Revisiting Pixel-Level Contrastive Pre-Training on Scene Images

Zongshang Pang[1] Yuta Nakashima[1] Mayu Otani[2] Hajime Nagahara[1]

[1] Osaka University [2] CyberAgent, Inc.

Abstract

Contrastive image representation learning through instance discrimination has shown impressive transfer performance. Recent strategies have focused on pushing the limit of their transfer performance for dense prediction tasks, particularly when conducting pre-training on scene images with complex structures. Initial approaches employ pixel-level contrastive pre-training to optimize dense spatial features, while subsequent methods utilize region-mining algorithms to capture holistic regional semantics and address the issue of semantically inconsistent scene image crops. In this paper, we revisit pixel-level contrastive pre-training on scene images. Contrary to the assumption that pixel-level learning falls short in achieving these objectives, we demonstrate its under-explored potentials: (1) it can effectively learn holistic regional semantics more simply compared to region-level methods, and (2) it intrinsically provides tools to mitigate the impact of semantically inconsistent views involved with scene-level training images. We propose PixCon, a pixel-level contrastive learning framework, and explore two variants with different positive matching strategies to investigate the potential of pixel-level learning. Additionally, when PixCon incorporates a novel semantic reweighting approach tailored for scene image pre-training, it outperforms or matches the performance of previous region-level methods in object detection and semantic segmentation tasks across multiple benchmarks.[1]

1. Introduction

Contrastive image representation learning [1–3, 5, 15, 16, 34, 38] has remarkably advanced transfer learning for vision tasks. Such contrastive learning methods conduct *instance discrimination* [38] by pulling closer the features of two augmented views of the same image (positive pairs) while repelling them from those of different images (negative samples). As such methods work with global average-pooled feature vectors of random image crops, they are often referred to as *image-level learning* methods [35, 36, 40, 45]. Image-level methods learn representations promising for image classification but less satisfactory for dense prediction tasks such as object detection [30] and semantic segmentation [26]. Therefore, there have been various efforts to leverage more spatial information for better dense prediction performance [19, 21, 23, 31, 35, 36, 40, 41, 45], usually referred to as *dense learning* methods due to their focus on dense spatial features.

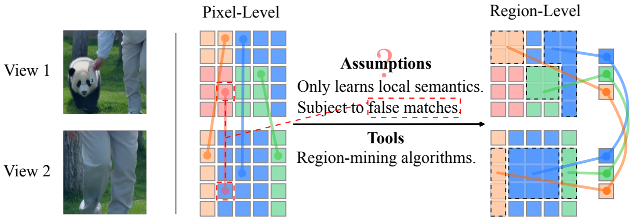
Pixel-Level
Region-Level
View 1
View 2
Assumptions
?
Only learns local semantics.
Subject to false matches.
Tools
Region-mining algorithms.

Figure 1. An illustration of the problems this paper studies. Girds' colors roughly indicate pixels' associated semantic classes based on the two input views for illustration purposes. The cross-view pixels connected by solid lines with round markers indicate positive matches. The matching process for pixel-level learning imitates the similarity-based matching from [35]. Region-level methods are motivated by the shown assumptions about pixel-level learning and rely on region-mining algorithms as tools to perform learning based on regional features. In this paper, we question these assumptions about pixel-level learning and revisit it to further exploit its potential.

Besides, image-level learning methods primarily benefit from meticulously curated datasets, *e.g.*, ImageNet [10]. However, they are insufficient for leveraging complex scene-centric images [23, 33, 36, 40] such as those from the MS COCO dataset [24]. Overall, two issues limit their performance on scene images. Firstly, it is challenging for image-level learning to exploit the rich semantic information in multi-object scene images. Secondly, the random views from scene images can be semantically inconsistent as they often encompass different semantic contents, promoting the correlation between representations of different objects or objects and backgrounds. An example of semantically inconsistent views is provided in Figure 1, where the panda only appears in the first view. Due to their wise utilization of spatial information, dense learning methods have

[1] Code available at https://github.com/pangzss/PixCon

proven effective in dealing with these issues.

Dense learning methods are usually categorized as *pixel-level* [29, 35] or *region-level* [19, 21, 23, 31, 36, 40, 41, 45], as the former learns with individual spatial feature vectors, whereas the latter works with selective aggregations of them. Pixel-level methods rely on conceptually simple ways to retrieve pixel-level positive pairs but are usually assumed by some work [23, 36, 40, 45] to be not capable of capturing holistic regional semantics. Region-level methods are usually equipped with various region-mining algorithms such as unsupervised object detection [5, 15, 27, 32] or segmentation [2, 13] to obtain regions of interest that are used to pool spatial features. A conceptual illustration of their positive matching processes is provided in Figure 1. Either pixel-level or region-level learning helps better exploit complex scene images compared to image-level learning. In contrast to pixel-level methods, with the help of region-mining algorithms, region-level methods usually come with natural mitigation to the problem of semantically inconsistent views cropped from scene images.

