

RegionCL: Exploring Contrastive Region Pairs for Self-supervised Representation Learning

Yufei Xu^{1*}, Qiming Zhang^{1*}, Jing Zhang¹, and Dacheng Tao^{2,1}

¹ University of Sydney, Australia

² JD Explore Academy, China

{yuxu7116,qzha2506}@uni.sydney.edu.au,

jing.zhang1@sydney.edu.au, dacheng.tao@gmail.com

Abstract. Self-supervised learning methods (SSL) have achieved significant success via maximizing the mutual information between two augmented views, where cropping is a popular augmentation technique. Cropped regions are widely used to construct positive pairs, while the remained regions after cropping have rarely been explored in existing methods, although they together constitute the same image instance and both contribute to the description of the category. In this paper, we make the first attempt to demonstrate the importance of both regions in cropping from a complete perspective and the effectiveness of using both regions via designing a simple yet effective pretext task called Region Contrastive Learning (RegionCL). Technically, to construct the two kinds of regions, we randomly crop a region (called the paste view) from each input image with the same size and swap them between different images to compose new images together with the remained regions (called the canvas view). Then, instead of taking the new images as a whole for positive or negative samples, contrastive pairs are efficiently constructed from the regional perspective based on the following simple criteria, i.e., each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. With minor modifications to popular SSL methods, RegionCL exploits those abundant pairs and helps the model distinguish the regions features from both canvas and paste views, therefore learning better visual representations. Experiments on ImageNet, MS COCO, and Cityscapes demonstrate that RegionCL improves MoCov2, DenseCL, and SimSiam by large margins and achieves state-of-the-art performance on classification, detection, and segmentation tasks. The code is publicly available at <https://github.com/Annbless/RegionCL>.

1 Introduction

Self-supervised learning (SSL) has become an active research topic in computer vision because of its ability to learn generalizable representations from large-scale unlabeled data and offer good performance in downstream tasks [59,60,28,6,44].

* Equal contribution.

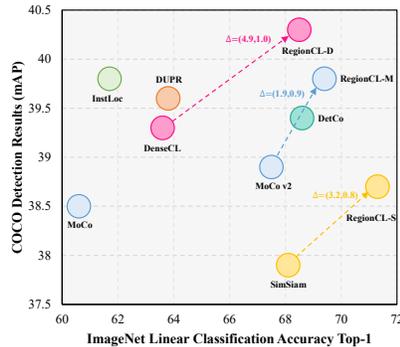


Fig. 1: Transfer results on the ImageNet [15] (classification) and MS COCO [10] (detection) datasets. Involving region-level contrastive pairs during pretraining, RegionCL helps DenseCL [47], MoCov2 [9], and SimSiam [11] achieve a better performance trade-off between image classification and object detection tasks.

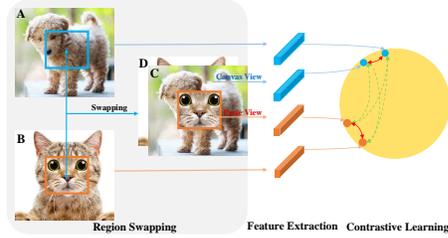


Fig. 2: Illustration of the region swapping strategy. Taking two images A and B as input, it randomly crops and swaps the paste views and generates the composite images C and D. As a result, the canvas and paste views in image C form a negative pair, while the canvas view and A (the paste view and B) are positive pairs.

Contrastive learning, one of the popular directions in SSL, has attracted a lot of attention due to its ease of use in pretext designing and capacity to generalize across various visual tasks.

Current contrastive learning methods typically use augmented views of the same image as positive pairs and maximize their mutual information. Cropping is by far the most popular augmentation technique. By randomly cropping regions from the same images and treating the cropped regions as positive pairs, the methods in [7,9,11,19,21,8] have shown promising results in image classification. Multi-crop [3,4] has been investigated as a way to improve performance even further by generating more diverse candidates and facilitating the model learning a better feature representation. Constrained cropping strategies [62,50,56,37,52] have recently been developed to ensure that two cropped views contain shared regions of a specific size and to improve models’ performance on dense prediction tasks by constructing contrastive pairs within the shared regions. These methods have achieved superior performance on a variety of visual tasks by leveraging various cropping strategies to construct contrastive pairs during pretraining.

However, the remained regions after cropping have received little attention, despite the fact that the cropped and remained regions together make up the same image instance and both contribute to the category’s description. We argue that using both regions during pretraining would help the model learn better complete visual representations of object instances, which will improve the model’s performance on downstream classification and dense prediction tasks.

Based on this motivation, we propose a simple yet effective pretext task called Region Contrastive Learning (RegionCL) to demonstrate the effectiveness of using both regions for contrastive learning. Technically, given two different images, RegionCL randomly crops a region (called the **paste view**) from each image with the same size and swaps them to compose two new images together with the remained regions (called the **canvas view**), respectively. It is worth noting that the two views that compose the new images are from different source images. Then, contrastive pairs can be constructed from the regional perspective following the simple criteria, *i.e.*, each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. In this way, RegionCL generates abundant pairs that contain not only the instance-level pairs as other methods [9,7] but also the region-level pairs, *e.g.*, the paste and canvas views in the composite images. By exploiting these pairs in popular SSL frameworks, RegionCL helps the models learn better feature representations of object instances owing to the abundant contrastive supervisory signals at both instance and region levels, delivering better performance on various downstream tasks. As shown in Figure 1, RegionCL helps MoCov2 [9], DenseCL [47], and SimSiam [11] improve their linear classification accuracy by 2%~5% on the ImageNet [15] dataset and object detection performance by 0.8~1.0 mAP on the MS COCO [10] dataset, simultaneously.

