Deeply Unsupervised Patch Re-Identification for Pre-Training Object Detectors

Jian Ding, Enze Xie, Hang Xu, Chenhan Jiang, Zhenguo Li, Ping Luo, and Gui-Song Xia

Abstract—Unsupervised pre-training aims at learning transferable features that are beneficial for downstream tasks. However, most state-of-the-art unsupervised methods concentrate on learning global representations for image-level classification tasks instead of discriminative local region representations, which limits their transferability to region-level downstream tasks, such as object detection. To improve the transferability of pre-trained features to object detection, we present Deeply Unsupervised Patch Re-ID (DUPR), a simple yet effective method for unsupervised visual representation learning. The patch Re-ID task treats individual patch as a pseudo-identity and contrastively learns its correspondence in two views, enabling us to obtain discriminative local features for object detection. Then the proposed patch Re-ID is performed in a deeply unsupervised manner, appealing to object detection, which usually requires multi-level feature maps. Extensive experiments demonstrate that DUPR outperforms state-of-the-art unsupervised pre-trainings and even the ImageNet supervised pre-training on various downstream tasks related to object detection.

Index Terms—Self-supervised learning, visual representation learning, contrastive learning, object detection

1 INTRODUCTION

Pre-training then fine-tuning has been a widely used paradigm when approaching computer vision problems with deep models [1], [2], [3], [4]. In recent years, fine-tuning in object detection tasks has been dominated by the ImageNet supervised pre-training [1], [3], [5], [6]. However, there exists misalignment between image-level classification pre-training and object detection tasks, which make region-level predictions. One solution to eliminate the misalignment [7] is to pre-train representations directly on a large-scale and high-quality object detection dataset; but annotations of such datasets are time-consuming, laborious, and even hard to obtain for some areas.

Alternatively, unsupervised learning aims at pre-training representations without human annotations, which allows us to pre-train representations with plenty of unlabeled data for free [8], [9], [10], [11], [12], [13]. Among them, contrastive learning methods [8], [14], [15] have achieved comparable performance compared to ImageNet supervised pre-training in many downstream tasks, such as image classification, object detection, and semantic segmentation. Contrastive learning can learn view-invariant representations by maximizing the similarity between positive pairs of views while minimizing the similarity between negative pairs.

Recent state-of-the-art contrastive learning methods take global views from the same image as positive pairs and views from different images as negative pairs. The well-designed augmentations [14], [16], including random resized crop, are used to generate these views. Thus globally spatial invariant representations will be learned, which are beneficial for classification. For example, classification models should predict the same category "dog" for two views generated from the same image, containing a dog at the bottom-left and top-left (see Fig. 1). So, these two views containing the same object but at different locations should have similar global representations.

The representations learned by contrasting global views can encode much information related to category and largely improve the performance on ImageNet linear evaluation, approaching the accuracy of supervised classification [14], [17], [18], [19].

However, there is a gap between pre-training global representations and region-level downstream tasks, such as object detection. Being different from image-level classification, which predicts the category for an entire image by a globally spatial invariant feature, object detection is a region-level task which predicts categories and regression targets for multiple regions by the region features. The region features at different locations should be discriminative since the prediction targets for these regions are different. For example, regions of interest (RoIs) are assigned with foreground or background categories. Also, foreground RoIs overlapping objects at different positions should predict different regression targets. Based on the above reasons, features at
Thus, the previous methods that only optimize the single feature after globally averaged pooling are problematic for object detection since they do not learn discriminative local representations in a feature map. So the previous methods with higher performance on ImageNet classification do not always lead to better transfer performance on object detection [17], [19], [20] (see also in Table 5).

Moreover, previous works focus on learning discriminative features at the final layer (e.g., on the $32 \times$ feature map) [8], [14]. However, most deep learning based object detectors need to extract features from the multi-level representations (such as FPN [21] and PANet [22]). Thus, object detection requires discriminative features at different feature layers instead of only the final layer.

To address the aforementioned problems, we propose to pre-train region-level discriminative representations across multi-level feature maps for object detection by Deeply Unsupervised Patch Re-ID (DUPR). The task of patch Re-ID is to match the corresponding patch identity (denoted by the same number in Fig. 1) of two views, which is inspired by the person Re-ID that matches a human identity between cameras under different viewing conditions. By pre-training with patch Re-ID, the features of matched patches should be more similar than unmatched patches in local feature space (see Fig. 3), so the region representations at different locations in a feature map are discriminative and beneficial for region-level tasks such as object detection. Furthermore, we propose a deeply unsupervised training strategy to learn multi-level representations. Specifically, we extract features from different intermediate layers to construct both image-level and patch-level contrastive loss. Our DUPR is independent of the detailed self-supervised learning framework. We simply adopt the MoCo framework [8], [17] and InfoNCE [23] as loss function in this work, but patch Re-ID can also be used in other self-supervised learning frameworks [14], [19]. The whole pipeline is shown in Fig. 3.

Our contributions can be summarized as follows:

- We propose a self-supervised pretext task, named patch Re-ID that learns to match the same patch identity between two views to get region-level discriminative feature maps, which is tailored for object detection.

- We present a deeply unsupervised training strategy to improve the transferability of pre-trained models to object detection, which requires extracting features from multi-level feature maps for prediction.

- Our DUPR pre-training outperforms other unsupervised and supervised pre-training counterparts when serving as the initialization for fine-tuning. For example, when fine-tuning Mask R-CNN R-50-FPN on MS COCO [24], DUPR outperforms MoCo v2 [17] and supervised pre-training at all different iterations as shown in Fig. 2. More importantly, it outperforms the strong baseline MoCo v2 when serving as initialization for fine-tuning in other location-sensitive tasks, such as VOC [25] object detection (+2.4 AP$_{bbox}$), Cityscapes [26] semantic segmentation (+1.0 mIoU), and LVIS [27] instance segmentation (+1.0 AP$_{bbox}$).

2 RELATED WORK

2.1 Object Detection

The task of object detection is to locate objects in images and classify their categories. Unlike the image-level classification prediction task, object detection is a region-level prediction task: it needs to regress the location and classify the category of an object simultaneously for each region. In two-stage object detectors [3], [5], region features are extracted from the proposals (generated by selective search [28] or
RPN [5]) by RoI Align [3]. For one-stage object detectors [6], [29], [30], [31], region features are extracted from sliding windows.