In this paper, we take a step back to revisit pixel-level learning with scene images. We aim to show that: (1) the potential of previous pixel-level baselines is under-exploited; (2) pixel-level learning can also encourage the emergence of regional semantics; and (3) pixel-level learning also naturally comes with tools to mitigate the problem of semantically inconsistent scene crops. Specifically, we make the following contributions:

- We first propose a pixel-level contrastive learning framework, *PixCon-Sim*, which improves upon a previous pixel-level learning method, DenseCL [35], by aligning its training pipeline with that of state-of-the-art (SOTA) region-level methods [20, 21, 23, 36, 40, 45] and achieves competitive transfer performance compared to SOTA region-level methods.
- We then investigate the potential of pixel-level learning by comparing two positive matching schemes, namely similarity-based [35] and coordinate-based [29, 36], with their corresponding models named PixCon-Sim and *PixCon-Coord*. We show that the similarity-based scheme intrinsically encourages the learning of regional semantics that region-level methods focus on.
- Finally, we propose *PixCon-SR* with a *Semantic Reweighting* strategy for tackling the problem of semantically inconsistent scene crops by simultaneously utilizing pixel spatial coordinates and pixel feature similarities. PixCon-SR achieves better or competitive transfer performance compared to current SOTA methods on dense prediction tasks, including PASCAL VOC object detection [12], COCO object detection and instance segmentation [24], PASCAL VOC semantic segmentation [12] and Cityscapes semantic segmentation [9].

2. Related Work

Image-level Self-Supervised Learning. Initial efforts in self-supervised image representation learning train models to predict colors [44], relative positions [11], or rotations of pixels [14]. Recent work exploits the task of instance discrimination [38] based on contrastive learning, where features of two augmented views of the same image are optimized to be closer than those of different images in the dataset [3, 5, 16]. The commonly utilized InfoNCE loss [28] can be decomposed into alignment and uniformity loss terms [34]. BYOL [15] removes the uniformity term and only aligns positive pairs by applying specific strategies to avoid the trivial solution that the network outputs the same values for all inputs.

The methods above work on the average-pooled versions of spatial features and are thus referred to as *image-level learning* methods. The resulting representations are promising for image classification but less impressive in terms of dense prediction tasks, as the spatial features are not directly optimized to be discriminative enough to provide a decent starting point for such downstream tasks.

Dense Self-Supervised Learning. Current efforts in crafting better representations for dense prediction tasks usually focus on exploiting more spatial information. *Pixel-level* methods usually find cross-view pixel-level positive matches. Pinheiro *et al.* [29] propose to match cross-view pixels with the spatial coordinate. DenseCL [35] instead finds matches based on cross-view feature similarities. Later work usually considers such pixel-level learning as limited and proposes to work on *region-level* features to tap into more holistic spatial information. Such methods usually require region-mining algorithms to produce region-level features, based on which contrastive learning [5] or self-distillation [15] is performed. Some methods [23, 31, 40] apply unsupervised object region proposal algorithms [5, 15, 27, 32] to directly generate object-centric crops. Some methods utilize segmentation masks to perform selective pooling of spatial feature maps. Specifically, Henaff *et al.* [20] applies the FH algorithm [13] to obtain segmentation masks. Feature-level KMeans [25] is used by [21] to generate segmentation masks on the fly, resulting in an alternating training scheme. Learnable prototypes are applied by [36, 45] to produce masks. PixPro [41] connects pixel-level and region-level learning by first considering spatially close cross-view pixel features as matches and transforming one of the pixel features into a region-level feature using its self-attention map. Hence, we still consider it a region-level method.

Despite the common assumption made by some region-level methods [23, 36, 40, 41, 45] that pixel-level learning only focuses on local rather than regional semantics, we will show that pixel-level learning can also promote learning regional semantics but without using region-mining algorithms, and can eventually obtain decent transfer performance as well.

Learning with Scene-Centric Images. Compared to image-level learning methods, dense representation learning has proven to be more effective when it comes to pre-training with scene-centric images [20,21,23,33,35,36,39,45] such as those from MS COCO [24]. On the one hand, such methods can leverage richer spatial information compared to image-level methods. On the other hand, region-level methods, with the help of their region-mining algorithms, can often readily handle the situation where two random crops of a scene image are semantically inconsistent, but at the cost of complicated pre-processing [20,23,33,39], nontrivial computational burden during training [21], or less transferrable features [36,45] compared to pixel-level methods. However, as we will show, pixel-level learning also provides natural tools to mitigate the negative influence of semantically inconsistent views and does not resort to any region-mining algorithms.

3. Preliminaries

This section reviews two popular image-level learning pipelines, MoCo-v2 [16] and BYOL [15], where the latter is the default pipeline of most of the region-level methods. We also introduce a variant of MoCo-v2 with a similar architecture to that of BYOL, coined MoCo-v2+ by [22].

Common in MoCo-v2 and BYOL, each input image is augmented into two different views $\mathbf{x}_1 \sim \mathcal{T}_1(\mathbf{x})$ and $\mathbf{x}_2 \sim \mathcal{T}_2(\mathbf{x})$, which are then fed into the *online* encoder f_θ and the *target* encoder f_ξ, where θ represents the learnable parameters and ξ is the exponential moving average of θ. The encoders are backbone networks, *e.g.*, ResNet [18], appended with two-layer multilayer perceptions (MLPs). The MLPs are usually called *projection heads*. The f_θ in BYOL has an additional two-layer MLP called *predictor*, resulting in an asymmetric structure between the two encoders. Moreover, MoCo-v2 feeds each view into either the online or the target encoder to compute a loss $\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2)$, while BYOL sends each view to both encoders and symmetrizes the loss computation w.r.t. the two views, *i.e.*, $\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}_{\text{img}}(\mathbf{x}_2, \mathbf{x}_1)$. Huang *et al.* [22] add to MoCo-v2 the asymmetric encoder structure, where the online encoder contains a predictor, and the symmetrized loss with