In summary, the contribution of the paper is threefold:

1. We make the first attempt to demonstrate the importance of both regions, *i.e.*, the cropped and remained regions in cropping, from a complete perspective for self-supervised learning.
2. We propose a simple yet effective pretext task, *i.e.*, RegionCL, to demonstrate the effectiveness of using both regions for learning. It is compatible with various popular SSL methods with minor modifications and improves their performance on many downstream visual tasks.
3. Extensive experimental results with MoCov2, SimSiam, and DenseCL on the ImageNet, MS COCO, and Cityscapes datasets demonstrate the effectiveness of the proposed RegionCL on classification, detection, and instance and semantic segmentation tasks.

2 Related Work

Self-supervised learning has shown great potential in learning visual representations that can generalize to a series of downstream visual tasks. Early works generate pseudo labels using specific tasks [33,25,61,35] such as image corruption and restoration, reordering, re-colorization. However, the models pretrained in these tasks may be too coupled with the designed tasks and the transfer results on other visual tasks may not be competitive.

Recently, contrastive learning [11,9,7,21,19,42,8] has made rapid progress and shown promising transfer performance. Typically, they take augmented views from the same (different) images as positive (negative) pairs and learn to pull the features from positive pairs while pushing away those from the negative

pairs via a contrastive loss. Among the augmentation techniques, cropping plays an important role in improving the performance, as shown in [7]. Taking the cropped augmented views as input, SimCLR [7] obtains superior results on image classification. MoCo [21,9] utilizes a momentum encoder to better utilize the cropped views during pretraining, as it provides consistent optimization direction. However, as cropping at a single resolution may not provide enough descriptions of the target object, a multi-crop strategy is explored in [3,4] by fusing several cropped views at different resolutions. Such a strategy helps the models learn a better feature representation at different scales and boost their performance on the image classification task.

On the other hand, [47,52,37,36] focus on advancing the performance on dense prediction tasks by establishing dense correspondences between the augmented cropped views. Some methods [62,37,5,30] further design constrained cropping strategies during pretraining to improve the transfer results on detection, *e.g.*, they require the two cropped views have some shared regions and attract the dense positive features within the shared regions based on explicit spatial correspondences. By exploring different properties of cropping-based augmented views, these methods obtain superior performance. However, the remained regions after cropping have rarely been explored. Different from them, we make the first attempt to investigate the importance of both regions in this study via adopting a simple region swapping strategy to generate abundant contrastive pairs at both instance and region levels for contrastive learning (RegionCL), from which the model can learn better visual representations of object instances.

Although several methods also explore region-level contrastive learning, they have not yet explored the complementary remained regions after cropping, *i.e.*, the canvas view in our paper. For example, SCRL [37], DUPR [16], and MaskCo [62] incorporate bounding boxes generation and alignment between the shared area of two cropped views during pretraining. InstLoc [56] further introduces anchors with bounding boxes augmentations to boost the transfer results on dense prediction tasks at the cost of decreased image classification accuracy. DetCo [50] designs delicate cropping strategies to generate separate patches at different resolutions and uses extra memory banks to capture patch features. Without the requirement of extra information such as bounding boxes alignment, RegionCL adopts the most simple strategy to demonstrate that using both regions for contrastive learning isolated helps to boost the SSL methods performance without bells and whistles. The simple task is compatible with popular SSL frameworks, *e.g.*, we validate the effectiveness of using both regions for training with MoCov2 [9], SimSiam [11], and DenseCL [47], with only minor modifications to them. The theoretical analysis of these works have been well studied in [43,40,45,46] and is beyond this paper’s scope. It is also noteworthy that although the adopted swapping strategy is similar to co-current methods like UnMix [38]/HEXA [26]/InsCon [57], they do not explore region-level pairs, *i.e.*, they still treat the composite images as a whole from the global perspective. RegionCL explores both region- and global-level pairs from the composite images, which is more efficient and obtains better performance on various vision tasks.

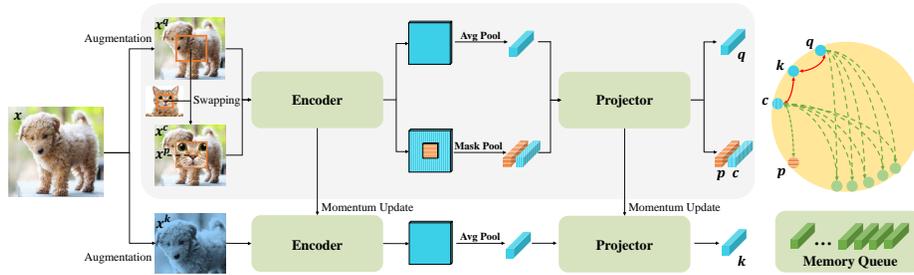


Fig. 3: Illustration of the proposed RegionCL with the MoCov2 framework, *i.e.*, RegionCL-M. Taking the two augmented views x^q , x^k as inputs, RegionCL employs region swapping among the batch of x^q to generate the composite images with paste views x^p and canvas views x^c . Then, mask pooling is used to extract the features belonging to the paste and canvas views, respectively. The resultant region-level features (with stripes in the figure) are batched with the instance-level features and processed by the projector.

3 Method

3.1 The region swapping strategy

Different from current methods that only use the cropped regions, we take both the cropped and remained regions into consideration for self-supervised learning. Given two different images, we randomly crop a region with the same size from each image and swap them to compose two new images. As shown in Figure 2-C, the composite image after region swapping contains two views: one is the paste view (the cropped region), *i.e.*, the cat’s face, and the other is called the canvas view (the remained region), *i.e.*, the dog’s body. Specifically, we first sample a size of the paste view, *i.e.*, the height and width, and then determine the coordinates of the origin point from which the cropping starts. We make sure the size and location of the cropped region match the network’s downsampling ratio \mathcal{R} during region swapping so that the region’s feature can be directly extracted from the feature map by a simple operation of mask pooling. The height and width are determined by \mathcal{R} and a discrete uniform distribution $\mathcal{C} \sim U(\mathcal{C}_L, \mathcal{C}_U)$, where the ratio \mathcal{R} is typically 32 for ResNet [23] and $\mathcal{C}_L, \mathcal{C}_U$ are two predefined hyper-parameters shared for both spatial dimensions for simplicity. They are set to 3 and 5 in the paper unless specified. We sample twice from the distribution \mathcal{C} and get two observations c_h and c_w . Then, we calculate the width and height as $r_h = c_h \times \mathcal{R}, r_w = c_w \times \mathcal{R}$, respectively. Then we uniformly sample the origin point coordinates (r_x, r_y) from a valid range that guarantees there is enough remaining area to crop a patch of size $r_w \times r_h$. In this way, the candidate region is determined by (r_x, r_y, r_w, r_h) . It is noteworthy that within a mini-batch of the training images, we use the same coordinates (r_x, r_y, r_w, r_h) for efficient batch-wise implementation during training.