The prediction targets are different for different regions in an image. For each region, the classification branch usually predicts the confidence score of categories (foreground and background in RPN [5], or categories of objects in one-stage detector [6] and R-CNN [1]). A region of interest (RoI) is assigned to be a positive example if it has an intersection-over-union (IoU) overlap with a ground-truth box above a threshold; otherwise, it is assigned to be a negative example. Therefore, the category prediction target is sensitive to the change of regions (when the IoU changes from below the threshold to above the threshold, or from one object to another object with a different category). The localization branch usually predicts the regression targets relative to the anchor (see definition of regression targets in [1], [29]), and different positive RoIs should predict different regression targets. For example, an RoI having high IoU overlap with an object should predict small regression targets, while an RoI having low IoU overlap with an object should predict large regression targets (see Fig. 5). So the local region representations at different locations should be mapped to far apart points in local feature space for object detection. Besides, object detection is also the combination of region-level classification and localization. Features need to be sensitive to the location of a feature map for object localization while maintaining the strong semantic information for classification at the same time.

Object detection also requires multi-level representations, as predictions are directly made by using multi-level features [29], or the fusion of multi-level feature maps [21], [22]. For example, FPN [21] is a widely used structure in object detection to handle the scale variations, which combines low-resolution, semantically strong features with high-resolution, semantically weak features via top-down connections. Thus, our work will focus on learning discriminative region-level features from multiple layers to pre-train representations, which are tailored for object detection.

2.2 Pre-Training for Object Detection

R-CNN [1] has shown that ImageNet supervised pre-training followed by domain-specific fine-tuning on a small dataset is an effective paradigm for learning high-capacity representations. Pre-training for object detection largely improves the performance on small dataset [1], [32] and also speeds up the convergence of object detectors [32]. However, ImageNet supervised pre-training is weak at localization and helps less if the downstream task is localization-sensitive [1], [32]. To obtain better pre-trained representations for object detection, Objects365 [7] is proposed. Pre-training on this large-scale and high-quality object detection dataset can significantly surpass the ImageNet supervised pre-training in convergence speed and mAP. Since annotations of object detection are expensive, weakly supervised pre-training [33] has been explored for object detection. However, the weakly supervised pre-training pipeline [33] is complicated and still needs annotations. In contrast to these works, our paper presents a method for unsupervised pre-training for object detection, which has rarely been studied before.

2.3 Self-Supervised Visual Representation Learning

Self-supervised visual representation learning leverages input data itself as supervision via pretext tasks. After pre-training, features are transferred to downstream tasks. Early pretext tasks for self-supervised representation learning include rotation prediction [34], relative location prediction [35] and jigsaw [36], etc. These hand-crafted pretext tasks achieve promising results but still have a large gap.
compared to ImageNet supervised pre-training. Recently, the most successful methods in self-supervised learning are contrastive learning [8], [14], [16], [18], [19], [20], [23], [37], [38], [39] via instance discrimination pretext task [8], [15], [40]. The core idea of contrastive learning is to pull together positive view pairs while pushing apart negative view pairs.

The success of contrastive learning relates to learning the invariant representations to a family of similar views (positive pairs). The selection of positive pairs and data augmentations on views is important, and varies in different methods. For example, CPC [23] and CPC v2 [39] take the context and future as positive pairs. Deep infomax [41] and AMIDIM [42] take the global and local features as two positive pairs. MoCo [8] and SimCLR [14] adopt the instance discrimination task [15], [40] which takes the randomly augmented global views from the same image as two positive pairs. SimCLR has also studied many data augmentation strategies to generate views. SwAV has additionally proposed a novel multi-crop data augmentation, which increases the number of local views and maximizes the similarity between global and local views. The influence of augmentations and view selection for different downstream tasks has been studied in [16], [43]. Different downstream tasks have different optimal selections of positive view pairs. It has been proven in [16] that the optimal views should share the minimum information necessary to perform well at the downstream tasks. However, InfoMin [16] still takes the classification as the downstream task and designs augmentations to improve the classification performance.

Most of the previous methods maximize the similarity between spatial misaligned views and focus on learning globally spatial invariant representations for image-level classification, although the details are different. However, the better classification accuracy of linear probing does not always lead to better transfer performance to object detection. For example, SwAV [20] and BYOL [19] are much higher than MoCo v2 [17] in ImageNet linear probing, but lower in the transferring performance on object detection (see Table 5). Being different form these works, our approach maximizes the similarity between the spatially consistent local and local views, and aims at learning spatial sensitive, multi-level, entire feature maps for region-level tasks such as object detection. In the way of extracting local views, most of the previous methods send extra patches from the initial image to the network, such as SwAV [20] and CPC [23], [39]. In contrast, our method directly extracts the local representation from the feature map, which is more efficient.

### 2.4 Unsupervised Dense Representation Learning

There are some early works related to pixel or region-level representation learning. The auto-encoders [44], [45], [46] are trained by generating or modeling pixels in the input space. However, pixel-level generation is computationally expensive and needs extra heavy decoders which are not used by the downstream tasks. And the cycle-consistency of time in video sequences [47], [48] use self-supervised tracking as a pretext task and learns the pixel-level correspondence in the video sequences. However, these works aim at directly being deployed in visual correspondence tasks in videos without fine-tuning, rather than being transferred to other downstream tasks.

Recently, VADeR [49] has explored the pixel-level contrastive learning for transferring to multiple dense prediction tasks. But the VADeR requires the initialization of MoCo [8] and only optimizes the local representation; our model is trained from scratch, and optimizes both global and local representation. The VADeR [49] also does not perform well at object detection.

Being concurrent to our work, there are several self-supervised learning methods [50], [51], [52], [53], [54] targeting at object detection and semantic segmentation. InstLoc [50] crops two spatially misaligned patches and pastes them on two background images to form two positive views for contrastive learning, which is quite different from our method. DetCo [52] focuses on the trade-off between classification and object detection tasks, and proposes to use global and local contrastive learning. Self-EMD [51], DenseCL [54], and PixPro [53] are three dense self-supervised learning methods. Self-EMD [51] and DenseCL [54] find the positive pairs of patch features by the similarities between patch features, which are unstable if the initialization is not good. Different from them, DUPR finds the positive patches by their locations in the original images, which is more accurate and stable.

PixPro [53] is mostly related to our work since it also finds the positive pairs of patches by locations in the original images; but we are different in the details for the construction of positive pairs, as shown in Fig. 4. PixPro [53] finds the matched patches in two spatially misaligned grids.
I. So we can naturally obtain the