$$\mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) = -\log \frac{\exp(\mathbf{q}\cdot\mathbf{k}^+/\tau)}{\sum\limits_{\mathbf{k}\in\{\mathbf{k}^+\}\cup\mathcal{K}} \exp(\mathbf{q}\cdot\mathbf{k}/\tau)}, \tag{1}$$

where $\mathbf{q} = f_\theta(\mathbf{x}_1)/\|f_\theta(\mathbf{x}_1)\|_2$ is the query feature and $\mathbf{k}^+ = f_\xi(\mathbf{x}_2)/\|f_\xi(\mathbf{x}_2)\|_2$ is the positive key feature. $\mathcal{K}$ is the set of f_ξ's outputs from other images which are $\mathbf{q}$'s negative key features stored in a fixed-length queue [16], and τ is the temperature coefficient. $\mathcal{L}_{\text{img}}(\mathbf{x}_2, \mathbf{x}_1)$ is computed by obtaining the query from $\mathbf{x}_2$ and the positive key from $\mathbf{x}_1$. The loss in Eq (1) is usually referred to as the InfoNCE loss [28]. In contrast, BYOL only aligns the positive features by maximizing their cosine similarities [15].

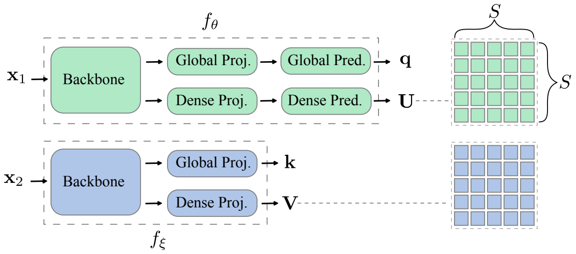

Figure 2. Both the online and the target encoders output two sets of outputs: global image-level outputs $(\mathbf{q}, \mathbf{k})$ and dense outputs $(\mathbf{U}, \mathbf{V})$. The dense outputs are of size $S \times S \times C$ before flattening the spatial dimensions. We leave out the visualization of global features and dense features' last dimension (C).

Besides, BYOL also applies a momentum ascending strategy for updating ξ and the synchronized batch normalization [43] as opposed to the shuffling batch normalization [16] in MoCo-v2. When MoCo-v2 is equipped with these BYOL-style designs, it is called MoCo-v2+ in [22], demonstrating similar linear probing and transfer learning performance to that of BYOL but better than that of MoCO-v2. Moreover, SimSiamese [6] is a simplified version of BYOL, achieving better performance under similar training settings. For simplicity, we refer to BYOL, MoCo-v2+, and SimSiamese all as BYOL pipelines if not stated otherwise.

4. Proposed Method

Based on MoCo-v2+, we add another asymmetric prediction structure to the backbone that outputs dense spatial feature maps, or *pixel-level*[2] features [36,40,45]. The online encoder f_θ now gives two sets of feature vectors $\mathbf{q} \in \mathbb{R}^C$ and $\mathbf{U} \in \mathbb{R}^{S^2 \times C}$ (after flattening the first two dimensions), where C is the feature dimensionality, and S is the length and width of the dense feature maps, which are set equal for simplicity. Similarly, the target encoder f_ξ gives $\mathbf{k} \in \mathbb{R}^C$ and $\mathbf{V} \in \mathbb{R}^{S^2 \times C}$. Figure 2 provides a schematic illustration of the forward process. Based on this forward pipeline, we propose different variants of a pixel-level contrastive learning framework, namely *PixCon*, with the loss function being

$$\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) = \mathcal{L}_{\text{img}}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2), \tag{2}$$

where $\mathcal{L}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ is the pixel-level contrastive loss to be defined. The final loss is symmetrized w.r.t. the two views,

[2] *Pixels* in this context refer to spatial components of dense feature maps as opposed to those of the input RGB images.

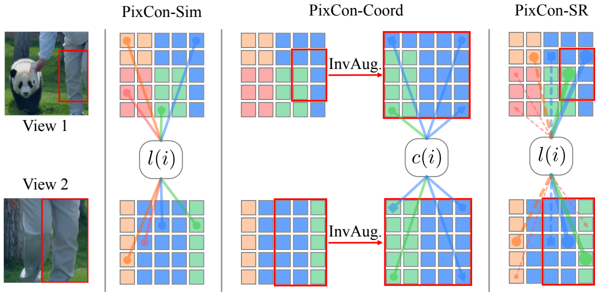

Figure 3. An illustration of different PixCon variants' matching schemes. The red bounding boxes indicate the intersected area of the two views. Girds' colors roughly indicate pixels' associated semantic classes for illustration purposes. We treat `view 1` as the `query` view and `view 2` as the `key` view. PixCon-Sim's matching scheme is the similarity-based matching in Eq. (3). PixCon-Coord uses the matching function in Eq. (5), and the involved inverse augmentation includes RoIAlign [17] and optional horizontal flipping depending on if the input is flipped. PixCon-SR uses similarity-based matching but applies the semantic reweighting in Eq. (7). For the illustration of PixCon-SR, solid lines indicate matches with `query` pixels in the red bounding box, dashed lines represent the rest of the matches, and different line widths indicate the magnitudes of semantic weights. The matches are drawn for illustration purposes, and not all are drawn for clarity.

i.e., $\mathcal{L}(\mathbf{x}_1, \mathbf{x}_2) + \mathcal{L}(\mathbf{x}_2, \mathbf{x}_1)$.