3.2 The region contrastive learning

The architecture. We take MoCov2 as an example here to describe the proposed RegionCL method in depth, denoted as RegionCL-M. The overall architecture is presented in Figure 3. As can be seen, RegionCL-M has exactly the same architecture with MoCov2 and only requires marginal modifications to the inputs and learning objectives, *i.e.*, a region-level branch in the middle.

RegionCL-M uses a Siamese network structure in pretraining. Given image instances x , RegionCL-M first creates two randomly augmented views, *i.e.*, the query view x^q and the key view x^k following the same augmentation strategy as in MoCov2. The online network processes the query view, and the other branch, *i.e.*, the momentum updated network, processes the key view. Unlike other methods that also utilize region-level contrastive learning [37,16,62], we follow the same cropping strategies as in MoCov2 and do not need the two views x^q, x^k to have a sufficiently large overlap, which keeps the diversity of the contrastive pair candidates. We construct the region-level contrastive pairs using the region-swapping strategy. Specifically, given two image instances from the query view x^q , we randomly crop a region with the same size in each image instance and swap them to compose two new images x^{pc} , where the cropped region after swapping and the remained region in the new images are the paste view x^p and canvas view x^c , respectively.

The region- and instance-level contrastive loss. In this way, we have a total of four different views, *i.e.*, the query view, the paste view, the canvas view, and the key view, denoted as x^q, x^p, x^c, x^k , respectively. RegionCL-M projects these views into the corresponding feature representations q, p, c, k , among which the features q, k are instance-level feature representations while the features p, c are region-level feature representations. Note that features of the paste view and canvas view are extracted from the feature maps of x^{pc} via mask pooling, where the mask is obtained according to the coordinates (r_x, r_y, r_w, r_h) as described in Section 3.1. The other views' features are from the global average pooling upon the corresponding feature maps. Then we can efficiently construct the contrastive pairs for these views according to the simple criteria, *i.e.*, each view is (1) positive with views augmented from the same original image and (2) negative with views augmented from other images. We follow the practice of MoCov2 in our implementation and ignore the positive pairs whose features are both generated by the online network to stabilize the training.

We use contrastive loss [20,21] as the learning objectives, which can be thought of as training an encoder for a dictionary lookup task at both instance and region levels. We first introduce the instance-level contrastive loss and then present the region-level one. Assume that we have a set of encoded samples $\{k_i | i = 1, 2, \dots, K\}$ as keys of a dictionary. For each query feature q , if there is a single key (k^+) that matches the query q , the contrastive loss aims to increase the similarity between q and k^+ meanwhile reducing the similarity between q and all other keys (considered as the negative counterparts for q). We use L2-normalized dot product to measure the similarity between the queries and keys,

and the contrastive loss, *i.e.*, the InfoNCE [34] loss, is therefore formulated as:

$$\mathcal{L}_{ins} = -\log \frac{\exp(q \cdot k^+/\tau)}{\sum_{i=0}^K \exp(q \cdot k_i/\tau)}, \quad (1)$$

where τ is a temperature hyper-parameter (set to 0.2 by default) [48,21]. Following MoCov2, the dictionary keys $\{k_i|i = 1, 2, \dots, K\}$ in RegionCL-M are maintained using a first-in-first-out queue with a predefined maximum number of samples (K), which is set to 65,536. The features from the key view k is treated as the positive sample and used to progressively update the memory queue, which serves as the negative samples. This form of contrastive loss is the exact one that appeared in MoCov2 [21], while it can have other forms for different SSL methods [34,11]. Apart from the instance-level pairs, the features of other views formulate the region-level pairs with the modified contrastive loss:

$$\begin{aligned} \mathcal{L}_{reg} = & -\frac{1}{2} \log \frac{\exp(p \cdot k_p^+/\tau)}{\sum_{i=0}^K \exp(p \cdot k_i/\tau) + \exp(p \cdot sg(c)/\tau)} \\ & -\frac{1}{2} \log \frac{\exp(c \cdot k_c^+/\tau)}{\sum_{i=0}^K \exp(c \cdot k_i/\tau) + \exp(c \cdot sg(p)/\tau)}, \end{aligned} \quad (2)$$

where the features p, c are obtained from the identical composite image x^{pc} (thus the term is divided by $\frac{1}{2}$ for normalization). $sg(\cdot)$ represents ‘stop gradient’, which helps stabilize the training. p and c are indeed hard negative pairs since they involve some context information from each other due to convolution and pooling operations, thereby helping the model to learn robust and discriminative feature representations. Thus the total contrastive loss is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{ins} + \mathcal{L}_{reg}. \quad (3)$$

Since the query, canvas, and paste views share the online network, we believe that the features of these views should be in the same feature space. Thus, RegionCL-M only needs a single queue to provide negative samples for features of all the three views, in contrast to the usage of multiple queues as in [50,56].