| Table 1 |

| pre-train | epoch | AP | AP50 | AP75 |
|-----------|-------|----|------|------|
| random init | -     | 33.8 | 60.2 | 33.1 |
| supervised | 100   | 33.5 | 81.3 | 58.8 |
| InsDetS* [19] | 200   | 55.2 (+1.7) | 80.9 (-0.4) | 61.2 (+2.4) |
| BYOL [19] | 200   | 55.3 (-1.5) | 81.4 (-0.1) | 61.1 (+1.9) |
| PIRL* [58] | 200   | 55.5 (-2.0) | 81.0 (-0.3) | 61.3 (+2.3) |
| SimCLR [14] | 200   | 55.5 (-2.0) | 81.6 (+0.5) | 61.4 (+2.6) |
| SwAV [20] | 800   | 56.1 (+2.6) | 82.6 (+1.3) | 62.7 (+3.9) |
| BoWNet [59] | 200   | 55.8 (+2.3) | 81.3 (+0.0) | 61.1 (+2.3) |
| SimSiam [18] | 200   | 57.0 (+3.5) | 82.4 (+1.1) | 63.6 (+4.9) |
| MoCo [8] | 200   | 55.9 (+2.4) | 81.5 (+0.2) | 62.6 (+3.8) |
| MoCo v2 [17] | 200   | 57.0 (+3.5) | 82.4 (+1.1) | 63.6 (+4.8) |
| MoCo v2 [17] | 800   | 57.4 (+3.9) | 82.5 (+1.2) | 64.0 (+5.2) |
| Infinom [16] | 200   | 57.6 (+4.1) | 82.6 (+1.3) | 64.3 (+5.5) |
| DeCo [52] | 200   | 57.8 (+4.3) | 82.6 (+1.3) | 64.2 (+5.4) |
| InstLoc [50] | 200   | 57.9 (+4.4) | 82.9 (+1.6) | 64.9 (+6.1) |
| DenseCL [54] | 200   | 58.7 (+5.2) | 82.8 (+1.5) | 65.2 (+6.4) |
| PixPro [53] | 200    | 58.8 (+5.3) | 83.0 (+1.7) | 66.3 (+7.7) |
| DUPR (ours) | 200   | 59.0 (+5.5) | 83.2 (+1.9) | 66.0 (+7.2) |

We fine-tune the Faster R-CNN R-50-C4 on Pascal VOC trainval2007+12 and evaluate it on test2007. We show the gap compared to the ImageNet-supervised counterpart in brackets. The increases of at least 0.5 are in green. The results of random init, supervised, and MoCo v2 are from [8]. The results of BYOL [19] and SimCLR [14] are from [18]. '*' means that results are implemented by us. Other methods are from their original papers. Our method outperforms the supervised counterpart by 5.5 points and the MoCo v2 baseline by 2.0 points. 200*: the model is trained with 100 epochs but with symmetric loss [19], which makes one more network forward pass for each image; therefore, the number of network’s forward pass is the same as that of MoCo v2 [17] with 200 epochs.

and a threshold of the distance between misaligned patches is introduced. In contrast, we apply RoI Align [3] on the intersection of two views and generate feature maps with spatially aligned grids. So we can naturally obtain the matched patches, and do not need an extra hyperparameter. PixPro [53] also introduces a pixel propagation module (PPM) to further improve the performance. Being simpler than PixPro [53], our method performs slightly better, as shown in Tables 1 and 2.

3 Method

3.1 Preliminary: Contrastive Learning

The main idea of contrastive learning is to pull together positive views while pushing apart negative views. Take MoCo [8] as an example, suppose $I$ is the original image, $I_1$ and $I_2$ can be considered as two views of the same image with different augmentations. Denote $v_1$ and $v_2$ to be the normalized embeddings of $I_1$ and $I_2$. The target of contrastive learning is to pull together positive pairs $(v_{1}, v_{2})$ while pushing apart negative pairs $(v_{1}, v_{2})$. The conventional learning objective is the InfoNCE [23] loss:

$$L_{v_{1}, v_{2}} = -\log \frac{\exp(v_1 \cdot v_2 / \tau)}{\sum_{j=0}^{M} \exp(v_1 \cdot v_{2j} / \tau)}.$$ (1)

Here $\tau$ is a temperature hyper-parameter. $v_1 \cdot v_{2j}$ is the cosine similarity to measure the distance between two image features. It can be considered as a non-parametric softmax-based classifier to identify $v_{2j}$ as $v_1$.

Selecting positive pairs is important [15], [41] for contrastive learning since they will learn invariant representation against the transformation applied on positive pairs. What kind of transformation should the representation be invariant to is determined by the downstream tasks, and varies across different downstream tasks [16], [43]. For example, suppose the downstream task is classification; in such case, the representation should be invariant to the location of an object. And, suppose the downstream task is to predict the location of an object. In this case, the representation should not be invariant to the locations. But other factors (e.g., category and light condition) are irrelevant information, to which the representation should be invariant.

In the previous works, $v_1$ and $v_2$, are pairs of global-local features [41], [42] or spatial misaligned global-global features [8], [14]. On the one hand, these global averaged features will lose spatial information. On the other hand, the positive pairs are not spatially aligned. Thus, these global view based methods tend to learn globally spatial invariant feature, which is suitable for image-level classification but not for region-level and location-sensitive tasks such as object detection. In contrast to the previous methods, our method can be considered as spatially aligned local-local views selection, where the representations are sensitive to the locations of objects, while invariant to other factors.

3.2 Deeply Unsupervised Patch Re-ID

The pipeline of DUPR is shown in Fig. 3, which consists of patch-level and image-level contrastive learning in parallel across multi-feature levels. The patch-level contrastive learning directly optimizes an entire feature map before averaged pooling and maximizes the similarity between matched patches (denoted by the same number in Fig. 3) to strengthen the spatial information for localization. Since object detection is the combination of localization and classification, we also include the image-level contrastive learning to strengthen the semantic information for classification. For the contrastive learning framework, we simply choose the MoCo v2 [17] as our strong baseline, although other contrastive learning frameworks are also possible.

Finally, we add the patch-level and image-level contrastive loss to multi-feature levels, as most object detectors need a multi-level representation, such as FPN [21] and PANet [22]. The overall loss is defined as:

$$L = \sum_{m=0}^{M} \alpha_m L^{(m)}_{image} + \sum_{m=0}^{M} \beta_m L^{(m)}_{patch},$$ (2)

where $M$ is the number of feature maps. $L^{(m)}_{image}$ and $L^{(m)}_{patch}$ are the image and patch contrastive loss for feature map of $n$th level. $\alpha_m$ and $\beta_m$ are the weights to balance the importance of different levels. We will describe the details of patch-level and image-level contrastive learning across multi-level feature maps in the following.

3.2.1 Patch-Level Contrastive Loss

The previous methods optimize global representations and do not learn discriminative local representations. However, for location-sensitive tasks, features of different patches in a feature map should be different since their prediction...
targets are different. For example, different regions may represent different categories of objects or the background. The regression targets for different regions are also different (see Fig. 5 for illustration). So the local features of the matched patches should be mapped to the nearby points in local feature space, while local features of different patches should be mapped to far apart points in local feature space as shown in Fig. 3.

