This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Dense Contrastive Learning for Self-Supervised Visual Pre-Training

Xinlong Wang[1], Rufeng Zhang[2], Chunhua Shen[1]*, Tao Kong[3], Lei Li[3]
[1]The University of Adelaide, Australia [2]Tongji University, China [3]ByteDance AI Lab

Abstract

To date, most existing self-supervised learning methods are designed and optimized for image classification. These pre-trained models can be sub-optimal for dense prediction tasks due to the discrepancy between image-level prediction and pixel-level prediction. To fill this gap, we aim to design an effective, dense self-supervised learning method that directly works at the level of pixels (or local features) by taking into account the correspondence between local features. We present dense contrastive learning (DenseCL), which implements self-supervised learning by optimizing a pairwise contrastive (dis)similarity loss at the pixel level between two views of input images.

Compared to the baseline method MoCo-v2, our method introduces negligible computation overhead (only <1% slower), but demonstrates consistently superior performance when transferring to downstream dense prediction tasks including object detection, semantic segmentation and instance segmentation; and outperforms the state-of-the-art methods by a large margin. Specifically, over the strong MoCo-v2 baseline, our method achieves significant improvements of 2.0% AP on PASCAL VOC object detection, 1.1% AP on COCO object detection, 0.9% AP on COCO instance segmentation, 3.0% mIoU on PASCAL VOC semantic segmentation and 1.8% mIoU on Cityscapes semantic segmentation.

Code and models are available at: `https://git.io/DenseCL`

1. Introduction

Pre-training has become a well-established paradigm in many computer vision tasks. In a typical pre-training paradigm, models are first pre-trained on large-scale datasets and then fine-tuned on target tasks with less training data. Specifically, the supervised ImageNet pre-training has been dominant for years, where the models are pre-trained to solve image classification and transferred to downstream tasks. However, there is a gap between image classification pre-training and target dense prediction tasks, such as object detection [9, 25] and semantic segmentation [5]. The former focuses on assigning a category to an input image, while the latter needs to perform dense classification or regression over the whole image. For example, semantic segmentation aims to assign a category for each pixel, and object detection aims to predict the categories and bounding boxes for all object instances of interest. A straightforward solution would be to pre-train on dense prediction tasks directly. However, these tasks' annotation is notoriously time-consuming compared to the image-level labeling, making it hard to collect data at a massive scale to pre-train a universal feature representation.

Recently, unsupervised visual pre-training has attracted much research attention, which aims to learn a proper visual representation from a large set of unlabeled images. A few methods [17, 2, 3, 14] show the effectiveness in downstream tasks, which achieve comparable or better results compared to supervised ImageNet pre-training. However, the gap between image classification pre-training and target

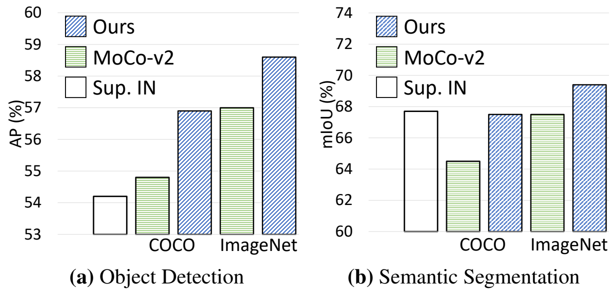


Figure 1 – Comparisons of pre-trained models by fine-tuning on object detection and semantic segmentation datasets. 'Sup. IN' denotes the supervised pre-training on ImageNet. 'COCO' and 'ImageNet' indicate the pre-training models trained on COCO and ImageNet respectively. (a): The object detection results of a Faster R-CNN detector fine-tuned on VOC `trainval07+12` for 24k iterations and evaluated on VOC `test2007`; (b): The semantic segmentation results of an FCN model fine-tuned on VOC `train_aug2012` for 20k iterations and evaluated on `val2012`. The results are averaged over 5 independent trials.

