

DetCo: Unsupervised Contrastive Learning for Object Detection

Enze Xie^{1*}, Jian Ding^{3*}, Wenhai Wang⁴, Xiaohang Zhan⁵,
Hang Xu², Peize Sun¹, Zhenguo Li², Ping Luo¹

¹The University of Hong Kong ²Huawei Noah’s Ark Lab

³Wuhan University ⁴Nanjing University ⁵Chinese University of Hong Kong

Abstract

We present *DetCo*, a simple yet effective self-supervised approach for object detection. Unsupervised pre-training methods have been recently designed for object detection, but they are usually deficient in image classification, or the opposite. Unlike them, *DetCo* transfers well on downstream instance-level dense prediction tasks, while maintaining competitive image-level classification accuracy. The advantages are derived from (1) multi-level supervision to intermediate representations, (2) contrastive learning between global image and local patches. These two designs facilitate discriminative and consistent global and local representation at each level of feature pyramid, improving detection and classification, simultaneously.

Extensive experiments on VOC, COCO, Cityscapes, and ImageNet demonstrate that *DetCo* not only outperforms recent methods on a series of 2D and 3D instance-level detection tasks, but also competitive on image classification. For example, on ImageNet classification, *DetCo* is 6.9% and 5.0% top-1 accuracy better than *InsLoc* and *DenseCL*, which are two contemporary works designed for object detection. Moreover, on COCO detection, *DetCo* is 6.9 AP better than *SwAV* with Mask R-CNN C4. Notably, *DetCo* largely boosts up *Sparse R-CNN*, a recent strong detector, from 45.0 AP to 46.5 AP (+1.5 AP), establishing a new SOTA on COCO.

1. Introduction

Self-supervised learning of visual representation is an essential problem in computer vision, facilitating many downstream tasks such as image classification, object detection, and semantic segmentation [23, 35, 43]. It aims to provide models pre-trained on large-scale unlabeled data for downstream tasks. Previous methods focus on designing different pretext tasks. One of the most promising directions among them is contrastive learning [32], which transforms one im-

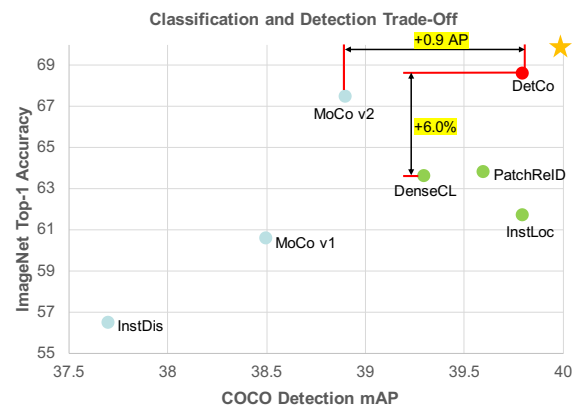


Figure 1. **Transfer accuracy on Classification and Detection.** *DetCo* achieves the best performance trade-off on both classification and detection. For example, *DetCo* outperforms its strong baseline, MoCo v2 [5], by 0.9 AP on COCO detection. Moreover, *DetCo* is significant better than recent work *e.g.* DenseCL [39], *InsLoc* [41], PatchReID [8] on ImageNet classification while also has advantages on object detection. Note that these three methods are concurrent work and specially designed for object detection (mark with **green**). The yellow asterisk indicates that a desired method should have both high performance in detection and classification.

age into multiple views, minimizes the distance between views from the same image, and maximizes the distance between views from different images in a feature map.

In the past two years, some methods based on contrastive learning and online clustering, *e.g.* MoCo v1/v2 [19, 5], BYOL [18], and SwAV [3], have achieved great progress to bridge the performance gap between unsupervised and fully-supervised methods for image classification. However, their transferring ability on object detection is not satisfactory. Concurrent to our work, recently DenseCL [39], *InsLoc* [41] and PatchReID [8] also adopt contrastive learning to design detection-friendly pretext tasks. Nonetheless, these methods only transfer well on object detection but sacrifice image classification performance, as shown in Figure 1 and Table 1. So, it is challenging to design a pretext task that can reconcile instance-level detection and image

*equal contribution

Method	Place	ImageNet Cls.		COCO Det.	Cityscapes Seg.
		Top-1	Top-5	mAP	mIoU
MoCo v1[19]	CVPR'20	60.6	-	38.5	75.3
MoCo v2[5]	Arxiv	67.5	-	38.9	75.7
InstLoc[41]	CVPR'21	61.7	-	39.8	-
DenseCL[39]	CVPR'21	63.6	85.8	39.3	75.7
PatchReID[8]	Arxiv	63.8	85.6	39.6	76.6
DetCo	-	68.6	88.5	39.8	76.5

Table 1. **Classification and Detection trade-off for recent detection-friendly self-supervised methods.** Compared with concurrent InstLoc[41], DenseCL[39] and PatchReID[8], DetCo is significantly better by 6.9%, 5.0% and 4.8% on ImageNet classification. Moreover, DetCo is also on par with these methods on dense prediction tasks, achieving best trade-off.

classification.

We hypothesize that there is no unbridgeable gap between image-level classification and instance-level detection. Intuitively, image classification recognizes global instance from a single high-level feature map, while object detection recognizes local instance from multi-level feature pyramids. From this perspective, it is desirable to build instance representation that are (1) discriminative at each level of feature pyramid (2) consistent for both global image and local patch (*a.k.a* sliding windows). However, existing unsupervised methods overlook these two aspects. Therefore, detection and classification cannot mutually improve.

In this work, we present DetCo, which is a contrastive learning framework beneficial for instance-level detection tasks while maintaining competitive image classification transfer accuracy. DetCo contains (1) multi-level supervision on features from different stages of the backbone network. (2) contrastive learning between global image and local patches. Specifically, the multi-level supervision directly optimizes the features from each stage of backbone network, ensuring strong discrimination in each level of pyramid features. This supervision leads to better performance for dense object detectors by multi-scale prediction. The global and local contrastive learning guides the network to learn consistent representation on both image-level and patch-level, which can not only keep each local patch highly discriminative but also promote the whole image representation, benefiting both object detection and image classification.