To learn patch-level discriminative representations, we design the patch Re-ID pretext task to match the same patch identities of two views (which are denoted by the same number in Fig. 3). First, we sample two augmentations \((t \in T \text{ and } t' \in T)\) from the same family of augmentations: each augmentation is a composition of multiple transformations (e.g., cropping, resizing, flipping, and color distortion.) Then we apply augmentations to input image \(I\) and get two views: \(I_1 = t(I)\) and \(I_2 = t'(I)\). The corresponding rectangular regions of \(I_1\) and \(I_2\) in the original image \(I\) are recorded and denoted as:

\[
R_1 = (t_1^{(1)}, t_1^{(2)}) \quad R_2 = (t_2^{(1)}, t_2^{(2)})
\]

where the \((t_x, t_y)\) denotes the top-left vertex and \((b_x, b_y)\) denotes the bottom-right vertex of a rectangular region. Then we can calculate the intersection area \(B = (t_{\min}^{(1)}, t_{\max}^{(1)}, b_{\min}^{(2)}, b_{\max}^{(2)})\) from \(R_1\) and \(R_2\):

\[
\begin{align*}
    t_{\min}^{(1)} &= \max(t_1^{(1)}, t_2^{(1)}) \\
    t_{\max}^{(1)} &= \max(t_1^{(2)}, t_2^{(2)}) \\
    b_{\min}^{(2)} &= \min(b_1^{(1)}, b_2^{(1)}) \\
    b_{\max}^{(2)} &= \min(b_1^{(2)}, b_2^{(2)})
\end{align*}
\]

The intersection \(B\) in the coordinate system of \(I_1\) and \(I_2\) can be obtained by \(B_1 = T_1(B)\) and \(B_2 = T_2(B)\), where the \(T_1\) and \(T_2\) are coordinate transformations from \(I_1\) to \(I_1\) and \(I_2\), respectively. Instead of using a global averaged feature, which loses spatial information, we split \(B_1\) and \(B_2\) of the two views into \(S \times S\) patches and maximize the similarity between the corresponding patch features (denoted by the same number).

For the detailed implementation of patch feature map, we apply RoI Align [3] to extract region feature followed by a pixel-wise MLP layer which is implemented by a \(1 \times 1\) convolution as:

\[
\begin{align*}
    r_1^{(m)} &= g_1^{(m)}(RoIAlign(f_1^{(m)}(I_1), B_1)) \\
    r_2^{(m)} &= g_2^{(m)}(RoIAlign(f_2^{(m)}(I_2), B_2))
\end{align*}
\]

where \(g_1^{(m)}\) and \(g_2^{(m)}\) are MLP layer and momentum layer of patch feature map, respectively, \(r_1\) and \(r_2\) region features of a fixed shape \((C, S \times S)\). Then \(r_1^{(m)}\) and \(r_2^{(m)}\) is a positive pair of normalized feature vectors, where the subscript \(p \in [0, S \times S)\) denotes the position in the intersection area. Although our patch Re-ID pretext task is independent of loss functions, we simply adopt the InfoNCE [23] loss and follow MoCo v2 [17] to use a dynamic memory bank to store features from momentum updated encoder.

We construct the patch-level contrastive loss for the \(m\) feature map as:

\[
L_{\text{patch}}^{(m)} = - \sum_p \log \frac{\exp(r_1^{(m)}(p) \cdot r_2^{(m)}/\tau)}{\sum_{q=1}^K \exp(r_1^{(m)}(p) \cdot r_i^{(m)})} + \sum_{i=1}^K \exp(r_1^{(m)} \cdot r_i^{(m)})/\tau)
\]

where \(\{r_i^{(m)}\}_{i=1}^K\) are patch features of other images from memory bank. \(m\) denotes the index of feature map. By minimizing the patch-level contrastive loss, our encoder can learn patch-wise matching of identity between two views. Such matching ability results in spatially sensitive feature maps,
which facilitates object detection. Note that if there is no overlap between \( R_1 \) and \( R_2 \), we will ignore the patch-level contrastive loss. The probability of such a case is very low (e.g., 51 out of 1472 positive pairs), so it has not much effect on the results.

### 3.2.2 Image-Level Contrastive Loss

We also optimize the image-level contrastive loss as it is important to improve the classification ability, which is also required by object detection. Denote \( v_1^{(m)} = h_1(GAP(f_1^{(m)}(I_1))) \) and \( v_2^{(m)} = h_2(GAP(f_2^{(m)}(I_2))) \) to be the normalized image features of positive pairs. For simplicity, we neglect the notation of normalization. The image-level contrastive loss for the \( m \)th feature map can be written as:

\[
\mathcal{L}_{image}^{(m)} = -\log \frac{\exp(v_1^{(m)} \cdot v_2^{(m)}/\tau)}{\sum_{j=0}^{K} \exp(v_1^{(m)} \cdot v_j^{(m)}/\tau)}
\]  

(8)

### 3.2.3 Implementation Details

We use unlabelled ImageNet to pre-train our models for our experiments. For the ablation experiments, we follow the data augmentation settings in [17]. For the main experiments, we add the Rand-Augmentation [56] following [16]. We choose ResNet 50 [57] as our backbone and extract multi-level features from conv2_x, conv3_x, conv4_x and conv5_x. The stride of each feature map is \( \{4 \times 8 \times, 16 \times, 32 \times\} \), respectively. By default, we set \( \alpha_{0.3} = (0.1, 0.4, 0.7, 1.0) \) and \( \beta_{0.3} = (0, 0, 1, 1) \) for Eq. (2). The RoI size \( S \) of patch features on conv5_x and conv4_x are 14 and 7. \( \tau \) is 0.2 for ablation experiments and 0.15 for main experiments. Unless otherwise specified, we train with a batch size of 256 for 200 epochs. We use a learning rate of 0.06 with a cosine decay schedule.

We maintain a unique memory bank for each image-level and patch-level contrastive loss in Eq. (2). For \( \mathcal{L}_{patch}^{(m)} \), the memory bank stores the patch features of the \( m \)th feature map of other images. For \( \mathcal{L}_{image} \), the memory bank stores the image features of the \( m \)th feature map of other images. We store 65536 keys for each memory bank. For patch features, there are \( S \times S \) features for a single level on a single image, where \( S = 7 \) on conv5_x and \( S = 14 \) on conv4_x. In a batch with 256 images, there are 256 \( \times S \times S \) patch features. Since most patch features from a single image are similar, we sample 32 patch features per batch, when we enqueue and dequeue a batch of patch features.

All the algorithms of object detection, instance segmentation, and semantic segmentation are implemented in detectron2. For Mask R-CNN R-50-FPN, Mask R-CNN R-50-C4, and Faster R-CNN-C4, we follow the settings in MoCo [8]. For RetinaNet R-50-FPN, we also add an extra normalization layer similar to Mask R-CNN R-50-FPN in MoCo [8]. The pre-trained weights of PIRL [58], InsDis [15] are downloaded from PyContrast, while the pre-trained weights of SwAV is downloaded from the official code.