*Corresponding author.

dense prediction tasks still exists. First, almost all recent self-supervised learning methods formulate the learning as image-level prediction using global features. They all can be thought of as classifying each image into its own version, *i.e.*, instance discrimination [41]. Moreover, existing approaches are usually evaluated and optimized on the image classification benchmark. Nevertheless, better image classification does not guarantee more accurate object detection, as shown in [18]. Thus, self-supervised learning that is customized for dense prediction tasks is on demand. As for unsupervised pre-training, dense annotation is no longer needed. A clear approach would be pre-training as a dense prediction task *directly*, thus removing the gap between pre-training and target dense prediction tasks.

Inspired by the supervised dense prediction tasks, *e.g.*, semantic segmentation, which performs dense per-pixel classification, we propose dense contrastive learning (DenseCL) for self-supervised visual pre-training. DenseCL views the self-supervised learning task as a dense pairwise contrastive learning rather than the global image classification. First, we introduce a dense projection head that takes the features from backbone networks as input and generates dense feature vectors. Our method naturally preserves the spatial information and constructs a dense output format, compared to the existing global projection head that applies a global pooling to the backbone features and outputs a single, global feature vector for each image. Second, we define the positive sample of each local feature vector by extracting the correspondence across views. To construct an unsupervised objective function, we further design a dense contrastive loss, which extends the conventional InfoNCE loss [29] to a dense paradigm. With the above approaches, we perform contrastive learning densely using a fully convolutional network (FCN) [26], similar to target dense prediction tasks.

Our main contributions are thus summarized as follows.

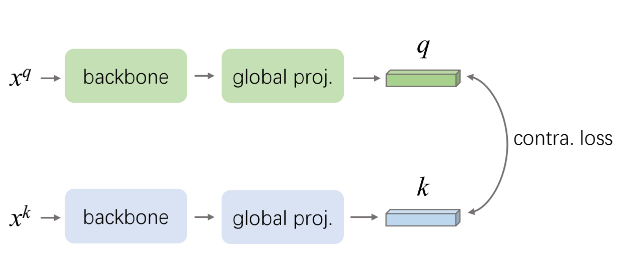
- We propose a new contrastive learning paradigm, *i.e.*, dense contrastive learning, which performs dense pairwise contrastive learning at the level of pixels (or local features).
- With the proposed dense contrastive learning, we design a simple and effective self-supervised learning method tailored for dense prediction tasks, termed DenseCL, which fills the gap between self-supervised pre-training and dense prediction tasks.
- DenseCL significantly outperforms the state-of-the-art MoCo-v2 [3] when transferring the pre-trained model to downstream dense prediction tasks, including object detection (+2.0% AP), instance segmentation (+0.9% AP) and semantic segmentation (+3.0% mIoU), and far surpasses the supervised ImageNet pre-training.

1.1. Related Work

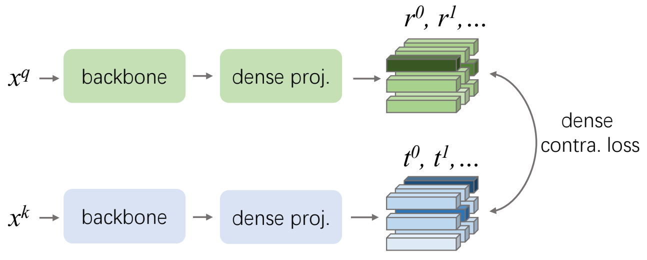
Self-supervised pre-training. Generally speaking, the success of self-supervised learning [41, 17, 42, 47, 16, 14] can be attributed to two important aspects namely *contrastive learning*, and *pretext tasks*. The objective functions used to train visual representations in many methods are either reconstruction-based loss functions [7, 30, 12], or contrastive loss that measures the co-occurrence of multiple views [38]. Contrastive learning, holds the key to most state-of-the-art methods [41, 17, 2, 42], in which the positive pair is usually formed with two augmented views of the same image (or other visual patterns), while negative ones are formed with different images.