Extension to other SSL methods. As RegionCL defines a model-agnostic pretext task and requires minor modifications to the SSL methods, we also choose two other representative approaches, *i.e.*, DenseCL [47] and SimSiam [11], to further validate its effectiveness, denoted as RegionCL-D and RegionCL-S, respectively. Specifically, DenseCL [47] focuses on dense prediction tasks and has two learning objectives, *i.e.*, the instance-level contrastive loss as in MoCov2 and the pixel-level dense loss. Therefore, RegionCL-D includes the proposed region-level loss seamlessly as in RegionCL-M and keeps the pixel-level loss unchanged. For SimSiam [11], it only adopts instance-level positive pairs $\{p, k^+\}$ for training. Thus, we only enrich the positive pairs by collecting those abundant instance- and region-level positive pairs provided by RegionCL-S while retaining the other components. Similar strategy is adopted for vision transformer backbones [55,31,17] with MoBy [51]. Please refer to the supplementary for details.

4 Experiments

To thoroughly validate the improvements brought by introducing both regions into pretraining, we incorporate the RegionCL in representative state-of-the-art SSL methods, *i.e.*, MoCov2 [9], DenseCL [47], and SimSiam [11], and propose the RegionCL compatible models, *i.e.*, RegionCL-M, RegionCL-D, and RegionCL-S. The models are pretrained following the same settings as their own base methods, *i.e.*, we train RegionCL-M and RegionCL-D for 200 epochs, and RegionCL-S for 100 epochs, with an SGD [39] optimizer and corresponding augmentations, respectively. All the methods are based on ResNet-50 [23] backbone. Please refer to the supplementary material for more details. Vision transformer-based methods [51] are also explored to further evaluate RegionCL’s effectiveness.

4.1 Image classification on ImageNet

Settings. To evaluate the effectiveness of regional contrastive learning for image classification, we benchmark the RegionCL on ImageNet [15], which contains 1.28M images in the training set and 50K images in the validation set from 1,000 classes, respectively. The pretrained models of other SSL methods are either obtained from their authors or reproduced using their official codes. The performance of Top-1 and Top-5 accuracy on a single crop is reported. We have two experimental settings regards evaluation: linear classification and few-shot finetuning. The former setting follows the default setting of MoCov2 [21,9] and SimSiam [11] with SGD [39] and LARS [58] optimizer. The latter one using randomly sampled data per class from the training set.

Results with linear classification. We report the linear classification results of different methods in Table 1 and 2. ‘Real’ indicates that the labels used for evaluation are provided by [2]. From the table, we can see that RegionCL improves the aforementioned SSL baseline methods significantly by a large margin: +1.9% for RegionCL-M, +4.8% for RegionCL-D, and +3.2% for RegionCL-S. This proves RegionCL helps various SSL methods learn better feature representations owing to the abundant contrastive supervisory signals with contrastive pairs from both regional and global perspective. Comparing with methods that also exploring mixing two images for contrastive learning but from the global perspective only, *i.e.*, UnMix [38] and HEXA [26], RegionCL-M obtains better performance no matter using 200/800 epochs for training, demonstrating the benefits of leveraging regional contrastive pairs. Besides, RegionCL-S reaches 71.3% Top-1 accuracy using only 100 epochs, while the vanilla SimSiam requires a significantly longer training schedule of 800 epochs, proving the effectiveness of RegionCL in accelerating the model convergence and improving the performance. It is noteworthy that DenseCL focuses on dense prediction tasks and does not perform that well on classification. In contrast, RegionCL brings a large improvement on DenseCL for image classification, indicating that introducing remained regions to promote pretraining is not only compatible with classification-favored SSL methods but also generalizes well on dense prediction-favored approaches.

Results with linear few-shot finetuning. Table 3 presents the results of different methods at the linear few-shot finetuning setting. Thanks to the abundant contrastive pairs brought by RegionCL, the models pretrained by RegionCL have learned better feature representations from a complete perspective and can generalize well on classification tasks, thus delivering much more significant improvements over their baselines when only a limited number of data are available for finetuning, *i.e.*, a gain of +2.5%, +8.9%, +9.5% for 1% data and 1.8%, 6.4%, 8.1% for 10% data achieved by RegionCL-M, RegionCL-D, RegionCL-S.

	Epochs	ImageNet		Real
		Top-1	Top-5	Top-1
MoCo [21]	200	60.6	-	69.1
SimCLR [7]	1000	69.3	89.0	77.6
PIRL [32]	800	64.3	-	71.7
CPC-v2 [24]	200	63.8	85.3	-
InstLoc [56]	200	61.7	-	-
MaskCo [62]	200	65.1	-	-
ISD [41]	200	69.8	-	-
PCLv1 [27]	200	61.5	-	-
PCLv2 [27]	200	67.6	-	-
DUPR [16]	200	63.8	85.6	-
DetCo [50]	200	68.6	88.5	-
UnMix [38]	200	68.6	-	-
HEXA [26]	200	68.9	-	-
MoCov3 [12]	100	68.9	-	-
SimSiam [11]	800	71.3	-	-
MoCov2 [9]	200	67.5	88.2	77.8
RegionCL-M	200	69.4	89.6	78.7
DenseCL [47]	200	63.6	85.5	72.3
RegionCL-D	200	68.5	89.0	78.4
SimSiam [11]	100	68.1	88.2	77.8
RegionCL-S	100	71.3	90.4	80.8

Table 1: Linear classification results comparison on ImageNet [15].

	Epochs	ImageNet	
		Top-1	Top-5
MoCov2 [9]	800	71.1	90.2
UnMix [38]	800	71.8	-
HEXA [26]	800	71.7	-
RegionCL-M	800	73.1	91.5
MoCov2* [11]	800	72.3	-
RegionCL-M*	800	73.9	92.0
MoCov3 [12]	1000	74.6	-
RegionCL-M3	1000	75.4	92.6

Table 2: Linear classification results with more pretraining epochs. * means using symmetric loss.

	1% Data		10% Data	
	Top-1	Top-5	Top-1	Top-5
MoCov2 [9]	43.6	70.9	58.8	82.4
RegionCL-M	46.1	72.9	60.4	83.5
DenseCL [47]	38.9	66.2	54.0	79.3
RegionCL-D	47.8	74.0	60.4	83.1
SimSiam [11]	32.8	61.5	51.8	77.7
RegionCL-S	42.3	70.6	59.9	83.8

Table 3: Linear classification results on ImageNet [15] using 1% and 10% data.