1. https://github.com/facebookresearch/detectron2
2. https://github.com/HobbitLong/PyContrast
3. https://github.com/facebookresearch/swav

### 4 Experimental Analysis

We evaluate our DUPR and compare it with recent state-of-the-art unsupervised and supervised counterparts in various object detection related downstream tasks. The results of PASCAL VOC [25] object detection are reported in Section 4.1. The results of COCO [24] object detection and instance segmentation are presented in Section 4.2. Other localization-sensitive tasks (COCO keypoint detection, Cityscapes [26] semantic segmentation, and instance segmentation, and LVIS instance segmentation [27]) are presented in Section 4.3. The comparisons with other methods in object detection versus classification performance are reported in Section 4.4. Then we analyze some ablation experiments in Section 4.5, and give the visualization of features in Section 4.6.

#### 4.1 Pascal VOC Object Detection

##### 4.1.1 Experimental Setup

PASCAL VOC [25] is a widely used small dataset for object detection, on which training from scratch cannot catch up the performance compared to the pre-trained counterparts even with longer training [32]. We fine-tune the Faster R-CNN with R-50-C4 backbone on Pascal VOC trainval07+12 and evaluate the results on test 2007. All the settings are the same as MoCo [8]. The RPN is built on conv_4x feature map and R-CNN is built on the conv_5x feature map in this detector. All the parameters of the network are fine-tuned end-to-end. The image size is [480, 800] in the training and 800 at inference. During fine-tuning, we train and synchronize all batch normalization layers. The batch normalization is used in the newly initialized RoI head layer. The fine-tuning takes a total of 24k iterations.

##### 4.1.2 Results Comparisons

The results in Table 1 show that DUPR outperforms other unsupervised methods and supervised counterparts. Most unsupervised methods outperform the supervised counterpart in AP 75 (which requires high localization accuracy), which indicates that the representations learned by supervised classification may lose much information irrelevant to classification but useful for localization. However, the previous unsupervised pre-training methods are still designed for classification. In contrast, our DUPR is designed to encode the spatial information explicitly. DUPR pre-training outperforms the MoCo v2 strong baseline by 2.4 points in AP 75 and 0.8 points in AP 50, and further significantly improves the localization accuracy. DUPR also obtains state-of-the-art performance in AP and AP 50. When compared to the pre-training by supervised classification, DUPR largely improves the AP 75 by 7.3 points. It verifies that DUPR contains more spatial information than MoCo v2 and ImageNet supervised pre-training.

#### 4.2 COCO Object Detection and Segmentation

##### 4.2.1 Experimental Setup

We compare the fine-tuning results of Mask R-CNN R-50-FPN, Mask R-CNN R-50-C4, and RetinaNet R-50-FPN with other unsupervised and ImageNet supervised counterparts, including both one-stage and two-stage detectors with
different backbones. We fine-tune these detectors on COCO train2017 with 118k images, and test on COCO val2017. For all detectors, input images are randomly resized to a scale within \([640, 800]\) during training and fixed at 800 for inference. All the layers are trained end-to-end.

For Mask R-CNN R-50-FPN and Mask R-CNN R-50-C4, we strictly follow the settings in MoCo [8]. For Mask R-CNN R-50-FPN and RetinaNet R-50-FPN, the batch normalization is used in the newly initialized FPN. Other parameters of RetinaNet follow the default setting of Detectron2 [60]. We explore the standard 1x/C2 and 2x/C2 schedule for these detectors. For Mask R-CNN R-50-FPN, we also compare the fine-tuned results with the strong baseline MoCo v2 [17] at fewer training iterations (12k, 18k, and 36k iterations) to study the convergence speed.

### 4.2.2 Mask R-CNN R-50-FPN

The results of Mask R-CNN, R-50-FPN, 1x schedule are shown in Table 2(a). DUPR outperforms other unsupervised methods and the supervised counterparts (e.g., surpasses MoCo v2 baseline by 1.1 points in mAP). In 2x schedule, DUPR outperforms MoCo v2 by 0.7 points in mAP and the supervised counterpart by 1.0 point in mAP as shown in Table 2(c).

### 4.2.3 Mask R-CNN R-50-C4

As shown in Table 2(b), in 1x schedule, DUPR outperforms all other unsupervised and supervised counterparts (e.g., surpasses MoCo v2 by 1.2 points in mAP). Compared to the ImageNet supervised pre-training, the gain in AP_{75} is larger than AP_{50} (2.4 points versus 1.9 points), indicating that DUPR pre-training improves the localization ability. In 2x schedule, where pre-training is less important, our method still outperforms the MoCo v2 by 0.5 points and the ImageNet supervised pre-training by 1.5 points in mAP as shown in Table 2(d).

### 4.2.4 RetinaNet R-50-FPN

We choose to fine-tune the RetinaNet [6] with R-50-FPN on COCO in 1x and 2x schedule. As shown in Table 3, in the 1x schedule, MoCo v2 has the same AP as the supervised counterpart. Our method outperforms MoCo v2 pre-training and supervised counterpart by 0.7 points in AP. Our method generalizes well on the one-stage object detector.

### 4.2.5 Fine-Tune With Fewer Iterations

Pre-training can speed up the convergence of the object detectors [32]. So, we explore the performance of different unsupervised pre-trained models, when serving as initialization for fine-tuning Mask R-CNN R-50-FPN at iterations of 12k, 18k, 36k, 90k, and 180k in Fig. 2. Our DUPR outperforms MoCo v2 and even the ImageNet supervised pre-training at all iterations. When fine-tuning with only 12k iterations, DUPR significantly outperforms the MoCo v2 by 2.9 points in mAP. It indicates that DUPR provides a better initialization and faster convergence speed than other methods. When fine-tuning with 90 iterations, DUPR still outperforms the supervised counterpart by 1.1 points.

### 4.3 Other Localization-Sensitive Tasks

#### 4.3.1 Instance Seg. on Cityscapes

Cityscapes [26] is a dataset that focuses on semantic understanding of urban street scenes. We fine-tune the Mask R-CNN R-50-FPN following the settings in MoCo [8]. Batch normalization is added before the FPN. All layers are trained end-to-end. Other hyperparameters follow the default settings of Detectron2 [60]. We fine-tune the model on the train_fine set (2975 images) for 90k iterations, and test on the val set. DUPR outperforms MoCo v2 by 0.5 points in mAP as shown in Table 4, which indicates the DUPR pre-training has a good generalization.

#### 4.3.2 Semantic Seg. on Cityscapes

For the semantic segmentation on Cityscapes, we follow the FCN-based structure and settings used in MoCo [8].