A wide range of pretext tasks have been explored to learn a good representation. These examples include colorization [46], context autoencoders [7], inpainting [30], spatial jigsaw puzzles [28] and discriminate orientation [11]. These methods achieved very limited success in computer vision. The breakthrough approach is SimCLR [2], which follows an instance discrimination pretext task, similar to [41], where the features of each instance are pulled away from those of all other instances in the training set. Invariances are encoded from low-level image transformations such as cropping, scaling, and color jittering. Contrastive learning and pretext tasks are often combined to form a representation learning framework. DenseCL belongs to the self-supervised pre-training paradigm, and we naturally make the framework friendly for dense prediction tasks such as semantic segmentation and object detection.

Pre-training for dense prediction tasks. Pre-training has enabled surprising results on many dense prediction tasks, including object detection [34, 32] and semantic segmentation [26]. These models are usually fine-tuned from ImageNet pre-trained model, which is designed for image-level recognition tasks. Some previous studies have shown the gap between ImageNet pre-training and dense prediction tasks in the context of network architecture [24, 22, 37, 36]. YOLO9000 [33] proposes to joint train the object detector on both classification and detection data. He et al. [18] demonstrate that even we pre-train on extremely larger classification dataset (*e.g.*, Instagram [27], which is 3000$\times$ larger than ImageNet), the transfer improvements on object detection are relatively small. Recent works [23, 48] show that pre-trained models utilizing object detection data and annotations (*e.g.* MS COCO [25]) could achieve on par performance on object detection and semantic segmentation compared with ImageNet pre-trained model. While the supervised pre-training for dense prediction tasks has been explored before DenseCL, there are few works on designing an unsupervised paradigm for dense prediction tasks. Concurrent and independent works [31, 1] also find that contrastive learning at the level of local features matters. One of



(a) Global Contrastive Learning



(b) Dense Contrastive Learning

Figure 2 – Conceptual illustration of two contrastive learning paradigms for representation learning. We use a pair of query and key for simpler illustration. The backbone can be any convolutional neural network. (a): The contrastive loss is computed between the single feature vectors outputted by the global projection head, at the level of global feature; (b): The dense contrastive loss is computed between the dense feature vectors outputted by the dense projection head, at the level of local feature. For both paradigms, the two branches can be the same encoder or different ones, *e.g*., an encoder and its momentum-updated one.

the main differences is that they construct the positive pairs according to the geometric transformation, which brings the following issues. 1) Inflexible data augmentation. Need careful design for each kind of data augmentation to maintain the dense matching. 2) Limited application scenarios. It would fail when the geometric transformation between two views are not available. For example, two images are sampled from a video clip as the positive pair, which is the case of learning representation from video stream. By contrast, our method is totally decoupled from the data preprocessing, thus enabling fast and flexible training while being agnostic about what kind of augmentation is used and how the images are sampled.

Visual correspondence. The visual correspondence problem is to compute the pairs of pixels from two images that result from the same scene [43], and it is crucial for many applications, including optical flow [8], structure-from-motion [35], visual SLAM [20], 3D reconstruction [10] *etc*. Visual correspondence could be formulated as the problem of learning feature similarity between matched patches or points. Recently, a variety of convolutional neural network based approaches are proposed to measure the similarity between patches across images, including both supervised [4, 21] and unsupervised ones [45, 15]. Previous works usually leverage explicit supervision to learn the correspondence for a specific application. DenseCL learns general representations that could be shared among multiple dense prediction tasks.

2. Method

2.1. Background

For self-supervised representation learning, the breakthrough approaches are MoCo-v1/v2 [17, 3] and SimCLR [2], which both employ contrastive unsupervised learning to learn good representations from unlabeled data. We briefly introduce the state-of-the-art self-supervised learning framework by abstracting a common paradigm.