4.2 Detection and segmentation on MS COCO

Settings. We show the detection performance of the models pretrained with the RegionCL pretext task. The experiments are conducted on the MS COCO dataset [10], which contains about 118K images with bounding boxes and instance segmentation annotations and covers 80 object categories in total. We choose two representative detectors: the two-stage detector Mask-RCNN [22] and the one-stage detector RetinaNet [29], following the same settings as in [50,62].

	Mask-RCNN-C4-1x						Mask-RCNN-FPN-1x					
	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	6.9	7.6	14.8	7.2	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 [48]	37.7	57.0	40.9	33.0	54.1	35.2	37.4	57.6	40.6	34.1	54.6	36.4
PIRL [32]	37.4	56.5	40.2	32.7	53.4	34.7	37.5	57.6	41.0	34.0	54.6	36.2
SwAV [3]	32.9	54.3	34.5	29.5	50.4	30.4	38.5	60.4	41.4	35.4	57.0	37.7
MoCo [21]	38.5	58.3	41.6	33.6	54.8	35.6	38.5	58.9	42.0	35.1	55.9	37.7
DetCo [50]	39.4	59.2	42.3	34.4	55.7	36.6	39.5	60.3	43.1	35.9	56.9	38.6
DetCo-AA [50]	39.8	59.7	43.0	34.7	56.3	36.7	40.1	61.0	43.9	36.4	58.0	38.9
MoCo v2 [9]	38.9	58.4	42.0	34.2	55.2	36.5	38.9	59.4	42.4	35.5	56.5	38.1
RegionCL-M	39.8	59.8	43.0	34.8	56.4	36.9	40.1	60.7	43.9	36.3	57.7	39.0
DenseCL [47]	39.3	59.1	42.2	34.5	55.6	36.8	39.1	59.4	42.5	35.5	56.4	38.0
RegionCL-D	40.3	60.3	43.9	35.2	57.0	37.3	40.4	61.3	44.2	36.7	58.2	39.4
SimSiam [11]	37.9	57.5	40.9	33.2	54.2	35.2	37.3	57.2	40.5	33.9	54.2	36.1
RegionCL-S	38.7	58.2	41.3	33.7	55.0	35.6	38.8	58.8	42.4	35.2	56.0	37.6

Table 4: Object detection results on the MS COCO [10] dataset with Mask-RCNN [22] C4 and FPN (1x).

Results of Mask-RCNN on MS COCO. Table 4 and 5 summarize the Mask-RCNN results on 1x and 2x schedules respectively, where the RegionCL variants are highlighted in bold. We can see that RegionCL has significantly improved all approaches with ResNet50-C4 and ResNet50-FPN backbones, confirming the benefits of region-level contrastive learning on various SSL methods. According to the tables, incorporating RegionCL into MoCov2 (RegionCL-M) can further improve the performance over the MoCov2 baseline with both backbones. It is also noticeable that RegionCL-M has already surpassed the previous representative SSL methods designed for dense prediction, *e.g.*, DetCo [50] and DenseCL [47]. More importantly, when incorporating RegionCL into DenseCL, RegionCL-D achieves the best scores for all metrics in both the 1x and 2x settings. It suggests that RegionCL can still help the dense prediction-favored methods to learn more discriminative features from a complete perspective.

Results of RetinaNet on MS COCO. The results of RetinaNet on MS COCO using different SSL methods are presented in Table 6. From the table, we can see that the improvement brought by RegionCL still holds in all metrics and at all the training settings. Similarly, RegionCL-D achieves the best results at 38.8 AP and 40.6 AP for the two training schedules respectively, significantly surpassing the supervised baseline by 1.4 AP and 1.7 AP. It is also noted that the improvement in the more stringent metric AP₇₅^{bb} is more significant than that in the AP^{bb} metric, demonstrating that leveraging both cropped and remained regions for contrastive learning contributes to learning better feature representations for object detection and thus improving the detection accuracy. These results show that the simple strategy RegionCL can help existing SSL methods achieve a better trade-off between the classification and detection tasks (see Figure 1), further validating the benefits of using both regions for pretraining.

	Mask-RCNN-C4-2x						Mask-RCNN-FPN-2x					
	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	35.6	54.6	38.2	31.4	51.5	33.5	36.7	56.7	40.0	33.7	53.8	35.9
Supervised	40.0	59.9	43.1	34.7	56.5	36.9	40.6	61.3	44.4	36.8	58.1	39.5
MoCo [21]	40.7	60.5	44.1	35.4	57.3	37.6	40.8	61.6	44.7	36.9	58.4	39.7
MaskCo [62]	40.8	60.5	44.2	35.5	57.1	38.0	-	-	-	-	-	-
UnMix [38]	-	-	-	-	-	-	41.2	60.9	44.7	-	-	-
DetCo [50]	41.4	61.2	44.7	35.8	57.8	38.3	41.5	62.1	45.6	37.6	59.2	40.5
DetCo-AA [50]	41.3	61.2	45.0	35.8	57.9	38.2	41.5	62.5	45.6	37.7	59.5	40.5
MoCo v2 [9]	41.0	60.6	44.5	35.6	57.2	38.0	40.9	61.5	44.7	37.0	58.7	39.8
RegionCL-M	41.5	61.4	44.9	35.9	57.7	38.6	41.6	62.5	45.6	37.7	59.3	40.4
DenseCL [47]	39.7	59.1	43.0	34.5	55.9	37.3	41.4	62.1	45.1	37.5	58.8	40.3
RegionCL-D	41.8	61.6	45.4	36.4	58.5	39.2	42.1	62.9	45.9	38.0	60.0	40.7
SimSiam [11]	38.8	58.0	41.9	34.0	55.1	36.2	40.1	60.6	43.8	36.4	57.7	39.1
RegionCL-S	40.7	60.4	44.4	35.4	57.0	37.7	41.0	61.6	44.7	37.1	58.6	39.8

Table 5: Object detection results on the MS COCO [10] dataset with Mask-RCNN [22] C4 and FPN (2x).