---

**TABLE 3**

| pre-train | AP_{50} | AP_{75} |
|-----------|---------|---------|
| random init. | 24.5 | 39.0 | 25.7 |
| supervised | 37.4 | 56.5 | 39.7 |
| MoCo v2 [17] | 37.4 (+0.0) | 56.5 (+0.0) | 40.0 (+0.3) |
| DUPR (ours) | 38.1 (+0.7) | 57.3 (+0.8) | 41.1 (+1.4) |

(a) RetinaNet, R50-FPN, 1x schedule

**TABLE 4**

| pre-train | Cityscapes Instance seg. | Cityscapes Seg. |
|-----------|--------------------------|-----------------|
| AP_{max} | AP_{50} | AP_{75} |
| random init. | 25.4 | 51.1 | 65.3 |
| supervised | 32.9 | 59.6 | 74.6 |
| MoCo v2 [17] | 33.9 (+1.0) | 60.8 (+1.2) | 75.7 (+1.1) |
| DUPR (ours) | 34.4 (+1.5) | 62.3 (+1.7) | 76.7 (+2.1) |

(b) RetinaNet, R50-FPN, 2x schedule

All the unsupervised models are pre-trained on ImageNet for 200 epochs. All the results are implemented by us in the same settings.
FCN-based structure consists of the convolutional layers in ResNet 50 and the $3 \times 3$ convolutions with dilation 2 and stride 1 in conv5_x stage. Then it is followed by two extra $3 \times 3$ convolutions of 256 channels with dilation 6, and a $1 \times 1$ convolution is added for per-pixel classification. Since there is no officially released code, we reimplement the FCN-based structure. DUPR outperforms MoCo v2 by 0.9 points as shown in Table 4. The results on Cityscapes indicate that DUPR pre-training transfers well to other region-level tasks.

### 4.3.3 Keypoint Detection on COCO

The task of keypoint detection is to simultaneously detect people and locate their keypoints. We fine-tune the Mask R-CNN R-50-FPN (keypoint version) on COCO train2017 and evaluate it on COCO val2017, following [8]. DUPR outperforms the supervised counterpart by 1.3 points in mAP as shown in Table 4.

### 4.3.4 Instance Seg. on LVIS-v1.0

LVIS [27] is an instance segmentation dataset, which has 1203 long tail distributed categories and provides high-quality segmentation masks. We fine-tune the Mask R-CNN R-50-FPN on LVIS train_v1.0 and test on LVIS val_v1.0. We train the model for a total of 180k iterations. DUPR outperforms MoCo v2 by 1.0 point in AP\text{mAPk} as shown in Table 4.

### 4.4 Object Detection versus Classification

Object detection includes both classification and localization. To better understand why DUPR improves object detection, we also report the results of linear probing on ImageNet for reference to obtain more insight. We compare the classification accuracy and detection mAP for various unsupervised pre-training methods in Table 5. We notice that under these settings, with the compared models, there exists little correlation between ImageNet classification and object detection performance. For example, MoCo v2 [17] is lower than BYOL [19] by 3.1 points in ImageNet accuracy but higher than BYOL [19] by 1.7 points in VOC AP. DUPR and DenseCL [54] have a drop of 3.7 points in ImageNet linear evaluation compared to the MoCo v2 baseline. This drop is likely resulted from the joint optimization of both patch-level and image-level contrastive loss, which is more difficult and will affect the optimization of image-level contrastive loss. A better balance between global representations for classification and local representations for localization is possible but is not the focus of this paper. We can conclude that the improvements of DUPR in object detection are not from the ability of better classification but the ability of better localization.

### 4.5 Ablation Experiments

#### 4.5.1 Experimental Setup

The ablation experiments are conducted on PASCAL VOC with Faster R-CNN R-50-C4 and COCO with R-50-FPN. We also report the SVM classification on PASCAL VOC, following the settings in [61], where the feature is fixed and used to train an SVM classifier on PASCAL VOC classification task.

#### 4.5.2 Influence of $\alpha_{0,3}$ for Image Contrastive Loss

In this ablation study, the $\beta_{0.3}$ is set as (0, 0, 0, 0). The results in Table 6(a) show: (1) all configurations of $\alpha_{0,3}$ can improve the AP in VOC detection, especially for the high IoU metric AP$_{75}$, suggesting that intermediate level contrastive loss can improve the localization ability; (2) only the configuration of (0.1, 0.4, 0.7, 1.0) has improvements on both VOC and COCO detection. The reasons why large weights (i.e., configuration of (1, 1, 1, 1)) on shallow layers decrease the performance on COCO detection are two aspects: (1) large weights on shallow layers will influence the optimization of deep layers, which is more important for classification than shallow layers; (2) COCO contains more classes than VOC, and the performance on COCO relies more on the classification ability. In fact, setting the deep layers with larger weights is a common practice in [62], [63], [64], [65], although their tasks are different from ours.

Now we study the classification ability of intermediate layers. It can be seen in Table 6(b) that all three configurations of $\alpha_{0,3}$ largely improve the classification performance of shallow...
In this ablation study, we set $\alpha_{0.3} = (0.1, 0.4, 0.7, 1.0)$, it improves the classification performance of conv3_x by 10 points and conv4_x by 6.3 points. We also notice a slight decrease in the classification performance of the conv5_x, which makes sense as the optimization of multi-level contrastive loss is more challenging than single-level one. Combining Tables 6(a) and 6(b) we can conclude: better representations of shallow layers can improve transfer performance, especially for the localization aspect of object detection.

4.5.3 Influence of RoI Size

In this ablation study, we set $\alpha_{0.3} = (0.1, 0.4, 0.7, 1.0)$ and $\beta_{0.3} = (0.0, 0.1)$. The results of different RoI sizes are presented in Table 7(a). We can see that $\beta_{0.3} = (0.0, 0.1)$ with RoI size of 1 does not have much difference compared to $\beta_{0.3} = (0.0, 0, 0)$ in object detection, where the patch Re-ID degenerates to global view contrastive learning and has no constraint on the local representations, although it is spatially aligned. Large RoI size improves the performance, which indicates that larger RoI size can get more discriminative region-level representations for object detection. But it is not always the larger, the better: if the RoI size is larger than the size of a feature map, it will not acquire more useful information. For example, the improvements become saturated when the RoI size is larger than 7, which is the size of conv5_x feature map. So we set RoI size as 7 on conv5_x by default. Similarly, the size of conv4_x is 14, so we set the RoI size as 14 on conv4_x by default.

4.5.4 Influence of $\beta_{0.3}$ for Patch Contrastive Loss

In this ablation study, we set $\alpha_{0.3} = (0.1, 0.4, 0.7, 1.0)$. Simply adding patch loss to conv5_x (by setting $\beta_{0.3} = (0.0, 0.1)$) improves the AP by 0.7 points in VOC and 0.3 points in COCO. And adding patch loss to both conv4_x and conv5_x (by setting $\beta_{0.3} = (0.0, 1, 1)$) feature maps further improves the AP by 0.5 points in COCO and 0.2 points in VOC as shown in Table 7(b), which indicates that intermediate supervision and patch contrastive loss are complementary.