DetCo achieves state-of-the-art transfer performance on various 2D and 3D instance-level detection tasks *e.g.* VOC and COCO object detection, semantic segmentation and DensePose. Moreover, the performance of DetCo on ImageNet classification and VOC SVM classification is still very competitive. For example, as shown in Figure 1 and Table 1, DetCo improves MoCo v2 on both classification and dense prediction tasks. DetCo is significant better than DenseCL [39], InstLoc [41] and PatchReID [8] on ImageNet classification by 6.9%, 5.0% and 4.8% and slightly better on object detection and semantic segmentation. Please

note DenseCL, InstLoc and PatchReID are three concurrent works which are designed for object detection but sacrifice classification. Moreover, DetCo boosts up Sparse R-CNN [37], which is a recent end-to-end object detector without q, from a very high baseline 45.0 AP to 46.5 AP (+1.5 AP) on COCO dataset with ResNet-50 backbone, establishing a new state-of-the-art detection result. In the 3D task, DetCo outperforms ImageNet supervised methods and MoCo v2 in all metrics on COCO DensePose, especially +1.4 on AP₅₀.

Overall, the main **contributions** of this work are three-fold:

- We introduce a simple yet effective self-supervised pretext task, named DetCo, which is beneficial for instance-level detection tasks. DetCo can utilize large-scale unlabeled data and provide a strong pre-trained model for various downstream tasks.
- Benefiting from the design of multi-level supervision and contrastive learning between global images and local patches, DetCo successfully improves the transferring ability on object detection without sacrificing image classification, compared to contemporary self-supervised counterparts.
- Extensive experiments on PASCAL VOC [15], COCO [28] and Cityscapes [6] show that DetCo outperforms previous state-of-the-art methods when transferred to a series of 2D and 3D instance-level detection tasks, *e.g.* object detection, instance segmentation, human pose estimation, DensePose, as well as semantic segmentation.

2. Related Work

Existing unsupervised methods for representation learning can be roughly divided into two classes, generative and discriminative. Generative methods [11, 14, 12, 2] typically rely on auto-encoding of images [38, 24, 36] or adversarial learning [17], and operate directly in pixel space. Therefore, most of them are computationally expensive, and the pixel-level details required for image generation may not be necessary for learning high-level representations.

Among discriminative methods [9, 5], self-supervised contrastive learning [5, 19, 5, 3, 18] currently achieved state-of-the-art performance, arousing extensive attention from researchers. Unlike generative methods, contrastive learning avoids the computation-consuming generation step by pulling representations of different views of the same image (*i.e.*, positive pairs) close, and pushing representations of views from different images (*i.e.*, negative pairs) apart. Chen *et al.* [5] developed a simple framework, termed SimCLR, for contrastive learning of visual representations. It learns features by contrasting images after a composition of data augmentations. After that, He *et al.* [19] and Chen

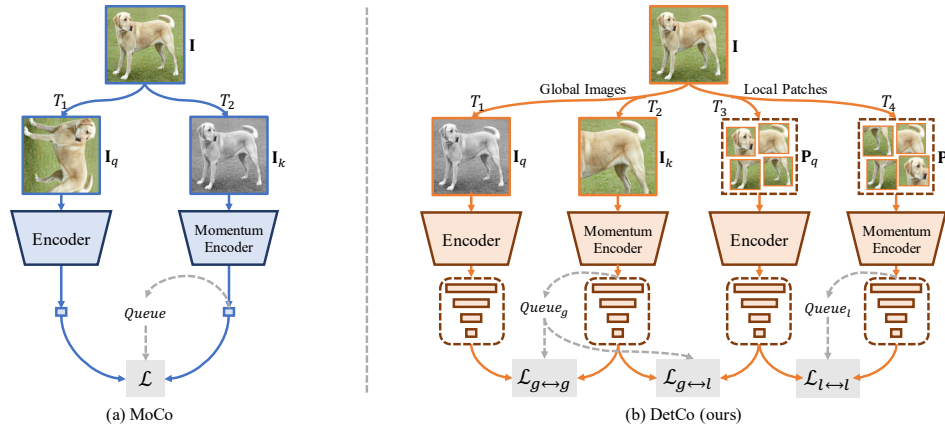


Figure 2. **The overall pipeline of DetCo compared with MoCo [19].** (a) is MoCo’s framework, which only considers the single high-level feature and learning contrast from a global perspective. (b) is our DetCo, which learns representation with multi-level supervision and adds two additional local patch sets for input, building contrastive loss cross the global and local views. Note that “ T ” means image transforms. “ $Queue_{g/l}$ ” means different memory banks [40] for global/local features.

et al. [5] proposed MoCo and MoCo v2, using a moving average network (momentum encoder) to maintain consistent representations of negative pairs drawn from a memory bank. Recently, SwAV [3] introduced online clustering into contrastive learning, without requiring to compute pairwise comparisons. BYOL [18] avoided the use of negative pairs by bootstrapping the outputs of a network iteratively to serve as targets for an enhanced representation.

Moreover, earlier methods rely on all sorts of pretext tasks to learn visual representations. Relative patch prediction [9, 10], colorizing gray-scale images [42, 25], image inpainting [33], image jigsaw puzzle [31], image super-resolution [26], and geometric transformations [13, 16] have been proved to be useful for representation learning.

Nonetheless, most of the aforementioned methods are specifically designed for image classification while neglecting object detection. Concurrent to our work, recently DenseCL [39], InsLoc [41] and PatchReID [8] design pretext tasks for object detection. However, their transferring performance is poor on image classification. Our work focuses on designing a better pretext task which is not only beneficial for instance-level detection, but also maintains strong representation for image classification.

3. Methods

In this section, we first briefly introduce the overall architecture of the proposed DetCo showed in Figure 2. Then, we present the design of multi-level supervision that keeps features at multiple stages discriminative. Next, we introduce global and local contrastive learning to enhance global and local representation. Finally, we provide the implementation details of DetCo.