Pipeline. Given an unlabeled dataset, an instance discrimination [41] pretext task is followed where the features of each image in the training set are pulled away from those of other images. For each image, random 'views' are generated by random data augmentation. Each view is fed into an encoder for extracting features that encode and represent the whole view. There are two core components in an encoder, *i.e*., the backbone network and the projection head. The projection head attaches to the backbone network. The backbone is the model to be transferred after pre-training, while the projection head will be thrown away once the pre-training is completed. For a pair of views, they can be encoded by the same encoder [2], or separately by an encoder and its momentum-updated one [17]. The encoder is trained by optimizing a pairwise contrastive (dis)similarity loss, as revisited below. The overall pipeline is illustrated in Figure 2a.

Loss function. Following the principle of MoCo [17], the contrastive learning can be considered as a dictionary look-up task. For each encoded query q, there is a set of encoded keys $\{k_0, k_1, ...\}$, among which a single positive key k_+ matches query q. The encoded query and keys are generated from different views. For an encoded query q, its positive key k_+ encode different views of the same image, while the negative keys encode the views of different images. A contrastive loss function InfoNCE [29] is employed to pull q close to k_+ while pushing it away from other negative keys:

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\exp(q \cdot k_+) + \sum_{k_-} \exp(q \cdot k_- / \tau)}, \tag{1}$$

where τ denotes a temperature hyper-parameter as in [41].

2.2. DenseCL Pipeline

We propose a new self-supervised learning framework tailored for dense prediction tasks, termed DenseCL. DenseCL extends and generalizes the existing framework to a dense paradigm. Compared to the existing paradigm revisited in 2.1, the core differences lie in the encoder and loss function. Given an input view, the dense feature maps are extracted by the backbone network, *e.g.*, ResNet [19] or any other convolutional neural network, and forwarded to the following projection head. The projection head consists of two sub-heads in parallel, which are global projection head and dense projection head respectively. The global projection head can be instantiated as any of the existing projection heads such as the ones in [17, 2, 3], which takes the dense feature maps as input and generates a global feature vector for each view. For example, the projection head in [3] consists of a global pooling layer and an MLP which contains two fully connected layers with a ReLU layer between them. In contrast, the dense projection head takes the same input but outputs dense feature vectors.

Specifically, the global pooling layer is removed and the MLP is replaced by the identical 1×1 convolution layers [26]. In fact, the dense projection head has the same number of parameters as the global projection head. The backbone and two parallel projection heads are end-to-end trained by optimizing a joint pairwise contrastive (dis)similarity loss at the levels of both global features and local features.

2.3. Dense Contrastive Learning

We perform dense contrastive learning by extending the original contrastive loss function to a dense paradigm. We define a set of encoded keys $\{t_0, t_1, ...\}$ for each encoded query r. However, here each query no longer represents the whole view, but encodes a local part of a view. Specifically, it corresponds to one of the $S_h \times S_w$ feature vectors generated by the dense projection head, where S_h and S_w denote the spatial size of the generated dense feature maps. Note that S_h and S_w can be different, but we use $S_h = S_w = S$ for simpler illustration. Each negative key t_- is the pooled feature vector of a view from a different image. The positive key t_+ is assigned according to the extracted correspondence across views, which is one of the S^2 feature vectors from another view of the same image. For now, let us assume that we can easily find the positive key t_+. A discussion is deferred to the next section. The dense contrastive loss is defined as:

$$\mathcal{L}_r = \frac{1}{S^2}\sum_s -\log\frac{\exp(r^s{\cdot}t_+^s/\tau)}{\exp(r^s{\cdot}t_+^s)+\sum_{t_-^s}\exp(r^s{\cdot}t_-^s/\tau)}, \quad (2)$$

where r^s denotes the s^{th} out of S^2 encoded queries.

Overall, the total loss for our DenseCL can be formulated as:

$$\mathcal{L} = (1-\lambda)\mathcal{L}_q + \lambda\mathcal{L}_r, \quad (3)$$

where λ acts as the weight to balance the two terms. λ is set to 0.5 which is validated by experiments in Section 3.3.