4.1. PixCon-Sim

Let the backbone networks' outputs be $\mathbf{F} \in \mathbb{R}^{S^2 \times C}$ and $\mathbf{F}' \in \mathbb{R}^{S^2 \times C}$ for the query and the key views, respectively, the spatial positions of features in $\mathbf{F}$ are matched to those in $\mathbf{F}'$ by

$$l(i) = \arg\max_j sim(\mathbf{F}(i), \mathbf{F}'(j)), \tag{3}$$

where $i, j \in [0, S^2 - 1]$ and $sim(\mathbf{a}, \mathbf{b}) = \mathbf{a}^\top \mathbf{b} / \|\mathbf{a}\| \|\mathbf{b}\|$. The similarity-based matching scheme aims to bootstrap feature similarities, *i.e.*, features with better semantic correlation give more semantically meaningful matches, which are in turn used to strengthen the correlation of such features. Similar bootstrapping strategies are also applied in region-level methods [21, 36, 41, 45].

With similarity-based matching, the pixel-level contrastive loss is then computed as follows

$$\mathcal{L}^l_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2) = -\frac{1}{S^2} \sum_i \log \frac{\exp(\mathbf{u}_i \cdot \mathbf{v}^+_{l(i)} / \tau)}{\sum\limits_{\mathbf{v} \in \{\mathbf{v}^+_{l(i)}\} \cup \mathcal{V}} \exp(\mathbf{u}_i \cdot \mathbf{v} / \tau)}, \tag{4}$$

where $\mathbf{u}_i = \mathbf{U}[i] \in \mathbb{R}^C$, $\mathbf{v}^+_{l(i)} = \mathbf{V}[l(i)] \in \mathbb{R}^C$, and $\mathcal{V}$ contains image-level negative key features from other images by following [35] for computational efficiency. The negative keys are stored in a fixed-length queue.

However, the matching function in Eq. (3) hardly makes sense at the beginning of training. As demonstrated in DenseCL [35], jointly conducting image-level and pixel-level learning can help mitigate the problem, as image-level learning also encourages the emergence of semantic relations among spatial features [4, 40]. Besides, image-level learning is also commonly conducted along with dense learning [23, 40, 41] and brings benefits. Therefore, by using $\mathcal{L}^l_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Eq. (2) and symmetrize the resulting loss w.r.t. the two views, we get the final loss for *PixCon-Sim*, *i.e.*, pixel-level contrastive learning with similarity-based matches. When using the MoCo-v2 pipeline instead of MoCo-v2+ and not using the symmetrized loss, PixCon-Sim becomes DenseCL [35].

4.2. PixCon-Coord

Though similarity-based matching gives increasingly better matches as the training proceeds [35], it still retrieves semantically inconsistent matches, especially at the beginning of training. To further investigate its pros and cons, we compare it with the coordinate-based matching scheme [29, 36, 41], which matches two cross-view spatial features only if they have (approximately) the same coordinates when mapped back to the input image space, thus guaranteeing the semantic consistency among the positive matches.

Therefore, we propose another variant of PixCon using *coordinate-based* matching based on the inverse augmentation [36], which involves RoIAlign [17] and horizontal flipping if the input image has been flipped. The schematic illustrations of both the similarity-based matching and the coordinate-based matching are provided in Figure 3.

By slightly overloading the notations $\mathbf{U}$ and $\mathbf{V}$ as the pixel-level outputs from the inverse augmentation, we have the corresponding pixel-level loss $\mathcal{L}^c_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$, which replaces the matching function l in Eq. (4) with c that is defined as

$$c(i) = i, \tag{5}$$

which connects the same positions in the two views' feature maps aligned by the inverse augmentation. By using $\mathcal{L}^c_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Eq. (2) and symmetrize the resulting loss w.r.t. the two views, we get the final loss for *PixCon-Coord*, *i.e.*, pixel-level contrastive learning with coordinate-based matches.

4.3. PixCon-SR

As shown in Figure 3, the two augmented views of the input multi-object image are semantically inconsistent, *i.e.*, the panda only appears in the first view. Thus, similarity-based matches for such view-specific objects' pixels will have different semantic classes.

While coordinate-based matching helps mitigate such false matches, it only matches cross-view pixel-level features at (approximately) the same spatial location in the input image. As a result, it fails to relate semantically related but spatially distant features, whereas pulling such features closer is crucial to learning regional semantics for better transfer performance [20, 23, 36, 40].

We notice that although similarity-based positive matches are cross-view features with maximal similarities, their similarities can still be low, indicating weak semantic correlations. However, some semantically related features can also have low similarities, constituting hard positive pairs that are important to leverage for better feature quality [42]. Therefore, we choose to fully trust a positive match whose query pixel lies in the intersection of two views regardless of the query-key similarity. We call such queries the "in-box" queries as the intersection area is always a box. The matched key for an in-box query is highly likely to be meaningful as the query is guaranteed to have semantic correspondences in the key view, *e.g.*, the same pixel itself in the key view in the worst case. We then reweight the matches with "out-of-box" queries by their query-key similarities readily available during the matching process. A schematic illustration of the reweighting process is provided in Figure 3.