3.1. DetCo Framework

DetCo is a simple pipeline designed mainly based on a strong baseline MoCo v2. It composes of a backbone network, a series of MLP heads and memory banks. The setting of MLP head and memory banks are same as MoCo v2 for simplicity. The overall architecture of DetCo is illustrated in Figure 2.

Specifically, DetCo has two simple and effective designs which are different from MoCo v2. (1) multi-level supervision to keep features at multiple stages discriminative. (2) global and local contrastive learning to enhance both global and local feature representation. The above two different designs make DetCo not only successfully inherit the advantages of MoCo v2 on image classification but also transferring much stronger on instance-level detection tasks.

The complete loss function of DetCo can be defined as follows:

$$\mathcal{L}(\mathbf{I}_q, \mathbf{I}_k, \mathbf{P}_q, \mathbf{P}_k) = \sum_{i=1}^4 w_i \cdot (\mathcal{L}_{g \leftrightarrow g}^i + \mathcal{L}_{l \leftrightarrow l}^i + \mathcal{L}_{g \leftrightarrow l}^i), \quad (1)$$

where \mathbf{I} represents a global image and \mathbf{P} represents the local patch set. Eqn. 1 is a multi-stage contrastive loss. In each stage, there are three cross local and global contrastive losses. We will describe the multi-level supervision $\sum_{i=1}^4 w_i \cdot \mathcal{L}^i$ in Section 3.2, the global and local contrastive learning $\mathcal{L}_{g \leftrightarrow g}^i + \mathcal{L}_{l \leftrightarrow l}^i + \mathcal{L}_{g \leftrightarrow l}^i$ in Section 3.3.

3.2. Multi-level Supervision

Modern object detectors predict objects in different levels, *e.g.* RetinaNet and Faster R-CNN FPN. They require the features at each level to keep strong discrimination. To meet the above requirement, we make a simple yet effective modification to the original MoCo baseline.

Specifically, we feed one image to a standard backbone ResNet-50, and it outputs features from different stages, termed Res2, Res3, Res4, Res5. Unlike MoCo that only uses Res5, we utilize all levels of features to calculate contrastive losses, ensuring that each stage of the backbone produces discriminative representations.

Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, it is first transformed to two views of the image \mathbf{I}_q and \mathbf{I}_k with two transformations randomly drawn from a set of transformations on global views, termed \mathcal{T}_g . We aim at training an encoder_q together with an encoder_k with the same architecture, where encoder_k update weights using a momentum update strategy [19]. The encoder_q contains a backbone and four global MLP heads to extract features from four levels. We feed \mathbf{I}_q to the backbone $b_q^\theta(\cdot)$, with parameters θ that extracts features $\{f_2, f_3, f_4, f_5\} = b_q^\theta(\mathbf{I}_q)$, where f_i means the feature from the i -th stage. After obtaining the multi-level features, we append four global MLP heads $\{mlp_q^2(\cdot), mlp_q^3(\cdot), mlp_q^4(\cdot), mlp_q^5(\cdot)\}$ whose weights are non-shared. As a result, we obtain four global representations $\{q_2^g, q_3^g, q_4^g, q_5^g\} = \text{encoder}_q(\mathbf{I}_q)$. Likewise, we can easily get $\{k_2^g, k_3^g, k_4^g, k_5^g\} = \text{encoder}_k(\mathbf{I}_k)$.

MoCo uses InfoNCE to calculate contrastive loss, formulated as:

$$\mathcal{L}_{g \leftrightarrow g}(\mathbf{I}_q, \mathbf{I}_k) = -\log \frac{\exp(q^g \cdot k_+^g / \tau)}{\sum_{i=0}^K \exp(q^g \cdot k_i^g / \tau)}, \quad (2)$$

where τ is a temperature hyper-parameter [40]. We extend it to multi-level contrastive losses for multi-stage features, formulated as:

$$Loss = \sum_{i=1}^4 w_i \cdot \mathcal{L}_{g \leftrightarrow g}^i, \quad (3)$$

where w is the loss weight, and i indicates the current stage. Inspired by the loss weight setting in PSPNet [43], we set the loss weight of shallow layers to be smaller than deep layers. In addition, we build an individual memory bank $queue_i$ for each layer. In the appendix, we provide the pseudo-code of intermediate contrastive loss.

3.3. Global and Local Contrastive Learning

Modern object detectors repurpose classifiers on local regions (*a.k.a* sliding windows) to perform detection. So, it requires each local region to be discriminative for instance classification. To meet the above requirement, we develop global and local contrastive learning to keep consistent instance representation on both patch set and the whole image. This strategy takes advantage of image-level representation to enhance instance-level representation, vice versa.

In detail, we first transform the input image into 9 local patches using jigsaw augmentation, the augmentation details are shown in section 3.4. These patches pass through

the encoder, and then we can get 9 local feature representation. After that, we combine these features into one feature representation by a MLP head, and build a cross global-and-local contrastive learning.

Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, first it is transformed into two local patch set \mathbf{P}_q and \mathbf{P}_k by two transformations selected from a local transformation set, termed \mathcal{T}_l . There are 9 patches $\{p_1, p_2, \dots, p_9\}$ in each local patch set. We feed the local patch set to backbone and get 9 features $F_p = \{f_{p1}, f_{p2}, \dots, f_{p9}\}$ at each stage. Taking a stage as an example, we build a MLP head for local patch, denoted as $mlp_{local}(\cdot)$, which does not share weights with $mlp_{global}(\cdot)$ in section 3.2. Then, F_p is concatenated and fed to the local patch MLP head to get final representation q^l . Likewise, we can use the same approach to get k^l .

The contrastive cross loss has two parts: the global \leftrightarrow local contrastive loss and the local \leftrightarrow local contrastive loss. The global \leftrightarrow local contrastive loss can be written as:

$$\mathcal{L}_{g \leftrightarrow l}(\mathbf{P}_q, \mathbf{I}_k) = -\log \frac{\exp(q^l \cdot k_+^g / \tau)}{\sum_{i=0}^K \exp(q^l \cdot k_i^g / \tau)}. \quad (4)$$

Similarly, the local \leftrightarrow local contrastive loss can be formulated as:

$$\mathcal{L}_{l \leftrightarrow l}(\mathbf{P}_q, \mathbf{P}_k) = -\log \frac{\exp(q^l \cdot k_+^l / \tau)}{\sum_{i=0}^K \exp(q^l \cdot k_i^l / \tau)}. \quad (5)$$

By learning representations between global image and local patches, the instance discrimination of image-level and instance-level are mutually improved. As a result, both the detection and classification performance boost up.