4.3 Segmentation on Cityscapes

Settings. Further, we evaluate the models’ transfer performance for both instance and semantic segmentation on the Cityscapes [14] dataset, which contains over 5K well-annotated images of street scenes from 50 different cities. We follow the same setting as in MoCov2 [21] for instance segmentation, with Mask-RCNN and trained for 24K iterations. UPerNet [49] in mmseg [13] is employed for semantic segmentation evaluation.

Results on Cityscapes. Table 7 presents the performance of different SSL methods and their variants with RegionCL. The second and third columns show the performance for instance and semantic segmentation, respectively. According to the table, RegionCL consistently improves the three representative SSL methods by large margins. For example, RegionCL-M reaches the best on instance segmentation at 34.9 AP and 62.5 AP₇₅, while RegionCL-D outperforms the others on semantic segmentation at 78.7 mIoU and 79.5 mIoU with different training schedules, confirming the generalization and the effectiveness of region-level contrastive learning with remained regions on segmentation tasks.

4.4 Generalization on vision transformer

To further evaluate the benefits of using both regions for pretraining, we apply RegionCL on vision transformer-based methods and train them following the design in MoBy [51]. As shown in Table 8, RegionCL improves MoBy by over 3.0 accuracy on both ImageNet and ImageNet Real with 100 epochs pretraining and over 1.5 accuracy with 300 epochs pretraining. It can be concluded that RegionCL can successfully improve the vision transformer’s performance, validating its effectiveness for advanced neural architectures like vision transformers.

	RetinaNet-1x			RetinaNet-2x		
	AP ^{bb}	AP ₅₀ ^{bb}	AP ₇₅ ^{bb}	AP ^{bb}	AP ₅₀ ^{bb}	AP ₇₅ ^{bb}
Rand Init	24.5	39.0	25.7	32.2	49.4	34.2
Supervised	37.4	56.5	39.7	38.9	58.5	41.5
InsDis [48]	35.5	54.1	38.2	38.0	57.4	40.5
PIRL [32]	35.7	54.2	38.4	38.5	57.6	41.2
SwAV [3]	35.2	54.9	37.5	38.6	58.8	41.1
MoCo [21]	36.3	55.0	39.0	38.7	57.9	41.5
DUPR [16]	38.0	57.2	40.7	40.0	59.6	43.0
DetCo [50]	38.0	57.4	40.7	39.8	59.5	42.4
DetCo-AA [50]	38.4	57.8	41.2	39.7	59.3	42.6
MoCov2 [9]	37.2	56.2	39.6	39.3	58.9	42.1
RegionCL-M	38.4	58.1	41.2	40.1	59.9	43.2
DenseCL [47]	37.7	56.4	40.2	39.8	59.2	42.8
RegionCL-D	38.8	58.6	41.6	40.6	60.4	43.6
SimSiam [11]	35.5	53.7	38.1	38.1	57.4	40.8
RegionCL-S	36.8	55.9	39.5	39.1	58.5	41.8

	Inst. Seg		Sem. Seg	
	AP	AP ₇₅	40K	80K
Supervised	32.9	59.6	77.1	78.2
MoCov2 [9]	33.9	60.8	77.8	78.6
RegionCL-M	34.9	62.5	78.1	79.0
DenseCL [47]	34.5	61.9	78.3	79.1
RegionCL-D	34.8	62.3	78.3	79.5
SimSiam [11]	33.6	61.0	76.2	78.1
RegionCL-S	34.9	61.6	77.8	78.7

Table 6: Detection results on MS COCO [10] with RetinaNet [29].

Table 7: Instance (Inst.) and semantic (Sem.) segmentation results (mIoU) on Cityscapes [14].

4.5 Ablation Study

We conduct the ablation studies with RegionCL-M. All models are trained for 100 epochs due to the limitation of computation resources and follow the practice of previous works [50,62]. We adopt a k -NN classifier to evaluate their classification accuracy on ImageNet [15] and train these models for 12K iterations on MS COCO [10] to evaluate their dense prediction performance.

	BackboneEpochs		ImageNet		Real
			Top-1	Top-5	Top-1
MoBy [51]	Swin-T	100	70.9	89.7	77.5
RegionCL+MoBy	Swin-T	100	73.9	91.8	81.2
MoBy [51]	Swin-T	300	75.3	92.2	82.4
RegionCL+MoBy	Swin-T	300	77.0	93.1	83.9

Table 8: Linear classification results comparison on ImageNet with vision transformer-based method MoBy [51].

The size of the paste view. We investigate the influence of the size of the paste view by varying the lower and upper bounds \mathcal{C}_L and \mathcal{C}_U . Note that the image size is set as 224 during the training and the downsampling ratio for the backbone network ResNet-50 [23] is 32, thus $\mathcal{C}_L, \mathcal{C}_U$ are valid in the range [1, 7]. The optimal hyper-parameters are determined through two steps. **(1)** We first fix the upper bound \mathcal{C}_U to 5 and search different configurations for the lower bound \mathcal{C}_L . As shown in Table 9, the performance on both classification and detection peaks with $\mathcal{C}_L = 3$. **(2)** Then we fix \mathcal{C}_L to 3 and search for \mathcal{C}_U .

Configuration		ImageNet		MS COCO	
\mathcal{C}_L	\mathcal{C}_U	20-NN	100-NN	AP ^{bb}	AP ^{mk}
-	-	49.3	47.3	26.3	24.0
1	5	51.8	49.8	27.3	24.9
2	5	53.0	51.2	27.9	25.5
3	5	54.7	52.7	28.0	25.6
4	5	53.9	51.8	27.9	25.5
3	4	55.1	52.8	27.8	25.5
3	6	54.3	52.5	27.8	25.5

Table 9: The influence of paste view’s size. \mathcal{C}_L and \mathcal{C}_U denote the lower and upper bound of the size for the paste view generation. ‘-’ denotes no paste view is used during the training, *i.e.*, downgrading to the original MoCov2.