2.4. Dense Correspondence across Views

We extract the dense correspondence between the two views of the same input image. For each view, the backbone network extracts feature maps $\mathbf{F} \in \mathbb{R}^{H\times W\times K}$, from which the dense projection head generates dense feature vectors $\mathbf{\Theta} \in \mathbb{R}^{S_h\times S_w\times E}$. Note that S_h and S_w can be different, but we use $S_h = S_w = S$ for simpler illustration. The correspondence is built between the dense feature vectors from the two views, *i.e.*, $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$. We match $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ using the backbone feature maps $\mathbf{F}_1$ and $\mathbf{F}_2$. The $\mathbf{F}_1$ and $\mathbf{F}_2$ are first downsampled to have the spatial shape of $S\times S$ by an adaptive average pooling, and then used to calculate the cosine similarity matrix $\mathbf{\Delta} \in \mathbb{R}^{S^2\times S^2}$. The matching rule is that each feature vector in a view is matched to the most similar feature vector in another view. Specifically, for all the S^2 feature vectors of $\mathbf{\Theta}_1$, the correspondence with $\mathbf{\Theta}_2$ is obtained by applying an `argmax` operation to the similarity matrix $\mathbf{\Delta}$ along the last dimension. The matching process can be formulated as:

$$c_i = \arg\max_j sim(\boldsymbol{f}_i, \boldsymbol{f}'_j), \quad (4)$$

where $\boldsymbol{f}_i$ is the i^{th} feature vector of backbone feature maps $\mathbf{F}_1$, and $\boldsymbol{f}'_j$ is the j^{th} of $\mathbf{F}_2$. $sim(\boldsymbol{u}, \boldsymbol{v})$ denotes the cosine similarity, calculated by the dot product between ℓ_2 normalized $\boldsymbol{u}$ and $\boldsymbol{v}$, *i.e.*, $sim(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top\boldsymbol{v}/\|\boldsymbol{u}\|\|\boldsymbol{v}\|$. The obtained c_i denotes the i^{th} out of S^2 matching from $\mathbf{\Theta}_1$ to $\mathbf{\Theta}_2$, which means that i^{th} feature vector of $\mathbf{\Theta}_1$ matches ${c_i}^{\text{th}}$ of $\mathbf{\Theta}_2$. The whole matching process could be efficiently implemented by matrix operations, thus introducing negligible latency overhead.

For the simplest case where $S = 1$, the matching degenerates into the one in global contrastive learning as the single correspondence naturally exists between two global feature vectors, which is the case introduced in Section 2.1.

According to the extracted dense correspondence, one can easily find the positive key t_+ for each query r during the dense contrastive learning introduced in Section 2.3.

Note that without the global contrastive learning term (*i.e.*, $\lambda = 1$), there is a chicken-and-egg issue that good features will not be learned if incorrect correspondence is extracted, and the correct correspondence will not be available if the features are not sufficiently good. In our default setting where $\lambda = 0.5$, no unstable training is observed. Besides setting $\lambda \in (0, 1)$ during the whole training, we introduce two more solutions which can also tackle this problem, detailed in Section 3.4.

3. Experiments

We adopt MoCo-v2 [3] as our baseline method, as which shows the state-of-the-art results and outperforms other methods by a large margin on downstream object detection task, as shown in Table 1. It indicates that it should serve as a very strong baseline on which we can demonstrate the effectiveness of our approach.

Technical details. We adapt most of the settings from [3]. A ResNet [19] is adopted as the backbone. The following global projection head and dense projection head both have a fixed-dimensional output. The former outputs a single 128-D feature vector for each input and the latter outputs dense 128-D feature vectors. Each ℓ_2 normalized feature vector represents a query or key. For both the global and dense contrastive learning, the dictionary size is set to 65536. The momentum is set to 0.999. Shuffling BN [17] is used during the training. The temperature τ in Equation (1) and Equation (2) is set to 0.2. The data augmentation pipeline consists of 224×224-pixel ramdom resized cropping, random color jittering, random gray-scale conversion, gaussian blurring and random horizontal flip.