3.4. Implementation Details

We use OpenSelfSup¹ as the codebase. We use a batch size of 256 with 8 Tesla V100 GPUs per experiment. We follow the most hyper-parameters settings of MoCo v2. For data augmentation, the global view augmentation is almost the same as MoCo v2 [5] with random crop and resized to 224×224 with a random horizontal flip, gaussian blur and color jittering related to brightness, contrast, saturation, hue and grayscale. Rand-Augmentation[7] is also used on global view. The local patch augmentation follows PIRL [30]. First, a random region is cropped with at least 60% of the image and resized to 255×255 , followed by random flip, color jitter and blur, sharing the same parameters with global augmentation. Then we divide the image into 3×3 grids and randomly shuffle them; each grid is 85×85 . A random crop is applied on each patch to get 64×64 to avoid continuity between patches. Finally, we obtain nine randomly shuffled patches. For a fair comparison, we use standard ResNet-50 [23] for all experiments. Unless other

¹<https://github.com/open-mmlab/OpenSelfSup>

specified, we pre-train 200 epochs on ImageNet for a fair comparison.

4. Experiments

We evaluate DetCo on a series of 2D and 3D dense prediction tasks, *e.g.* PASCAL VOC detection, COCO detection, instance segmentation, 2D pose estimation, DensePose and Cityscapes instance and semantic segmentation. We see that DetCo outperforms existing self-supervised and supervised methods.

4.1. Object Detection

Experimental Setup. We choose three representative detectors: Faster R-CNN [35], Mask R-CNN [22] RetinaNet [27], and a recent strong detector: Sparse R-CNN [37]. Mask R-CNN is two-stage and RetinaNet is one stage detector. Sparse R-CNN is an end-to-end detector without NMS, and it is also state-of-the-art with high mAP on COCO. Our training settings are the same as MoCo [19] for a fair comparison, including using “SyncBN” [34] in backbone and FPN.

PASCAL VOC. As shown in Table 9 and Figure 3, MoCo v2 is a strong baseline, which has already surpassed other unsupervised learning methods in VOC detection. However, our DetCo consistently outperforms the MoCo v2 at 200 epochs and 800 epochs. More importantly, with only 100 epoch pre-training, DetCo achieves almost the same performance as MoCo v2-800ep (800 epoch pre-training). Finally, DetCo-800ep establishes the new state-of-the-art, 58.2 in mAP and 65.0 in AP_{75} , which brings 4.7 and 6.2 improvements in AP and AP_{75} respectively, compared with supervised counterpart. The improvements on the more stringent AP_{75} are much larger than the AP, indicating that the intermediate and patch contrasts are beneficial to the localization.

COCO with 1× and 2× Schedule. Table 3 shows the Mask RCNN [22] results on 1× schedule, DetCo outperforms MoCo v2 baseline by 0.9 and 1.2 AP for R50-C4 and R50-FPN backbones. It also outperforms the supervised counterpart by 1.6 and 1.2 AP for R50-C4 and R50-FPN respectively. The results of 2× schedule is in Appendix. The column 2-3 of Table 7 shows the results of one stage detector RetinaNet. DetCo pretrain is 1.0 and 1.2 AP better than supervised methods and MoCo v2. DetCo is also 1.3 higher than MoCov2 on AP_{50} with 1× schedule.

COCO with Few Training Iterations. COCO is much larger than PASCAL VOC in the data scale. Even training from scratch [20] can get a satisfactory result. To verify the effectiveness of unsupervised pre-training, we conduct experiments on extremely stringent conditions: only train detectors with 12k iterations ($\approx 1/7\times$ vs. 90k-1× schedule). The 12k iterations make detectors heavily under-trained and

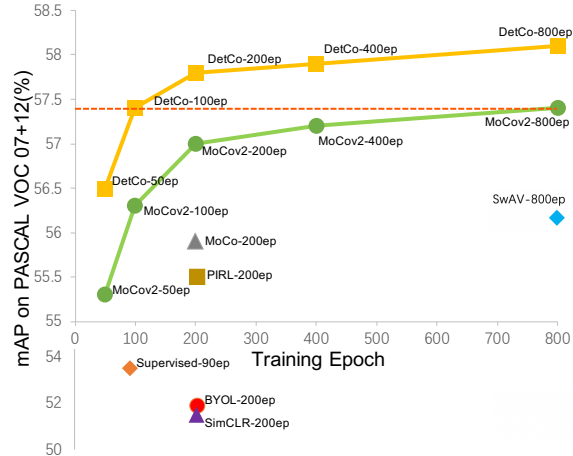


Figure 3. **Comparisons of mAP on PASCAL VOC 07+12 object detection.** For different pre-training epoches, we see that DetCo consistently outperforms MoCo v2[5], which is a strong competitor on VOC compared to other methods. For example, DetCo-100ep already achieves similar mAP compared to MoCov2-800ep. Moreover, DetCo-800ep achieves state-of-the-art and outperforms other counterparts.

far from converge, as shown in Table 2 and Table 7 column 1. Under this setting, for Mask RCNN-C4, DetCo exceeds MoCo v2 by 3.8 AP in AP_{50}^{bb} and outperforms supervised methods in all metrics, which indicates DetCo can significantly fasten the training convergence. For Mask RCNN-FPN and RetinaNet, DetCo also has significant advantages over MoCo v2 and supervised counterpart.

COCO with Semi-Supervised Learning. Transferring to a small dataset has more practical value. As indicated in the [21], when only use 1% data of COCO, the train from scratch’s performance can not catch up in mAP with ones that have pre-trained initialization. To verify the effectiveness of self-supervised learning on a small-scale dataset, we randomly sample 1%, 2%, 5%, 10% data to fine-tune the RetinaNet. For all the settings, we fine-tune the detectors with 12k iterations to avoid overfitting. Other settings are the same as COCO 1× and 2× schedule.