We term the consequent reweighting strategy *semantic reweighting*, with which the pixel-level loss becomes

$$\mathcal{L}^{l,w}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2) = -\sum_i \frac{w(i)}{A} \log \frac{\exp(\mathbf{u}_i \cdot \mathbf{v}^+_{l(i)}/\tau)}{\sum\limits_{\mathbf{v}\in\{\mathbf{v}^+_{l(i)}\}\cup\mathcal{V}} \exp(\mathbf{u}_i \cdot \mathbf{v}/\tau)}, \quad (6)$$

where $A = \sum_i w(i)$ is the the normalization factor. Let $\mathcal{Y}$ be the set of indices of the in-box query features, which can be easily obtained during data augmentation, we compute $w(i)$ by

$$w(i) = \begin{cases} 1, & \text{if } i \in \mathcal{Y}. \\ norm(\max\limits_j sim(\mathbf{F}(i), \mathbf{F}'(j)))^{\alpha}, & \text{otherwise.} \end{cases} \quad (7)$$

where $norm(x) = (x - \min\limits_{j\notin\mathcal{Y}} w(j))/(\max\limits_{j\notin\mathcal{Y}} w(j) - \min\limits_{j\notin\mathcal{Y}} w(j))$ guarantees the continuity of weights and enlarges their contrast, and α is for further sharpening the contrast and is set to 2 by default. Note that the formulation of Eq. (6) is not involved with the inverse augmentation, which is more computationally expensive, *i.e.*, $\mathbf{U}$ and $\mathbf{V}$ are dense outputs from f_θ and f_ξ. By using $\mathcal{L}^{l,w}_{\text{pix}}(\mathbf{x}_1, \mathbf{x}_2)$ as the pixel-level loss in Eq. (2) and symmetrize the resulting loss w.r.t. the two views, we get the final loss for *PixCon-SR*, *i.e.*, pixel-level contrastive learning with semantic reweighting.

PixPro [41] also simultaneously utilizes spatial information and feature similarities. However, they use spatial information to retrieve positive matches, whose quality highly depends on the pre-defined size of a spatial neighborhood. We impose no spatial constraint on the positive matches at all and only bootstrap feature similarities. Due to the use of spatially close positive matches, they need to use self-attention maps to relate spatially distant pixels, whereas we merely rely on pixel-level features together with default random cropping and the inherent uncertainty of similarity-based matching to achieve this purpose.

5. Experiments

5.1. Experimental Settings

Datasets. For pre-training, as we are mainly interested in pre-training with real-world scene images containing diverse and complex contents, we use the training set of MS COCO [24], which contains ∼118k images and is broadly used for scene-level pre-training. COCO is also widely used for benchmarking dense prediction tasks such as object detection, instance segmentation, and semantic segmentation. Moreover, a COCO image contains 7.3 objects on average, which is in stark contrast to meticulously curated ImageNet [10] images, for which the number of objects per image is 1.1 [35].

Architecture. We follow the architecture of MoCo-V2+ [22]. Following [35], we add dense learning branches to the global learning branches. Specifically, the online encoder has a ResNet50 [18] backbone, which is appended with a global projection head and a dense projection head. The former has two fully connected layers, while the latter has two 1×1 convolutional layers. Both heads have batch normalization followed by ReLU in between the two layers. For both heads, The hidden dimensionality and the output dimensionality are 2048 and 128, respectively. The global and dense heads are appended with their respective predictors, which have the same architectures as the heads with an input dimensionality of 128. The target encoder has the same architecture as the online encoder except that it does not have predictors.

Data Augmentation. The pre-training data augmentation follows [15], where each image is randomly cropped into two views which are then resized to 224×224, followed by random horizontal flipping, color distortion, Gaussian blur, and solarization. Crops without overlapping are skipped.

Pre-training setup. Following [35], the negative-storing queues for both global learning and dense learning are of length 65536. The momentum for updating the target encoder is initially set to 0.99 and increased to 1 at the end of training [15]. Synchronized batch normalization [43] is used for all batch normalization layers [15]. The temperature τ is set to 0.2. We use the SGD optimizer with an initial learning rate of 0.4 and a cosine learning rate decay schedule. We set the weight decay to 0.0001 and the momentum for the optimizer to 0.9. We train each model for 800 epochs on COCO with 4 GPUs and a total batch size of 512. The

Table 1. Main transfer results. All self-supervised models have been pre-trained for 800 epochs on COCO, except that DetCon has been trained for 1000 epochs. Among all the methods, MoCo-v2 and DenseCL are based on the MoCo-v2 pipeline, while the others are based on the BYOL pipeline. Refer to Section 4 for more details on the differences between the pipelines. We also categorize the methods into different types based on their training strategies, including image level, region level, and pixel level. Refer to Table 2 for more information about region- and pixel-level methods. On all the benchmarks, our method shows strong transfer performance. We use boldface to indicate single best results but underline multiple best results that have the same value. (†: re-impl. w/ official weights. ‡: full re-impl.)