3.1. Experimental Settings

Datasets. The pre-training experiments are conducted on two large-scale datasets: MS COCO [25] and ImageNet [6]. Only the training sets are used during the pre-training, which are ~118k and ~1.28 million images respectively. COCO and ImageNet represent two kinds of image data. The former is more natural and real-world, containing diverse scenes in the wild. It is a widely used and challenging dataset for object-level and pixel-level recognition tasks, such as object detection and instance segmentation. While the latter is heavily curated, carefully constructed for image-level recognition. A clear and quantitative comparison is the number of objects of interest. For example, COCO has a total of 123k images and 896k labeled objects, an average of 7.3 objects per image, which is far more than the ImageNet DET dataset's 1.1 objects per image.

Pre-training setup. For ImageNet pre-training, we closely follow MoCo-v2 [3] and use the same training hyper-parameters. For COCO pre-training including both baseline and ours, we use an initial learning rate of 0.3 instead of the original 0.03, as the former shows better performance in MoCo-v2 baseline when pre-training on COCO. We adopt SGD as the optimizer and we set its weight decay and momentum to 0.0001 and 0.9. Each pre-training model is optimized on 8 GPUs with a cosine learning rate decay schedule and a mini-batch size of 256. We train for 800 epochs for COCO, which is a total ~368k iterations. For ImageNet, we train for 200 epochs, a total of 1 million iterations.

Evaluation protocol. We evaluate the pre-trained models by fine-tuning on the target dense prediction tasks end-to-end. Challenging and popular datasets are adopted to fine-tune mainstream algorithms for different target tasks, *i.e*. VOC object detection, COCO object detection, COCO instance segmentation, VOC semantic segmentation, and Cityscapes semantic segmentation. When evaluating on object detection, we follow the common protocol that fine-tuning a Faster R-CNN detector (C4-backbone) on the VOC `trainval07+12` set with standard 2x schedule in [40] and testing on the VOC `test2007` set. In addition, we evaluate object detection and instance segmentation by fine-tuning a Mask R-CNN detector (FPN-backbone) with on COCO `train2017` split (~118k images) with the standard $1\times$ schedule and evaluating on COCO 5k `val2017` split. For semantic segmentation, an FCN model [26] is fine-tuned on VOC `train_aug2012` set (10582 images) for 20k iterations and evaluated on `val2012` set. We also evaluate semantic segmentation on Cityscapes dataset by training an FCN model on `train_fine` set (2975 images) for 40k iterations and test on `val` set. Detailed settings are in the supplementary.

3.2. Main Results

PASCAL VOC object detection. In Table 1, we report the object detection result on PASCAL VOC and compare it with other state-of-the-art methods. When pre-trained on COCO, our DenseCL outperforms the MoCo-v2 baseline by 2% AP. When pre-trained on ImageNet, the MoCo-v2 baseline has already surpassed other state-of-the-art self-supervised learning methods. And DenseCL still yields 1.7% AP improvements, strongly demonstrating the effectiveness of our method. The gains are consistent over all three metrics. It should be noted that we achieve much larger improvements on more stringent AP_{75} compared to those on AP_{50}, which indicates DenseCL largely helps improve the localization accuracy. Compared to the supervised ImageNet pre-training, we achieve the significant 4.5% AP gains.

COCO object detection and segmentation. The object detection and instance segmentation results on COCO are reported in Table 2. For object detection, DenseCL outperforms MoCo-v2 by 1.1% AP and 0.5% AP when pre-trained on COCO and ImageNet respectively. The gains are 0.9% AP and 0.3% AP for instance segmentation. Note that fine-tuning on COCO with a COCO pre-trained model is not a typical scenario. But the clear improvements still show the effectiveness.