Configuration			ImageNet		MS COCO	
Paste	Canvas	Neg	20-NN	100-NN	AP ^{bb}	AP ^{mk}
×	×	×	49.3	47.3	26.3	24.0
×	✓	×	51.8	50.0	26.9	24.7
✓	×	×	50.0	50.2	27.0	24.7
✓	✓	×	54.6	52.4	27.8	25.5
✓	✓	✓	54.7	52.7	28.0	25.6

Table 10: The influence of paste and canvas views. ‘Paste’/‘Canvas’ denote using paste/canvas views as positive pairs. ‘Neg’ means using the canvas and paste counterpart views in the composited images as negative pairs.

It is interesting to see that decreasing \mathcal{C}_U from 5 to 4 slightly improves the performance on classification but degrades that on detection. This suggests that the optimal configurations of $\mathcal{C}_L, \mathcal{C}_U$ for classification and dense prediction tasks may be slightly different, and we select $\mathcal{C}_L = 3, \mathcal{C}_U = 5$ as default values to achieve a trade-off on both classification and dense prediction tasks.

The influence of using paste and canvas views. We further investigate the importance of using both paste and canvas views during pretraining. The results are concluded in Table 10, where ✓ under Paste or Canvas denotes whether to use the former or latter term in Eq. (2). The ‘Negative’ option means whether to treat the canvas and paste counterpart views from the same composite image x^{pc} as negative pairs, *i.e.*, $\exp(c \cdot sg(p)/\tau)$ or $\exp(p \cdot sg(c)/\tau)$ in the denominator in Eq. (2). With all columns marked ×, the method becomes standard MoCov2. From the first three rows in the table, we can see that using either the paste or canvas views can bring performance gains, and the performance will be further boosted when considering both regions. For example, in the 4th row, the model gains more than 5% accuracy improvement over MoCov2 in both ImageNet 20- and 100-NN classification. We attribute it to that leveraging both regions during pretraining can help models learn better category features from a complete perspective. Comparing the 4th row with the last row, where intra-image negative pairs are utilized, the performance on both tasks is slightly improved. It demonstrates that using negative pairs within images can help models learn more discriminative features between different regions, again validating the importance of introducing region-level contrastive pairs in self-supervised learning.

4.6 Analysis of RegionCL

To better analyze the performance gains brought by RegionCL, we monitor the KNN accuracy and gradients of the target regions during training. We use

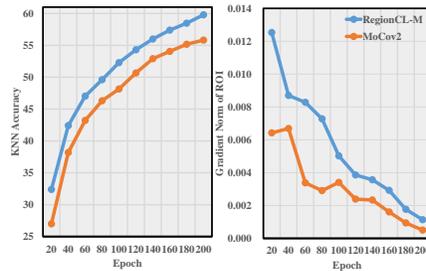


Fig. 4: The KNN accuracy and average gradient magnitude of MoCov2 [9] and RegionCL-M with different training epochs.

YOLOX [18] to detect the objects in the images and record the average magnitude of gradients back-propagated from the loss of MoCov2 and RegionCL-M in the object bounding boxes on the training set. As shown in Figure 4, the KNN accuracy of RegionCL-M consistently outperforms MoCov2 while the gradients inside the object regions of RegionCL-M are larger than those of MoCov2, implying that using both regions for training can help models pay more attention on the targets and learn better object representation from a complete perspective.

5 Limitation and Discussion

We make the first attempt to demonstrate the importance of considering both cropped and remained regions in SSL via a simple yet effective RegionCL pretext task. The simple task makes minor modifications to representative SSL methods to show the benefits brought by leveraging regional contrastive pairs. Although RegionCL effectively improve these methods, there is still much to be explored in utilizing the remained regions for more tasks [53,54,28,1], *e.g.*, introducing additional bounding box selection and alignment modules or adopting multi-level supervision for learning. Besides, RegionCL temporally costs more computations per iteration (about 50% for RegionCL-M). Although it demonstrates better results with fewer computation resources as discussed in the supplementary, it will be beneficial to reduce the training costs, which we leave as our future work.

6 Conclusion

This paper demonstrates the importance of using both cropped and remained regions after cropping for self-supervised learning. A simple yet effective pretext task RegionCL is proposed to validate the models can learn better category feature representation from a complete perspective. Experimental results on image classification, object detection, and instance and semantic segmentation benchmarks demonstrate the effectiveness of leveraging remained regions in pretraining and its compatibility to representative self-supervised learning methods. We hope that this study will provide valuable insights into the subsequent studies of self-supervised learning in exploring region-based contrast methodology.

Acknowledgement Mr. Yufei Xu, Mr. Qiming Zhang, and Dr. Jing Zhang are supported by ARC FL-170100117.

References

1. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: International Conference on Learning Representations (2021)
2. Beyer, L., Hénaff, O.J., Kolesnikov, A., Zhai, X., Oord, A.v.d.: Are we done with imagenet? arXiv preprint arXiv:2006.07159 (2020)
3. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint arXiv:2006.09882 (2020)
4. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. arXiv preprint arXiv:2104.14294 (2021)
5. Chen, K., Hong, L., Xu, H., Li, Z., Yeung, D.Y.: Multisiam: Self-supervised multi-instance siamese representation learning for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7546–7554 (2021)
6. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* **40**(4), 834–848 (2017)
7. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. vol. 1, pp. 1597–1607. PMLR (2020)
8. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. *Advances in Neural Information Processing Systems* **33**, 22243–22255 (2020)
9. Chen, X., Fan, H., Girshick, R.B., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
10. Chen, X., Fang, H., Lin, T.Y., Vedantam, R., Gupta, S., Dollar, P., Zitnick, C.L.: Microsoft coco captions: Data collection and evaluation server. arXiv preprint arXiv:1504.00325 (2015)
11. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 15750–15758 (2021)
12. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9640–9649 (2021)
13. Contributors, M.: MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/msegmentation> (2020)
14. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
15. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255. Ieee (2009)
16. Ding, J., Xie, E., Xu, H., Jiang, C., Li, Z., Luo, P., Xia, G.S.: Unsupervised pretraining for object detection by patch reidentification. arXiv preprint arXiv:2103.04814 (2021)

17. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2020)
18. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
19. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
20. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2, pp. 1735–1742. IEEE (2006)
21. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9729–9738 (2020)
22. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2961–2969 (2017)
23. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016)
24. Henaff, O.: Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. pp. 4182–4192. PMLR (2020)
25. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 577–593. Springer (2016)
26. Li, C., Li, X., Zhang, L., Peng, B., Zhou, M., Gao, J.: Self-supervised pre-training with hard examples improves visual representations. arXiv preprint arXiv:2012.13493 (2020)
27. Li, J., Zhou, P., Xiong, C., Hoi, S.: Prototypical contrastive learning of unsupervised representations. In: International Conference on Learning Representations (2020)
28. Li, Y., Mao, H., Girshick, R., He, K.: Exploring plain vision transformer backbones for object detection. arXiv preprint arXiv:2203.16527 (2022)
29. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017)
30. Liu, S., Li, Z., Sun, J.: Self-emd: Self-supervised object detection without imagenet. arXiv preprint arXiv:2011.13677 (2020)
31. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10012–10022 (2021)
32. Misra, I., Maaten, L.v.d.: Self-supervised learning of pretext-invariant representations. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6707–6717 (2020)
33. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 69–84. Springer (2016)
34. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

35. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2536–2544 (2016)
36. Pinheiro, P.O., Almahairi, A., Benmalek, R.Y., Golemo, F., Courville, A.: Unsupervised learning of dense visual representations. arXiv preprint arXiv:2011.05499 (2020)
37. Roh, B., Shin, W., Kim, I., Kim, S.: Spatially consistent representation learning. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1144–1153 (2021)
38. Shen, Z., Liu, Z., Liu, Z., Savvides, M., Darrell, T., Xing, E.: Un-mix: Rethinking image mixtures for unsupervised visual representation learning. arXiv preprint arXiv:2003.05438 (2020)
39. Sutskever, I., Martens, J., Dahl, G., Hinton, G.: On the importance of initialization and momentum in deep learning. In: International conference on machine learning. pp. 1139–1147. PMLR (2013)
40. Tao, C., Wang, H., Zhu, X., Dong, J., Song, S., Huang, G., Dai, J.: Exploring the equivalence of siamese self-supervised learning via a unified gradient framework. arXiv preprint arXiv:2112.05141 (2021)
41. Tejankar, A., Koohpayegani, S.A., Pillai, V., Favaro, P., Pirsiavash, H.: Isd: Self-supervised learning by iterative similarity distillation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9609–9618 (2021)
42. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 776–794. Springer (2020)
43. Tian, Y.: Deep contrastive learning is provably (almost) principal component analysis. arXiv preprint arXiv:2201.12680 (2022)
44. Wang, C., Xu, C., Tao, D.: Self-supervised pose adaptation for cross-domain image animation. *IEEE Transactions on Artificial Intelligence* **1**(1), 34–46 (2020)
45. Wang, F., Liu, H.: Understanding the behaviour of contrastive loss. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2495–2504 (2021)
46. Wang, T., Isola, P.: Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In: International Conference on Machine Learning. pp. 9929–9939. PMLR (2020)
47. Wang, X., Zhang, R., Shen, C., Kong, T., Li, L.: Dense contrastive learning for self-supervised visual pre-training. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3024–3033 (2021)
48. Wu, Z., Xiong, Y., Yu, S.X., Lin, D.: Unsupervised feature learning via non-parametric instance discrimination. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3733–3742 (2018)
49. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 418–434 (2018)
50. Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: Detco: Unsupervised contrastive learning for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 8392–8401 (2021)
51. Xie, Z., Lin, Y., Yao, Z., Zhang, Z., Dai, Q., Cao, Y., Hu, H.: Self-supervised learning with swin transformers. arXiv preprint arXiv:2105.04553 (2021)

52. Xie, Z., Lin, Y., Zhang, Z., Cao, Y., Lin, S., Hu, H.: Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16684–16693 (2021)
53. Xu, Y., Zhang, J., Maybank, S.J., Tao, D.: Dut: Learning video stabilization by simply watching unstable videos. *IEEE Transactions on Image Processing* (2022)
54. Xu, Y., Zhang, J., Zhang, Q., Tao, D.: Vitpose: Simple vision transformer baselines for human pose estimation. arXiv preprint arXiv:2204.12484 (2022)
55. Xu, Y., Zhang, Q., Zhang, J., Tao, D.: Vitae: Vision transformer advanced by exploring intrinsic inductive bias. *Advances in Neural Information Processing Systems* **34**, 28522–28535 (2021)
56. Yang, C., Wu, Z., Zhou, B., Lin, S.: Instance localization for self-supervised detection pretraining. In: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3987–3996 (2021)
57. Yang, J., Zhang, K., Cui, Z., Su, J., Luo, J., Wei, X.: Inscon: Instance consistency feature representation via self-supervised learning. arXiv preprint arXiv:2203.07688 (2022)
58. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
59. Zhang, J., Tao, D.: Empowering things with intelligence: a survey of the progress, challenges, and opportunities in artificial intelligence of things. *IEEE Internet of Things Journal* **8**(10), 7789–7817 (2020)
60. Zhang, Q., Xu, Y., Zhang, J., Tao, D.: Vitaev2: Vision transformer advanced by exploring inductive bias for image recognition and beyond. arXiv preprint arXiv:2202.10108 (2022)
61. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 649–666. Springer (2016)
62. Zhao, Y., Wang, G., Luo, C., Zeng, W., Zha, Z.J.: Self-supervised visual representations learning by contrastive mask prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 10160–10169 (2021)