Method	Type	VOC detection			COCO detection			COCO instance seg.			City. Seg.	VOC Seg.
		AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}	AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}	mIoU	mIoU
Random init. [35, 36]	-	32.8	59.0	31.6	32.8	50.9	35.3	29.9	47.9	32.0	65.3	39.5
MoCo-v2 [5]	Image	54.7	81.0	60.6	38.5	58.1	42.1	34.8	55.3	37.3	73.8	69.2
BYOL‡ [15]		55.7	81.8	61.6	39.5	59.4	43.3	35.6	56.6	38.2	75.3	70.2
MoCo-v2+‡ [22]		54.6	81.4	60.5	39.8	59.7	43.6	35.9	57.0	38.5	75.6	71.1
ORL† [40]	Region	55.8	82.1	62.3	40.2	60.0	44.3	36.4	57.4	38.8	75.4	70.7
PixPro [41]		-	-	-	40.5	60.5	44.0	36.6	57.8	39.0	75.2	72.0
DetCon [20]		-	-	-	39.8	59.5	43.5	35.9	56.4	38.7	76.1	70.2
UniVIP [23]		56.5	82.3	62.6	$\underline{40.8}$	-	-	$\underline{36.8}$	-	-	-	-
Odin‡ [21]		56.9	82.4	63.3	40.4	60.4	44.6	36.6	57.5	39.3	75.7	70.8
DenseSiam [45]		55.5	81.1	61.5	-	-	-	-	-	-	-	-
SlotCon† [36]		54.5	81.9	60.3	$\underline{40.8}$	$\underline{61.0}$	$\underline{44.8}$	$\underline{36.8}$	**58.0**	39.5	76.1	71.7
DenseCL [35]	Pixel	56.7	81.7	63.0	39.6	59.3	43.3	35.7	56.5	38.4	75.8	71.6
PixCon-Sim (ours)		57.3	82.4	63.9	40.5	60.5	44.2	36.6	57.5	39.2	76.1	72.6
PixCon-Coord (ours)		57.2	82.6	63.4	40.3	60.3	43.9	36.5	57.4	39.2	75.8	72.3
PixCon-SR (ours)		**57.6**	**82.8**	**64.0**	$\underline{40.8}$	$\underline{61.0}$	$\underline{44.8}$	$\underline{36.8}$	57.9	**39.6**	**76.6**	**73.0**

Table 2. Comparisons between region- and pixel-level methods. While most of the region-level methods require object priors, multi-stage training, or prototype learning, pixel-level methods need none of them.

Method	Scheme	Obj. Prior	Multi-stage	Proto.
ORL [40]	Region-level	✓	✓	×
PixPro [41]		×	×	×
DetCon [21]		✓	×	×
UniVIP [23]		✓	×	×
Odin [21]		×	✓	×
DenseSiam [45]		×	×	✓
SlotCon [36]		×	×	✓
DenseCL [35]	Pixel-level	×	×	×
PixCon-$*$		×	×	×

training is conducted under the MMSelfSup framework [8].

Evaluation settings. To evaluate feature transferability, we follow previous work [3, 16, 23, 35, 36] to fine-tune the pre-trained models on target downstream tasks and evaluate the resulting models by reporting the metrics used in the corresponding tasks, including VOC object detection [12], COCO object detection, COCO instance segmentation [24], VOC Semantic segmentation [12], and Cityscapes semantic segmentation [9].

For VOC object detection, We fine-tune a Faster R-CNN with a C4-backbone. The training is done on the VOC `trainval07+12` set for 24k iterations. The evaluation is done on the VOC `test2007` set. Both training and evaluation use the Detectron2 [37] code base.

For COCO object detection and instance segmentation, we fine-tune a Mask R-CNN with an FPN backbone on COCO's `train2017` split with the standard 1× schedule and evaluate the fine-tuned model on COCO's `val2017` split. Following previous work, we synchronize all the batch normalization layers. Detectron2 is used to conduct the training and evaluation.

We strictly follow the settings in [36] for VOC and Cityscapes semantic segmentation. Specifically, an FPN is initialized with the pre-trained model, fine-tuned on the `train_aug2012` set for 30k iterations, and evaluated on the `val2012` set. For Cityscapes, we conduct fine-tuning on the `train_fine` set for 90k iterations and evaluate the fine-tuned model on `val_fine`. The training and evaluation are conducted using MMSegmentation [7].

The results, including ours and reproducible previous methods', are reported as the average of 5, 3, 3, and 5 independent runs for VOC detection, COCO detection and instance segmentation, Cityscapes segmentation, and VOC segmentation, respectively.

5.2. Main Results

As discussed in Section 4.1, PixCon-Sim boils down to DenseCL [35] when not applying the BYOL pipeline, which is, however, invariantly used by the region-level methods in Table 2. As per Table 1, PixCon-Sim outperforms DenseCL across all the benchmarks. Besides, with a simple pixel-level learning algorithm, PixCon-Sim is already competitive compared to region-level methods across all the benchmarks. PixCon-Coord, with a geometric matching scheme, is also competitive.

For all four tasks, PixCon-SR brings consistent performance boosts to its image-level baseline MoCo-v2+ and surpasses previous region-level methods as well as the other two PixCon variants. Though PixCon-SR's performance on COCO detection and instance segmentation is similar to that of UniVIP [23] and SlotCon [36], it has better performance in terms of the other three tasks. It achieves this without relying on any region-mining algorithms, most of which resort to complex preprocessing or computationally expensive multi-stage training. Specifically, for prototype-based methods, *i.e.*, DenseSiamese [45] and SlotCon [36], their transfer performance on VOC detection is conspicuously lower than that of the other methods. This is likely caused by the fact that the dense features are trained to cluster around a fixed number of prototypes, which may cause the features to be overfitted to the prototypes and thus may hurt the transfer performance due to overly small intra-class variances [46]. The pre-training based on a specific number of prototypes also struggles to serve multiple downstream tasks equally well [36]. Overall, Table 1 sufficiently indicates the potential of pixel-level learning and the effectiveness of PixCon-SR.

5.3. Ablation Study

In this section, we provide detailed quantitative and qualitative analyses of the key components in our PixCon framework, namely the two pixel-level feature matching schemes as well as the semantic reweighting strategy. We also explore how the additional blocks that build MoCo-v2 into MoCo-v2+ affect the transfer performance of PixCon in the supplementary material.

