetCon, a simple but powerful variation on existing self-supervised learning algorithms such as SimCLR and BYOL. By exploiting low-level cues for organizing images into entities such as objects and background regions, DetCon boosts the efficiency of pretraining on large datasets by up to 5×, while also improving the accuracy of the learned representations on downstream tasks. Our best models achieve state-of-the-art performance among self-supervised methods pretrained on ImageNet and similar performance to a recent state-of-the-art method training a larger model on a much larger dataset [19].

We showed that the power of DetCon strongly correlates with how well the masks used align with object boundaries. This seems intuitive—the DetCon objective can only leverage independent learning signals from each image region if they contain distinct content. Similarly, the resulting negative samples are genuinely diverse only if they represent different objects. This however creates exciting prospects of research in jointly learning representations and discovering objects. Given the improved performance of DetCon representations for instance segmentation, a natural question is whether they could be used to perform better unsupervised segmentations than the ones used during pretraining. If so, these might be used to learn better representations still, leading to a virtuous crescendo of unsupervised scene understanding.

Acknowledgements

The authors are grateful to Carl Doersch, Raia Hadsell, and Evan Shelhamer for insightful discussions and feedback on the manuscript.

A. Appendix

A.1. Implementation: data augmentation

Self-supervised pretraining. Each image is randomly augmented twice, resulting in two images: $\boldsymbol{x}, \boldsymbol{x}'$. The augmentations are constructed as compositions of the following operations, each applied with a given probability:

1. random cropping: a random patch of the image is selected, whose area is uniformly sampled in $[0.08 \cdot \mathcal{A}, \mathcal{A}]$, where $\mathcal{A}$ is the area of the original image, and whose aspect ratio is logarithmically sampled in $[3/4, 4/3]$. The patch is then resized to 224 ×224 pixels using bicubic interpolation;
2. horizontal flipping;
3. color jittering: the brightness, contrast, saturation and hue are shifted by a uniformly distributed offset;
4. color dropping: the RGB image is replaced by its grey-scale values;
5. gaussian blurring with a 23×23 square kernel and a standard deviation uniformly sampled from $[0.1, 2.0]$;
6. solarization: a point-wise color transformation $x \mapsto x \cdot \mathbb{1}_{x<0.5} + (1-x) \cdot \mathbb{1}_{x\geq 0.5}$ with pixels x in $[0, 1]$.

The augmented images $\boldsymbol{x}, \boldsymbol{x}'$ result from augmentations sampled from distributions $\mathcal{T}$ and $\mathcal{T}'$ respectively. These distributions apply the primitives described above with different probabilities, and different magnitudes. The following table specifies these parameters for the SimCLR [8] and BYOL frameworks [20], which we adopt for DetCon$_S$ and DetCon$_B$ without modification.

	DetCon$_S$		DetCon$_B$	
Parameter	$\mathcal{T}$	$\mathcal{T}'$	$\mathcal{T}$	$\mathcal{T}'$
Random crop probability	1.0			
Flip probability	0.5			
Color jittering probability	0.8			
Color dropping probability	0.2			
Brightness adjustment max	0.8		0.4	
Contrast adjustment max	0.8		0.4	
Saturation adjustment max	0.8		0.2	
Hue adjustment max	0.2		0.1	
Gaussian blurring probability	1.0	0.0	1.0	0.1
Solarization probability	0.0	0.0	0.0	0.2

Transfer to COCO. When fine-tuning, image are randomly flipped and resized to a resolution of $u \cdot 1024$ pixels on the longest side, where u is uniformly sampled in $[0.8, 1.25]$, then cropped or padded to a 1024×1024 image. The aspect ratio is kept the same as the original image. During testing, images are resized to 1024 pixels on the longest side then padded to 1024×1024 pixels.

Transfer to PASCAL. During training, images are randomly flipped and scaled by a factor in $[0.5, 2.0]$. Training and testing are performed with 513×513-resolution images.

A.2. Implementation: architecture

Our default feature extractor is a ResNet-50 [27]. In Section 4.1 we also investigate deeper architectures (ResNet-101, -152, and -200), and a wider model (ResNet-200 ×2) obtained by scaling all channel dimensions by a factor of 2.

As detailed in Section 3.1, this encoder yields a grid of hidden vectors which we pool within masks to obtain a set of vectors $\boldsymbol{h_m}$ representing each mask. These are then transformed by a projection head g (and optionally a prediction head q) before entering the contrastive loss.

DetCon$_S$. Following SimCLR, the projection head is a two-layer MLP whose hidden and output dimensions are 2048 and 128. The network uses the learned parameters θ for both views.

DetCon$_B$. Following BYOL, the projection head is a two-layer MLP whose hidden and output dimensions are 4096 and 256. The network uses the learned parameters θ for processing one view, and an exponential moving average of these parameters ξ for processing the second. Specifically, ξ is updated using $\xi \leftarrow \lambda \cdot \xi + (1-\lambda) \cdot \theta$, where the decay rate λ is annealed over the course of training from λ_0 to 1 using a cosine schedule [20]. λ_0 is set to 0.996 when training for 1000 epochs and 0.99 when training for 300 epochs. The projection of the first view is further transformed with a prediction head, whose architecture is identical to that of the projection head.