The results for RetinaNet with 1%, 2%, 5%, 10% are shown in Table 8. We find that in four semi-supervised settings, DetCo significantly surpasses the supervised counterpart and MoCo v2 strong baseline. For instance, DetCo outperforms the supervised method by 2.3 AP and MoCo v2 by 1.9 AP when using 10% data. These results show that the DetCo pre-trained model is also beneficial for semi-supervised object detection. More results for Mask R-CNN with 1%, 2%, 5%, and 10% data are in the appendix.

DetCo + Recent Advanced Detector. In table 4, we find that DetCo can improve Sparse R-CNN[37] with 1.5 mAP

Method	Mask R-CNN R50-C4 COCO 12k						Mask R-CNN R50-FPN COCO 12k					
	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}
Rand Init	7.9	16.4	6.9	7.6	14.8	7.2	10.7	20.7	9.9	10.3	19.3	9.6
Supervised	27.1	46.8	27.6	24.7	43.6	25.3	28.4	48.3	29.5	26.4	45.2	25.7
InsDis[40]	25.8(-1.3)	43.2(-3.6)	27.0(-0.6)	23.7(-1.0)	40.4(-3.2)	24.5(-0.8)	24.2(-4.2)	41.5(-6.8)	25.1(-4.4)	22.8(-3.6)	38.9(-6.3)	23.7(-2.0)
PIRL[30]	25.5(-1.6)	42.6(-4.2)	26.8(-0.8)	23.2(-1.5)	39.9(-3.7)	23.9(-1.4)	23.7(-4.7)	40.4(-7.9)	24.4(-5.1)	22.1(-4.3)	37.9(-7.3)	22.7(-3.0)
SwAV[3]	16.5(-10.6)	35.2(-11.6)	13.5(-14.1)	16.1(-8.6)	32.0(-11.6)	14.6(-10.7)	25.5(-2.9)	46.2(-2.1)	25.4(-4.1)	24.8(-1.6)	43.5(-1.7)	25.3(-0.4)
MoCo[19]	26.9(-0.2)	44.5(-2.3)	28.2(+0.6)	24.6(-0.1)	41.8(-1.8)	25.6(+0.3)	25.6(-2.8)	43.4(-4.9)	26.6(-2.9)	23.9(-2.5)	40.8(-4.4)	24.8(-0.9)
MoCov2[5]	27.6(+0.5)	45.3(+1.5)	28.9(+1.3)	25.1(+0.4)	42.6(+1.0)	26.3(+1.0)	26.6(-1.8)	44.9(-3.4)	27.7(-1.8)	24.8(-1.6)	42.1(-3.1)	25.7(0.0)
DetCo	29.8(+2.7)	49.1(+2.3)	31.4(+3.8)	26.9(+2.2)	46.0(+2.4)	27.9(+2.6)	29.6(+1.2)	49.4(+1.1)	31.0(+1.5)	27.6(+1.2)	46.6(+1.4)	28.7(+3.0)

Table 2. **Object detection and instance segmentation fine-tuned on COCO.** All methods are pretrained 200 epochs on ImageNet. **Green** means increase and **gray** means decrease. DetCo outperforms all supervised and unsupervised counterparts.

Method	Mask R-CNN R50-C4 COCO 90k						Mask R-CNN R50-FPN COCO 90k					
	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}
Rand Init	26.4	44.0	27.8	29.3	46.9	30.8	31.0	49.5	33.2	28.5	46.8	30.4
Supervised	38.2	58.2	41.2	33.3	54.7	35.2	38.9	59.6	42.7	35.4	56.5	38.1
InsDis[40]	37.7(-0.5)	57.0(-1.2)	40.9(-0.3)	33.0(-0.3)	54.1(-0.6)	35.2(0.0)	37.4(-1.5)	57.6(-2.0)	40.6(-2.1)	34.1(-1.3)	54.6(-1.9)	36.4(-1.7)
PIRL[30]	37.4(-0.8)	56.5(-1.7)	40.2(-1.0)	32.7(-0.6)	53.4(-1.3)	34.7(-0.5)	37.5(-1.4)	57.6(-2.0)	41.0(-1.7)	34.0(-1.4)	54.6(-1.9)	36.2(-1.9)
SwAV[3]	32.9(-5.3)	54.3(-3.9)	34.5(-6.7)	29.5(-3.8)	50.4(-4.3)	30.4(-4.8)	38.5(-0.4)	60.4(+0.8)	41.4(-1.3)	35.4(0.0)	57.0(+0.5)	37.7(-0.4)
MoCo[19]	38.5(+0.3)	58.3(+0.1)	41.6(+0.4)	33.6(+0.3)	54.8(+0.1)	35.6(+0.4)	38.5(+0.4)	58.9(+0.7)	42.0(+0.7)	35.1(-0.3)	55.9(-0.6)	37.7(-0.4)
MoCov2[5]	38.9(+0.7)	58.4(+0.2)	42.0(+0.8)	34.2(+0.9)	55.2(+0.5)	36.5(+1.3)	38.9(0.0)	59.4(-0.2)	42.4(-0.3)	35.5(+0.1)	56.5(0.0)	38.1(0.0)
DetCo	39.8(+1.6)	59.7(+1.5)	43.0(+1.8)	34.7(+1.4)	56.3(+1.6)	36.7(+1.5)	40.1(+1.2)	61.0(+1.4)	43.9(+1.2)	36.4(+1.0)	58.0(+1.5)	38.9(+0.8)

Table 3. **Object detection and instance segmentation fine-tuned on COCO.** All methods are pretrained 200 epochs on ImageNet. DetCo outperforms all supervised and unsupervised counterparts.

	AP	AP ₅₀	AP ₇₅	AP _s	AP _m	AP _l
Supervised	45.0	64.1	49.0	27.7	47.5	59.6
DetCo	46.5	65.7	50.8	30.8	49.5	59.7

Table 4. **DetCo vs. Supervised pre-train** on Sparse R-CNN. DetCo largely improves 1.5 mAP and 3.1 AP_s.