Similarity-based matching encourages learning regional semantics. Compared to similarity-based matching used for PixCon-Sim, the coordinate-based matching of PixCon-Coord guarantees the semantic consistency between the positive matches, as the matches represent the same patch in the image, which undergoes different augmentations. However, such strict geometric matching does not encourage relating spatially distant pixels associated with the same object and is thus limited in learning regional semantics.

Though similarity-based matches do not always enjoy such geometric proximity, their semantic consistency becomes increasingly better as training proceeds if the query feature has semantic correspondences in the key view [35]. For query pixels not lying in the intersection of two views, *i.e.*, out-of-box queries, their matches in the key view are guaranteed to be spatially apart from them. When such matches are semantically related, they could strengthen the correlation of spatially distant pixels belonging to the same semantic group. A qualitative investigation in the form of self-attention maps is provided in Figure 4, where semantically related but spatially distant pixel features are more holistically correlated for PixCon-Sim than for PixCon-Coord and MoCo-v2+. Moreover, Table 1 shows that PixCon-Sim delivers better transfer performance compared to PixCon-Coord, which may be attributed to the better regional semantics made possible by the similarity-based matching.

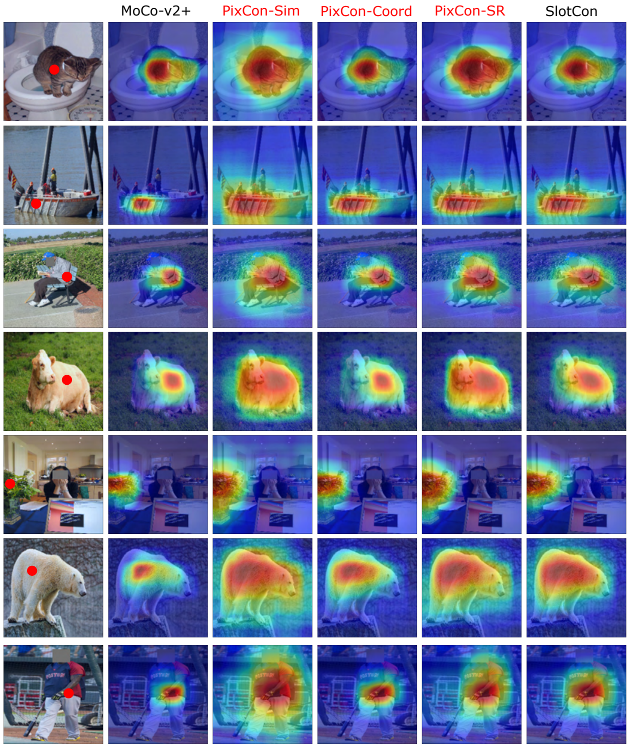

Figure 4. Visualizations of self-attention maps. For each row, the first image is the original image, with the red dot highlighting the pixel whose feature is used to calculate the cosine-similarity-based self-attention maps. The subsequent images are self-attention maps using different models' features. See main texts for analyses.

Semantic reweighting helps learn better regional semantics. The semantic reweighting strategy of PixCon-SR in Section 4.3 aims to discount the influence of inaccurate matches caused by semantically inconsistent views of scene images while utilizing as many semantically consistent matches as possible. Therefore, we expect the resulting features to be less correlated when they are associated with different semantic classes and have better intra-class

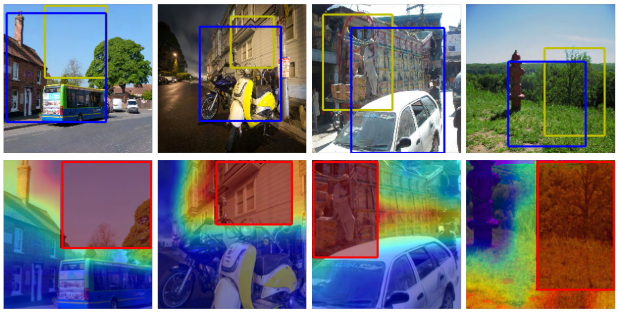

Figure 5. Visualizations of semantic weights. The first row shows the raw images with the blue bounding boxes indicating the query views and yellow bounding boxes the key views. The second row shows the heatmap of semantic weights for the query pixels (in the blue bounding box), where the red bounding boxes indicate the intersection between query and key views. All images and heatmaps are resized to the same size for visualization purposes.

coherence. Indeed, Figure 4 shows that PixCon-SR's self-attention maps have better localization of semantic objects compared to PixCon-Sim (less attention on features of different semantic classes) while guaranteeing a sufficient coverage of the whole objects (better intra-class cohesion) even when compared to the region-level method SlotCon [36]. Moreover, as shown in Table 1, PixCon-SR achieves better transfer performance compared to PixCon-Sim and PixCon-Coord as well as previous region-level methods, which further indicates the efficacy of the semantic reweighting strategy in helping learn decent regional semantics crucial for better transfer performance. Figure 5 provides visualizations of the semantic weights for the query features, where we can observe that the semantic contents not shared by the two views are given small weights and out-of-box query pixels with semantic correspondences in the key view are assigned with nontrivial weights.