Computational cost. The forward pass through a ResNet-50 encoder requires roughly 4B FLOPS. Ignoring the cost of bias terms and point-wise nonlinearities, the projection head in DetCon$_S$ requires roughly 4.4M FLOPS (i.e. $2048 \times 2048 + 2048 \times 128$). Since this is calculated 16 times rather than once, it results in an overhead of 67M FLOPS compared to SimCLR. For DetCon$_B$ the combined cost of evaluating the projection and prediction heads results in an additional 173M FLOPS compared to BYOL. Finally, the cost of evaluating the contrastive loss is 134M FLOPS for DetCon$_S$ and 268M FLOPS for DetCon$_B$. In total DetCon$_S$ requires 201M additional FLOPS and DetCon$_B$ 441M which represent 5.3% and 11.6% of the cost of evaluating the backbone. This overhead is sufficiently small compared to the gain in training iterations required to reach a given transfer performance (*e.g.* a 500% gain for DetCon$_S$

over SimCLR, and a 333% for DetCon$_B$ over DetCon) for us not further distinguish between gains in computation and training time.

A.3. Implementation: optimization

Self-supervised pretraining. We train using the LARS optimizer [63] with a batch size of 4096 split across 128 Cloud TPU v3 workers. When training on ImageNet we again adopt the optimization details of SimCLR and BYOL for DetCon$_S$ and DetCon$_B$, scaling the learning rate linearly with the batch size and decaying it according to a cosine schedule. For DetCon$_S$ the base learning rate is 0.3 and the weight decay is 10^{-6}. DetCon$_B$ also uses these values when training for 300 epochs; when training for 1000 epochs they are 0.2 and $1.5 \cdot 10^{-6}$.

When pretraining on COCO, we replace the cosine learning rate schedule with a piecewise constant, which has been found to alleviate overfitting [24], dropping the learning rate by a factor of 10 at the 96$^{\text{th}}$ and 98$^{\text{th}}$ percentiles. For fair comparison we use the same schedules when applying SimCLR to the COCO dataset, which we also find to perform better than the more aggressive cosine schedule.

Transfer to COCO. We fine-tune with stochastic gradient descent, increasing the learning rate linearly for the first 500 iterations and dropping twice by a factor of 10, after $\frac{2}{3}$ and $\frac{8}{9}$ of the total training time, following [60]. We use a base learning rate of 0.3 for ResNet-50 models and 0.2 for larger ones, momentum of 0.9, weight decay of $4 \cdot 10^{-5}$ and a batch size of 64 images split across 16 workers.

Transfer to PASCAL. We fine-tune for 45 epochs with stochastic gradient descent, with a batch size of 16 and weight decay of 10^{-4}. The learning rate is 0.02 and dropped by a factor of 10 at the 70$^{\text{th}}$ and 90$^{\text{th}}$ percentiles.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 2

[2] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan T Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 328–335, 2014. 2, 4

[3] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. *arXiv preprint arXiv:1911.05371*, 2019. 2

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019. 2

[5] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2

[6] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Leveraging large-scale uncurated data for unsupervised pre-training of visual features. In *Leveraging Large-Scale Uncurated Data for Unsupervised Pre-training of Visual Features*, 2019. 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020. 1, 2, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 1, 2, 5, 6, 8, 9

[9] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020. 1

[10] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 6

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 2

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015. 2, 7

[13] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2051–2060, 2017. 2, 6

[14] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2, 7

[15] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014. 2

[16] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004. 2, 4

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2

[18] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 2

[19] Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, et al. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 2, 6, 8

[20] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach

to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 6, 9

[21] Jose A Guerrero-Colón, Eero P Simoncelli, and Javier Portilla. Image denoising using mixtures of gaussian scale mixtures. In *2008 15th IEEE International Conference on Image Processing*, pages 565–568. IEEE, 2008. 1

[22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 5, 6, 8

[24] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pages 4918–4927, 2019. 10

[25] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5

[26] 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. 2, 3

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016. 9

[28] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 2

[29] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018. 2

[30] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *arXiv preprint arXiv:2006.14613*, 2020. 2

[31] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019. 2

[32] Kurt Koffka. *Principles of Gestalt psychology*, volume 44. Routledge, 2013. 1

[33] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *CoRR*, abs/1901.09005, 2019. 6

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[35] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, pages 6874–6883, 2017. 2

[36] Yin Li, Manohar Paluri, James M Rehg, and Piotr Dollár. Unsupervised learning of edges. In *CVPR*, 2016. 2

[37] 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. 5

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2, 5

[39] David Lowe. *Perceptual organization and visual recognition*, volume 5. Springer Science & Business Media, 2012. 1

[40] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001. 1

[41] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 2

[42] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018. 2

[43] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 2

[44] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. *arXiv preprint arXiv:1612.06370*, 2016. 2, 7

[45] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016. 2

[46] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6181–6189, 2018. 5

[47] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. *arXiv preprint arXiv:2011.05499*, 2020. 2, 6

[48] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 6

[49] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy,

Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[51] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 2

[52] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

[53] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020. 2, 6

[54] Shimon Ullman et al. *High-level vision: Object recognition and visual cognition*, volume 2. MIT press Cambridge, MA, 1996. 1

[55] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[56] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2014. 5

[57] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. *arXiv preprint arXiv:2102.06191*, 2021. 2

[58] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008. 2

[59] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 2

[60] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 6, 10

[61] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[62] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. *arXiv preprint arXiv:2011.10043*, 2020. 2, 6

[63] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*, 2017. 10

[64] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 2, 7

[65] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017. 2

[66] Xiao Zhang and Michael Maire. Self-supervised visual representation learning from hierarchical grouping. *arXiv preprint arXiv:2012.03044*, 2020. 2