Method	Epoch	AP ^{dp}	AP ₅₀ ^{dp}	AP ₇₅ ^{dp}
Rand Init	-	40.8	78.6	37.3
Supervised	90	50.8	86.3	52.6
MoCo [19]	200	49.6(-1.2)	85.9(-0.4)	50.5(-2.1)
MoCo v2 [5]	200	50.9(+0.1)	87.2(+0.9)	52.9(+0.3)
DetCo	200	51.3(+0.5)	87.7(+1.4)	53.3(+0.7)

Table 5. **DetCo vs. other methods on Dense Pose task.** It also performs best on monocular 3D human shape prediction.

and **3.1** AP_s. Sparse R-CNN is a recent strong end-to-end detector with high performance, and DetCo can further largely boost up Sparse R-CNN’s performance and achieved the new state of the arts on COCO with **46.5** AP.

DetCo vs. Concurrent SSL Methods. InsLoc[41], DenseCL[39] and PatchReID[8] are recent works designed for object detection. They improved the performance of object detection but largely sacrifice the performance of image classification. As shown in Table 1, DetCo has significant advantages than InsLoc, DenseCL and PatchReID on ImageNet classification by **+6.9%**, **+5.0%** and **+4.8%**. Moreover, on COCO detection, DetCo is also better than these methods.

Discussions. We compared the performance when transferred to object detection at different dataset scales and fine-tuning iterations. First, DetCo largely boosts up the performance of the supervised method on small datasets (*e.g.*,

Methods	Instance Seg.		Semantic Seg.
	AP ^{mk}	AP ₅₀ ^{mk}	mIOU
Rand Init	25.4	51.1	65.3
supervised	32.9	59.6	74.6
InsDis [40]	33.0 (+0.1)	60.1 (+0.5)	73.3 (-1.3)
PIRL [30]	33.9 (+1.0)	61.7 (+2.1)	74.6 (0.0)
SwAV [3]	33.9 (+1.0)	62.4 (+2.8)	73.0 (-1.6)
MoCo [19]	32.3 (-0.6)	59.3 (-0.3)	75.3 (+0.7)
MoCov2 [5]	33.9 (+1.0)	60.8 (+1.2)	75.7 (+1.1)
DetCo	34.7 (+1.8)	63.2 (+3.6)	76.5 (+1.9)

Table 6. **DetCo vs. supervised and other unsupervised methods on Cityscapes dataset.** All methods are pretrained 200 epochs on ImageNet. We evaluate instance segmentation and semantic segmentation tasks.

PASCAL VOC). Second, DetCo also has large advantages with COCO 12k iterations. It indicates that DetCo can fasten training converge compared with other unsupervised and supervised methods. Third, even with enough data (*e.g.* COCO), DetCo still significantly improves the performance compared to other unsupervised and supervised counterparts. Finally, DetCo is friendly for detection tasks while it does not sacrifice the classification compared with concurrent SSL methods.

4.2. Segmentation and Pose Estimation

Multi-Person Pose Estimation. The last column of Table 7 shows the results of COCO keypoint detection results using Mask RCNN. DetCo also surpasses other methods in all metrics, *e.g.* **1.4** AP^{kp} and **1.5** AP₇₅^{kp} higher than supervised counterpart.

Segmentation on Cityscapes. Cityscapes is a dataset for autonomous driving in the urban street. We follow MoCo to evaluate on instance segmentation with Mask RCNN and

Method	RetinaNet R50 12k			RetinaNet R50 90k			RetinaNet R50 180k			Keypoint RCNN R50 180k		
	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP ^{kp}	AP ₅₀ ^{kp}	AP ₇₅ ^{kp}
Rand Init	4.0	7.9	3.5	24.5	39.0	25.7	32.2	49.4	34.2	65.9	86.5	71.7
Supervised	24.3	40.7	25.1	37.4	56.5	39.7	38.9	58.5	41.5	65.8	86.9	71.9
InsDis[40]	19.0(-5.3)	32.0(-8.7)	19.6(-5.5)	35.5(-1.9)	54.1(-2.4)	38.2(-1.5)	38.0(-0.9)	57.4(-1.1)	40.5(-1.0)	66.5(+0.7)	87.1(+0.2)	72.6(+0.7)
PIRL[30]	19.0(-5.3)	31.7(-9.0)	19.8(-5.3)	35.7(-1.7)	54.2(-2.3)	38.4(-1.3)	38.5(-0.4)	57.6(-0.9)	41.2(-0.3)	66.5(+0.7)	87.5(+0.6)	72.1(+0.2)
SwAV[3]	19.7(-4.6)	34.7(-6.0)	19.5(-5.6)	35.2(-2.2)	54.9(-1.6)	37.5(-2.2)	38.6(-0.3)	58.8(+0.3)	41.1(-0.4)	66.0(+0.2)	86.9(0.0)	71.5(-0.4)
MoCo[19]	20.2(-4.1)	33.9(-6.8)	20.8(-4.3)	36.3(-1.1)	55.0(-1.5)	39.0(-0.7)	38.7(-0.2)	57.9(-0.6)	41.5(0.0)	66.8(+1.0)	87.4(+0.5)	72.5(+0.6)
MoCov2[5]	22.2(-2.1)	36.9(-3.8)	23.0(-2.1)	37.2(-0.2)	56.2(-0.3)	39.6(-0.1)	39.3(+0.4)	58.9(+0.4)	42.1(+0.6)	66.8(+1.0)	87.3(+0.4)	73.1(+1.2)
DetCo	25.3(+1.0)	41.6(+0.9)	26.5(+1.4)	38.4(+1.0)	57.8(+1.3)	41.2(+1.5)	39.7(+0.8)	59.3(+0.8)	42.6(+1.1)	67.2(+1.4)	87.5(+0.6)	73.4(+1.5)

Table 7. **One-stage object detection and keypoint detection fine-tuned on COCO.** All methods are pretrained 200 epochs on ImageNet. DetCo outperforms all supervised and unsupervised counterparts.