Designs of semantic reweighting. In Eq. (7), spatial information is used to fully utilize matches with better guarantees for their semantic consistency regardless of their feature similarities, as their queries, *i.e.*, in-box queries, are present in the two views' intersected part and thus always have semantic correspondences in the key view. Besides, feature similarities are used to reweight the matches with out-of-box queries to diminish the effect of semantically inconsistent ones while exploiting those that are still informative. Table 3 ablates the impact of these two tools. Interestingly, when using similarity-based matches with in-box queries alone, PixCon-SR (Spa.) achieves slightly better performance than PixCon-Coord, which also merely utilizes matches having in-box queries but with coordinated-based matching. This indicates that similarity-based matching provides matches with sufficient semantic consistency. While only using either spatial information or feature similarities does not give apparent performance gain, combining them, *i.e.*, PixCon-SR (full), offers immediate benefits to the transfer performance, indicating the importance of sufficiently leveraging informative positives and mitigating the influence of false positives simultaneously.

Table 3. We ablate the influence of the tools used to formulate the semantic weights in Eq. (7). *PixCon-SR (Spa.)* means only matches whose query features lie in two views' intersected parts are accepted, and other matches have weights 0. Here only the spatial information is used for formulating the semantic weights. *PixCon-SR (Sim.)* means only the similarities between the matched features are used as semantic weights regardless of whether the query features exist in the two views' intersected area. *PixCon-SR (full)* utilizes both tools. The effect of the sharpening factor α in Eq. (7) is also investigated here.

Method	α	COCO		VOC Seg.
		AP^{bb}	AP^{mk}	mIoU
PixCon-Sim	-	40.5	36.6	72.6
PixCon-Coord	-	40.3	36.5	72.3
PixCon-SR (Spa.)	2	40.5	36.5	72.5
PixCon-SR (Sim.)	2	40.3	36.4	72.3
PixCon-SR (Full)	2	40.8	36.8	73.0
PixCon-SR (Full)	1	40.5	36.5	73.2
PixCon-SR (Full)	4	40.5	36.6	73.0

Effect of the sharpening factor α. As shown in Table 3, the sharpening factor α does not cause drastic fluctuations in the transfer performance, but a value of 2 helps strike a good balance between detection and semantic segmentation tasks, which is then applied as the default value.

6. Conclusion

In this paper, we revisit pixel-level contrastive pretraining on scene images. We show that pixel-level learning can be further exploited to match the transfer performance of more sophisticated region-level methods. Moreover, we find that pixel-level learning with similarity-based matching [35] already learns regional semantics that region-level methods are after. Finally, we propose a simple and effective semantic reweighting strategy to deal with the problem of semantically inconsistent crops of scene images, which helps pixel-level learning outperform or rival SOTA region-level methods on various transfer tasks. We believe pixel-level learning still has undiscovered potential yet with an attractively simple design, and we will keep exploring its possibility in future work.

7. Acknowledgement

This work was partly supported by JSPS KAKENHI Grant No. JP23H00497, JST CREST Grant No. JPMJCR20D3, and FOREST Grant No. JPMJFR216O.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 1

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 6

[4] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. In *NeurIPS*, 2021. 4

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2, 6

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 3

[7] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020. 6

[8] MMSelfSup Contributors. MMSelfSup: Openmmlab self-supervised learning toolbox and benchmark. `https://github.com/open-mmlab/mmselfsup`, 2021. 6

[9] 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 *CVPR*, 2016. 2, 6

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1, 5

[11] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[12] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 2, 6

[13] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59:167–181, 2004. 2

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[15] 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. In *NeurIPS*, 2020. 1, 2, 3, 5, 6

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3, 6

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017. 4

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 5

[19] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*. PMLR, 2020. 1, 2

[20] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, 2021. 2, 3, 5, 6

[21] Olivier J Hénaff, Skanda Koppula, Evan Shelhamer, Daniel Zoran, Andrew Jaegle, Andrew Zisserman, João Carreira, and Relja Arandjelović. Object discovery and representation networks. In *ECCV*, 2022. 1, 2, 3, 4, 6

[22] Junqiang Huang, Xiangwen Kong, and Xiangyu Zhang. Revisiting the critical factors of augmentation-invariant representation learning. In *ECCV*, 2022. 3, 5, 6

[23] Zhaowen Li, Yousong Zhu, Fan Yang, Wei Li, Chaoyang Zhao, Yingying Chen, Zhiyang Chen, Jiahao Xie, Liwei Wu, Rui Zhao, et al. Univip: A unified framework for self-supervised visual pre-training. In *CVPR*, 2022. 1, 2, 3, 4, 5, 6, 7

[24] 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 *ECCV*, 2014. 1, 2, 3, 5, 6

[25] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 2

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[27] Tam Nguyen, Maximilian Dax, Chaithanya Kumar Mummadi, Nhung Ngo, Thi Hoai Phuong Nguyen, Zhongyu Lou, and Thomas Brox. Deepusps: Deep robust unsupervised saliency prediction via self-supervision. In *NeurIPS*, 2019. 2

[28] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[29] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. In *NeurIPS*, 2020. 2, 4

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 1

[31] Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. In *CVPR*, 2021. 1, 2

[32] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 2

[33] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc V Gool. Revisiting contrastive methods for unsupervised learning of visual representations. In *NeurIPS*, 2021. 1, 3

[34] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *ICML*, 2020. 1, 2

[35] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[36] Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[37] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019. 6

[38] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 1, 2

[39] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, 2016. 3

[40] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. In *NeurIPS*, 2021. 1, 2, 3, 4, 5, 6

[41] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021. 1, 2, 4, 5, 6

[42] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning. In *ECCV*, 2022. 5

[43] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *CVPR*, 2018. 3, 5

[44] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[45] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. Dense siamese network for dense unsupervised learning. In *ECCV*, 2022. 1, 2, 3, 4, 6, 7

[46] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 7