4.6 Visualization

4.6.1 Visualization of Spatial Sensitivity

To verify that patch Re-ID can learn spatial-sensitive features, we draw the IoU-similarity curve: each point in the curve is obtained by calculating the IoU and similarity between a RoI and the ground truth box. We randomly select 1000 images from the ImageNet validation set. For each image, we sample 20 RoIs with different IoUs (range from 0 to 1) between the ground truth box and RoIs. We then use the RoI Align to extract region features from conv4_x feature map, according to the ground truth box and RoIs. Denote the feature extracted from a ground truth box as $q$ and a feature extracted from a RoI as $k_r$, where the $\alpha$ denotes the IoU between the ground truth box and the RoI. Both $q$ and $k_r$ are of shape $(C, S \times S)$. We calculate an average of cosine similarity between $q$ and $k_r$ as:

$$\text{Sim}(q, k_r) = \frac{1}{S \times S} \sum_{p=1}^{S \times S} q_p k_{r,p}$$

(9)
where $q_p$ and $k_{n_p}$ are normalized features at position $p$ of region features. We compare DUPR with MoCo v2 in Fig. 6. This visualization shows that DUPR is more sensitive to IoU: DUPR has a steeper slope than MoCo v2. When IoU between RoI and ground truth box is above 0.5 (usually assigned as positive samples in object detection [5]), DUPR has a higher similarity; when IoU is below 0.3 (usually assigned as negative samples), DUPR has a lower similarity. Such property makes the representations easy to suppress the false-negative RoIs.

4.6.2 Visualization of Correspondence

We use the learned representations to visualize the correspondence between two views. First, we create two views via random data augmentation following the settings in [8]. We set the image size to 448. We use the feature map of conv4_x (which is of shape $(1024, 28, 28)$) for matching. For each patch (one of $28 \times 28$) feature in one view, we find the patch with the highest similarity in the other view. For visualization, we use the point to represent a patch. The results shown in Fig. 7 demonstrate that patch Re-ID matches the corresponding patches better than MoCo v2.

Fig. 7. Visualization of Correspondence. We use pre-trained representations to match patches in two views, and compare the results of two self-supervised tasks: (1) patch-level contrastive learning (patch Re-ID); (2) image-level contrastive learning (MoCo v2). For fair comparison, both two tasks only add loss on the conv5_x with the same loss weight. For visualization, we use points to represent patches. It shows that representations of patch Re-ID can match the local patches in the other view better than MoCo v2.
5 Conclusion

This paper presents an unsupervised visual representation learning method, named DUPR, to bridge the gap between the unsupervised pre-training and downstream object detection tasks. Different from the previous methods that only learn discriminative image-level representations in the final layer, our method learns discriminative region-level multi-level representations. Therefore, our method outperforms other unsupervised models and even the supervised counterpart when transferred to downstream tasks related to object detection. Moreover, our method is robust to various object detectors, fine-tuning iterations. We hope our simple yet effective method could serve as a baseline of unsupervised pre-training for object detection tasks.

It is worth noticing that the DUPR presented in this paper learns representations for object detection from static images. While, rather than static images, we humans learn more from videos, which contain not only spatial configurations but also rich temporal information. Thus, it is of great interest to extend the proposed DUPR for learning spatial-temporal representations from videos in the future. Moreover, our current implementation of DUPR is based on MoCov2 [17] and needs multiple memory banks. To save the memory occupation, we will further study how to apply DUPR to other self-supervised learning methods (e.g., BYOL [19] or SimSiam [18]). Beyond the pretext task, how to design more suitable data augmentation pipeline for dense prediction tasks can also be studied in the future.

Acknowledgments

J. Ding and E. Xie have equally contributed to this work.

References

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587.

[2] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 3431–3440.

[3] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. Int. Conf. Comput. Vis., 2017, pp. 2961–2969.

[4] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” 2014, arXiv:1406.2199.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “ Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2015, pp. 91–99.

[6] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. Int. Conf. Comput. Vis., 2017, pp. 2980–2988.

[7] S. Shao et al., “Objects365: A large-scale, high-quality dataset for object detection,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 8430–8439.

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020, pp. 10.9729–10.9738.

[9] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 132–149.

[10] J. Donahue and K. Simonyan, “Large scale adversarial representation learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 10.542–10.552.

[11] I. Goodfellow et al., “Generative adversarial nets,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[12] C. Zhuang, A. L. Zhai, and D. Yamin, “Local aggregation for unsupervised learning of visual embeddings,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 6002–6012.

[13] G. E. Hinton and R. S. Zemel, “Autoencoders, minimum description length and helmholtz free energy,” in Proc. Int. Conf. Neural Inf. Process. Syst., 1994, pp. 3–10.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, “A simple framework for contrastive learning of visual representations,” in Proc. Int. Conf. Mach. Learn., 2020, pp. 1597–1607.

[15] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3733–3742.

[16] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good views for contrastive learning?,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 6852–6859.

[17] X. Chen, H. Fan, R. Girshick, and K. He, “Improved baselines with momentum contrastive learning,” 2020, arXiv:2003.04297.

[18] X. Chen and K. He, “Exploring simple siamese representation learning,” 2020, arXiv:2011.10566.

[19] J. Grill et al., “Bootstrap your own latent - A new approach to self-supervised learning,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 21271–21284.

[20] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, “Unsupervised learning of visual features by contrasting cluster assignments,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2020, pp. 9912–9924.

[21] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2117–2125.

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 8759–8768.

[23] A. v. d. Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” 2018, arXiv:1807.03748.

[24] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” Int. J. Comput. Vis., vol. 88, no. 2, pp. 303–338, 2010.

[26] M. Cordts et al., “The cityscapes dataset for semantic urban scene understanding,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 3213–3223.

[27] A. Gupta, P. Dollar, and R. Girshick, “LVIS: A dataset for large vocabulary instance segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5356–5364.

[28] J. R. Kiiliing, K. E. Van De Sande, T. Gevers, and A. W. Smuellers, “Selective search for object recognition,” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, 2013.

[29] W. Liu et al., “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 21–37.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.

[31] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 9627–9636.

[32] K. He, R. Girshick, and P. Dollar, “Rethinking imagenet pre-training,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 4918–4927.

[33] Y. Zhong, J. Wang, L. Wang, J. Peng, Y.-X. Wang, and L. Zhang, “DAP: Detection-aware pre-training with weak supervision,” 2021, arXiv:2103.16065.

[34] S. Gidaris, P. Singh, and N. Komodakis, “Unsupervised representation learning by predicting image rotations,” in Proc. Int. Conf. Learn. Representations, 2018.

[35] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in Proc. Int. Conf. Comput. Vis., 2015, pp. 1422–1430.

[36] M. Norouzi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in Proc. Eur. Conf. Comput. Vis., 2016, pp. 69–84.

[37] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2006, pp. 1735–1742.