Method	RetinaNet R50 COCO 1% Data			RetinaNet R50 COCO 2% Data			RetinaNet R50 COCO 5% Data			RetinaNet R50 COCO 10% Data		
	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅	AP	AP ₅₀	AP ₇₅
Rand Init	1.4	3.5	1.0	2.5	5.6	2.0	3.6	7.4	3.0	3.7	7.5	3.2
Supervised	8.2	16.2	7.2	11.2	21.7	10.3	16.5	30.3	15.9	19.6	34.5	19.7
MoCo[19]	7.0(-1.2)	13.5(-2.7)	6.5(-0.7)	10.3(-0.9)	19.2(-2.5)	9.7(-0.6)	15.0(-1.5)	27.0(-3.3)	14.9(-1.0)	18.2(-1.4)	31.6(-2.9)	18.4(-1.3)
MoCo v2[5]	8.4(+0.2)	15.8(-0.4)	8.0(+0.8)	12.0(+0.8)	21.8(+0.1)	11.5(+1.2)	16.8(+0.3)	29.6(-0.7)	16.8(+0.9)	20.0(+0.4)	34.3(-0.2)	20.2(+0.5)
DetCo	9.9(+1.7)	19.3(+3.1)	9.1(+1.9)	13.5(+2.3)	25.1(+3.4)	12.7(+2.4)	18.7(+2.2)	32.9(+2.6)	18.7(+2.8)	21.9(+2.3)	37.6(+3.1)	22.3(+2.6)

Table 8. **Semi-Supervised one-stage detection fine-tuned on COCO 1%, 2%, 5% and 10% data.** All methods are pretrained 200 epochs on ImageNet. DetCo is significant better than supervised / unsupervised counterparts in all metrics.

Method	Epoch	AP	AP ₅₀	AP ₇₅
Rand Init	-	33.8	60.2	33.1
Supervised	90	53.5	81.3	58.8
InsDis [40]	200	55.2(+1.7)	80.9(-0.4)	61.2(+2.4)
PIRL [30]	200	55.5(+2.0)	81.0(-0.3)	61.3(+2.5)
SwAV [3]	800	56.1(+2.6)	82.6(+1.3)	62.7(+3.9)
MoCo [19]	200	55.9(+2.4)	81.5(+0.2)	62.6(+3.8)
MoCov2 [5]	200	57.0(+3.5)	82.4(+1.1)	63.6(+4.8)
MoCov2 [5]	800	57.4(+3.9)	82.5(+1.2)	64.0(+5.2)
DetCo	100	57.4(+3.9)	82.5(+1.2)	63.9(+5.1)
	200	57.8(+4.3)	82.6(+1.3)	64.2(+5.4)
	800	58.2(+4.7)	82.7(+1.4)	65.0(+6.2)

Table 9. **Object Detection finetuned on PASCAL VOC07+12 using Faster RCNN-C4.** DetCo-100ep is on par with previous state-of-the-art, and DetCo-800ep achieves the best performance.

Method	Epoch	ImageNet		VOC07
		Top1	Top5	Acc
Jigsaw [31]	-	44.6	-	64.5
Rotation [16]	-	55.4	-	63.9
InsDis [40]	200	56.5	-	76.6
LocalAgg [44]	200	58.8	-	-
PIRL [30]	800	63.6	-	81.1
SimCLR [4]	1000	69.3	89.0	-
BYOL [18]	1000	74.3	91.6	-
SwAV [3]	200	72.7	-	87.6
MoCo [19]	200	60.6	-	79.2
MoCov2 [5]	200	67.5	-	84.1
DetCo	200	68.6	88.5	85.1

Table 10. **Comparison of ImageNet Linear Classification and VOC SVM Classification.** Although DetCo is designed for detection, it is also robust and competitive on classification task, and it substantially exceeds MoCov2 baseline by 1.1%.

semantic segmentation with FCN-16s [29]. The results are shown in Table 6.

Although its domain is totally different from COCO, DetCo pre-training can still significantly improve the transfer performance. On instance segmentation, DetCo outperforms supervised counterpart and MoCo v2 by **3.6** and **2.4** on AP₅₀^{mk}. On semantic segmentation, DetCo is also 1.9% and 0.8% higher than supervised method and MoCo v2.

DensePose. Estimating 3D shape from a single 2D image is challenging. It can serve as a good testbed for self-supervised learning methods, so we evaluate DetCo on COCO DensePose [1] task and find DetCo also transfer well on this task. As shown in Table 5, DetCo significantly outperforms ImageNet supervised method and MoCo v2 in all metrics, especially **+1.4** on AP₅₀.

4.3. Image Classification

We follow the standard settings: ImageNet linear classification and VOC SVM classification. For ImageNet linear classification, the training epoch is 100, and the learning rate is 30, the same as MoCo. Our DetCo also outperforms its strong baseline MoCo v2 by **+1.1%** in Top-1 Accuracy as shown in Table 10. It is also competitive on VOC SVM classification accuracy compared with state-of-the-art counterparts.

Discussion. While DetCo is designed for object Detection, its classification accuracy is still competitive. On ImageNet classification, DetCo largely outperforms concurrent DenseCL [39], PatchReID [8] and InstLoc [41], even surpasses the MoCo v2 baseline [5] by 1.1%. Although inferior to strongest classification method, SwAV, DetCo exhibits better detection accuracy. Overall, DetCo achieves best classification-detection trade-off.

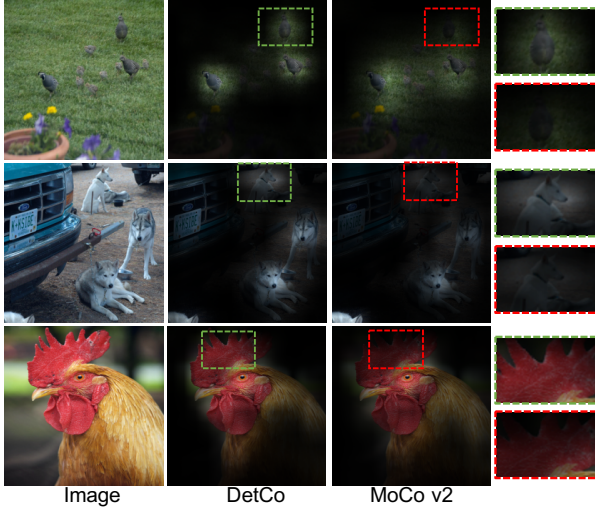


Figure 4. Attention maps generated by DetCo and MoCov2 [5]. DetCo can activate more accurate object regions in the heatmap than MoCov2. More visualization results are in Appendix.

4.4. Visualization Results

Figure 4 visualizes the attention map of DetCo and MoCo v2. We can see when there is more than one object in the image, DetCo successfully locates all the objects, while MoCo v2 fails to activate some objects. Moreover, in the last column, the attention map of DetCo is more accurate than MoCo v2 on the boundary. It reflects from the side that the localization capability of DetCo is stronger than MoCo v2, which is beneficial for object detection. More analysis, implementation details and visualization results are shown in Appendix.

4.5. Ablation Study

Experiment Settings. We conduct all the controlled experiments by training 100 epochs. We adopt MoCo v2 as our strong baseline. More ablation studies about hyper-parameters are shown in Appendix. In table 11 and 12, “MLS” means Multi-Level Supervision, and “GLC” means Global and Local Contrastive learning.

Effectiveness of Multi-level Supervision. As shown in Table 11 (a) and (b), when only adding the multi-level supervision on MoCo v2, the classification accuracy *drop* but detection performance *increase*. This is reasonable and expectable because for image classification, it is not necessary for each layer to remain discriminative, and only the final layer feature should be discriminative. However, keeping multiple level features discriminative is essential for object detection because modern detectors predict boxes in feature pyramids. We find that intermediate supervision will slightly decrease the final layer feature’s representation and

	+MLS	+GLC	Top1	Top5	mAP
(a)	×	×	64.3	85.6	56.3
(b)	✓	×	63.2 ↓	84.9 ↓	57.0 ↑
(c)	×	✓	67.1 ↑	87.5 ↑	56.8 ↑
(d)	✓	✓	66.6 ↑	87.2 ↑	57.4 ↑

Table 11. Ablation: multi-level supervision (MLS) and global and local contrastive learning (GLC). The results are evaluated on ImageNet linear classification and PASCAL VOC07+12 detection.

	+MLS	+GLC	Res2	Res3	Res4	Res5
(a)	×	×	47.1	58.2	70.9	82.1
(b)	✓	×	50.9 ↑	67.1 ↑	78.7 ↑	81.8 ↓
(c)	×	✓	47.8 ↑	59.8 ↑	75.0 ↑	84.6 ↑
(d)	✓	✓	51.6 ↑	69.7 ↑	82.5 ↑	84.3 ↑

Table 12. Ablation: multi-level supervision (MLS) and global and local contrastive learning (GLC). Accuracy of feature in different stages are evaluated by PASCAL VOC07 SVM classification.

improve the shallow layer features’ representation, which is beneficial to object detection.

We also evaluate the VOC SVM classification accuracy at four stages: Res2, Res3, Res4, Res5 to demonstrate the enhancement of the intermediate feature. As shown in Table 12 (a) and (b), the discrimination ability of shallow features vastly improves compared with baseline.

Effectiveness of Global and Local Contrastive Learning.

As shown in Table 11 (a) and (c), when only adding global and local contrastive learning, the performance of both classification and detection boosts up and surpasses MoCo v2 baseline. Moreover, as shown in Table 11 (d), GLC can further improve the detection accuracy as well as the classification accuracy. This improvement mainly benefits from the GLC successfully make network learn the image-level and patch-level representation, which is beneficial for object detection and image classification. From table 12 (a), (c) and (d), the GLC can also improve the discrimination of different stages.

5. Conclusion and Future work

This work presents DetCo, a simple yet effective pre-text task that can utilize large-scale unlabeled data to provide a pre-train model for various downstream tasks. It demonstrates state-of-the-art transfer performance on various instance-level detection tasks, e.g. VOC and COCO detection as well as semantic segmentation, while maintaining the competitive performance on image classification. We hope DetCo can serve as an alternative and useful pre-train model for dense predictions and facilitate future research.

Acknowledgement. This work was partially supported by the General Research Fund of Hong Kong No. 27208720 and the research donation from Huawei.

References

- [1] Rıza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7297–7306, 2018. 7
- [2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2
- [3] 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, 3, 6, 7
- [4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 7
- [5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2, 3, 4, 5, 6, 7, 8
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 2
- [7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 4
- [8] Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia. Unsupervised pretraining for object detection by patch reidentification. *arXiv*, 2021. 1, 2, 3, 6, 7
- [9] 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, 3
- [10] 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. 3
- [11] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2
- [12] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *Advances in Neural Information Processing Systems*, pages 10542–10552, 2019. 2
- [13] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Advances in neural information processing systems*, pages 766–774, 2014. 3
- [14] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 2
- [15] 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. 2
- [16] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Un-supervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 3, 7
- [17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2
- [18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 1, 2, 3, 7
- [19] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2, 3, 4, 5, 6, 7
- [20] 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. 5
- [21] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019. 5
- [22] 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
- [23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 4
- [24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2
- [25] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *European conference on computer vision*, pages 577–593. Springer, 2016. 3
- [26] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 3
- [27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5
- [28] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 2
- [29] 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. 7
- [30] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020. 4, 6, 7
- [31] 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. 3, 7
- [32] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1
- [33] 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. 3
- [34] 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
- [35] 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. 1, 5
- [36] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, volume 2, 2014. 2
- [37] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. *arXiv preprint arXiv:2011.12450*, 2020. 2, 5
- [38] 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
- [39] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *arXiv*, 2020. 1, 2, 3, 6, 7
- [40] 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. 3, 4, 6, 7
- [41] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. *arXiv*, 2021. 1, 2, 3, 6, 7
- [42] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pages 649–666. Springer, 2016. 3
- [43] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 4
- [44] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019. 7