[38] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” 2019, arXiv:1906.05849.
[39] O. J. Henaff et al., “Data-efficient image recognition with contrastive predictive coding,” 2019, arXiv:1905.09272.

[40] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. A. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with exemplar convolutional neural networks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[41] R. D. Hjelm et al., “Learning deep representations by mutual information estimation and maximization,” 2018, arXiv:1808.06670.

[42] P. Bachman, R. D. Hjelm, and W. Buchwalter, “Learning representations by maximizing mutual information across views,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2019, pp. 15 535–15 545.

[43] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, “What should not be contrastive in contrastive learning,” 2020, arXiv:2008.05659.

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, “Extracting and composing robust features with denoising autoencoders,” in Proc. Int. Conf. Mach. Learn., 2008, pp. 1096–1103.

[45] D. Pathak, P. Krizhevsky, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2536–2544.

[46] X. Wang, A. Jabri, and A. A. Efros, “Learning correspondence from the cycle-consistency of time,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 2566–2576.

[47] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, “Unsupervised deep tracking,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 1308–1317.

[48] P. O. Pinheiro, A. Almabairi, R. Y. Benmalek, F. Golembo, and A. Courville, “Unsupervised learning of dense visual representations,” 2020, arXiv:2011.05499.

[49] C. Yang, Z. Wu, B. Zhou, and S. Lin, “Instance localization for self-supervised detection pretraining,” 2021, arXiv:2102.08318.

[50] S. Liu, Z. Li, and J. Sun, “Self-emd: Self-supervised object detection without imageret,” 2020, arXiv:2011.13677.

[51] E. Xie et al., “Detoc: Unsupervised contrastive learning for object detection,” 2021, arXiv:2102.04803.

[52] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, “Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 16 684–16 693.

[53] X. Wang, R. Zhang, C. Shen, T. Kong, and L. Li, “Dense contrastive learning for self-supervised visual pre-training,” 2020, arXiv:2011.09157.

[54] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 9627–9636.

[55] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, “Randaugment: Practical automated data augmentation with a reduced search space,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops, 2020, pp. 792–793.

[56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[57] I. Misra and L. v. d. Maaten, “Self-supervised learning of pretext-invariant representations,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6707–6717.

[58] S. Gidaris, P. Komodakis, N. Komodakis, P. Perez, and M. Cord, “Learning representations by predicting bags of visual words,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 6928–6938.

[59] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[60] P. Goyal, D. Mahajan, A. Gupta, and I. Misra, “Scaling and benchmarking self-supervised visual representation learning,” in Proc. Int. Conf. Comput. Vis., 2019, pp. 6391–6400.

[61] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2881–2890.

[62] Y. Yuan, L. Huang, J. Guo, C. Zhang, X. Chen, and J. Wang, “OCNet: Object context for semantic segmentation,” Int. J. Comput. Vis., vol. 129, no. 8, pp. 1–24, 2021.

[63] C. Szegedy et al., “Going deeper with convolutions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1–9.

[64] M. Contributors, “MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark,” 2020. [Online]. Available: https://github.com/open-mmlab/mmsegmentation

Jian Ding received the BS degree in aircraft design and engineering from Northwestern Polytechnical University, Xian, China, in 2017. He is currently working toward the master degree in the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University. His research interests include object detection, instance segmentation, and remote sensing.

Enze Xie received the BS degree from the Nanjing University of Aeronautics and Astronautics (NUAA), China, in 2016, and the MS degree from Tongji University, China, in 2019. He is currently working toward the PhD degree in the Department of Computer Science, University of Hong Kong(HKU). His main research interests include object detection and instance segmentation.

Chenchen Jiang received the BS degree in software engineering from XiDian University, in 2016, and the master’s degree from the School of Data and Computer Science, Sun Yat-Sen University, in 2019. She is currently a researcher with Huawei Noah Ark Lab. Her research interests include machine Learning, object detection, AutoML, and nonparametric Statistics. He has published more than 20 papers in Top AI conferences: NeurIPS, CVPR, ICCV, AAAI and some statistics journals, e.g., Computational Statistics & Data Analysis, Statistical Computing.

Zhenguo Li received the bachelor’s and master’s degrees in mathematics from Peking University, in 2002 and 2005, respectively, and the doctorate degree from the School of Information Engineering, Chinese University of Hong Kong, in 2008. He is the director of AI Theory Lab of Noah’s Ark Laboratory. Before joining Huawei, he did post-doctoral research with the Department of Electrical Engineering, Columbia University. His research interests include machine learning and artificial intelligence. His main research results have been published in top international conferences such as NeurIPS, ICML, CVPR, ICCV, VLDB, and ICDE.
Ping Luo received the PhD degree in information engineering from the Chinese University of Hong Kong (CUHK), in 2014, supervised by Prof. Xiaou Tang and Prof. Xiaogang Wang. He is an assistant professor with the Department of Computer Science, University of Hong Kong (HKU). He was a postdoctoral fellow in CUHK from 2014 to 2016. He joined SenseTime Research as a principal research scientist from 2017 to 2018. His research interests include machine learning and computer vision. He has published more than 100 peer-reviewed articles in top-tier conferences and journals such as IEEE Transactions on Pattern Analysis and Machine Intelligence, International Journal of Computer Vision, ICML, ICLR, CVPR, and NIPS. His work has high impact with 13,000 citations according to Google Scholar. He has won a number of competitions and awards such as the first runner up in 2014 ImageNet ILSVRC Challenge, the first place in 2017 DAVIS Challenge on Video Object Segmentation, Gold medal in 2017 Youtube 8M Video Classification Challenge, the first place in 2018 Drivable Area Segmentation Challenge for Autonomous Driving, 2011 HK PhD Fellow Award, and 2013 Microsoft Research Fellow Award (ten PhDs in Asia). He is named one of the young innovators under 35 by MIT Technology Review (TR35) Asia Pacific.

Gui-Song Xia received the PhD degree in image processing and computer vision from CNRS LTCI, Télécom ParisTech, Paris, France, in 2011. From 2011 to 2012, he has been a post-doctoral researcher with the Centre de Recherche en Mathématiques de la Decision, CNRS, Paris-Dauphine University, Paris, for one and a half years. He is currently working as a full professor in computer vision and photogrammetry with Wuhan University. He has also been working as visiting scholar with DMA, École Normale Supérieure (ENS-Paris) for two months in 2018. He is also a guest professor with the Future Lab AI4EO, Technical University of Munich (TUM). His current research interests include mathematical modeling of images and videos, structure from motion, perceptual grouping, and remote sensing image understanding. He serves on the editorial boards of several journals, including the ISPRS Journal of Photogrammetry and Remote Sensing, Pattern Recognition, Signal Processing: Image Communications, EURASIP Journal on Image & Video Processing, Journal of Remote Sensing, and Frontiers in Computer Science: Computer Vision.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.