PASCAL VOC semantic segmentation. We show the largest improvements on semantic segmentation. As shown in Table 3, DenseCL yields 3% mIoU gains when pre-training on COCO and fine-tuning an FCN on VOC semantic segmentation. The COCO pre-trained DenseCL achieves the same 67.5% mIoU as ImageNet pre-trained MoCo-v2. Note that compared to 200-epoch ImageNet pre-training, 800-epoch COCO pre-training only uses ~$1/10$ images and ~$1/3$ iterations. When pre-trained on ImageNet, DenseCL consistently brings 1.9% mIoU gains. It should be noted that the ImageNet pre-trained MoCo-v2 shows no transfer superiority compared with the supervised counterpart

pre-train	AP	AP_{50}	AP_{75}
random init.	32.8	59.0	31.6
super. IN	54.2	81.6	59.8
MoCo-v2 CC	54.7	81.0	60.6
DenseCL CC	56.7	81.7	63.0
SimCLR IN [2]	51.5	79.4	55.6
BYOL IN [14]	51.9	81.0	56.5
MoCo IN [17]	55.9	81.5	62.6
MoCo-v2 IN [3]	57.0	82.4	63.6
MoCo-v2 IN*	57.0	82.2	63.4
DenseCL IN	58.7	82.8	65.2

Table 1 – Object detection fine-tuned on PASCAL VOC. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet respectively. The models pre-trained on the same dataset are with the same training epochs, *i.e.*, 800 epochs for COCO and 200 epochs for ImageNet. '*' means re-implementation. The results of other methods are either from their papers or third-party implementation. All the detectors are trained on `trainval07+12` for 24k iterations and evaluated on `test2007`. The metrics include the VOC metric AP_{50} (*i.e.*, IoU threshold is 50%) and COCO-style AP and AP_{75}. The results are averaged over 5 independent trials.

pre-train	AP^{b}	AP^{b}_{50}	AP^{b}_{75}	AP^{m}	AP^{m}_{50}	AP^{m}_{75}
random init.	32.8	50.9	35.3	29.9	47.9	32.0
super. IN	39.7	59.5	43.3	35.9	56.6	38.6
MoCo-v2 CC	38.5	58.1	42.1	34.8	55.3	37.3
DenseCL CC	39.6	59.3	43.3	35.7	56.5	38.4
SimCLR IN	38.5	58.0	42.0	34.8	55.2	37.2
BYOL IN	38.4	57.9	41.9	34.9	55.3	37.5
MoCo-v2 IN	39.8	59.8	43.6	36.1	56.9	38.7
DenseCL IN	40.3	59.9	44.3	36.4	57.0	39.2

Table 2 – Object detection and instance segmentation fine-tuned on COCO. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet respectively. All the detectors are trained on `train2017` with default $1\times$ schedule and evaluated on `val2017`. The metrics include bounding box AP (AP^{b}) and mask AP (AP^{m}).

(67.5% vs. 67.7% mIoU). But DenseCL outperforms the supervised pre-training by a large margin, *i.e.*, 1.7% mIoU.

Cityscapes semantic segmentation. Cityscapes is a benchmark largely different from the above VOC and COCO. It focuses on urban street scenes. Nevertheless, in Table 3, we observe the same performance boost with DenseCL. Even the COCO pre-trained DenseCL can surpass the supervised ImageNet pre-trained model by 1.9% mIoU.

3.3. Ablation Study

We conduct extensive ablation experiments to show how each component contributes to DenseCL. We report ablation studies by pre-training on COCO and fine-tuning on VOC0712 object detection, as introduced in Section 3.1. All the detection results are averaged over 5 independent trials. We also provide results of VOC2007 SVM Classification, following [13, 44] which train linear SVMs on the VOC `train2007` split using the features extracted from the frozen backbone and evaluate on the `test2007` split.

pre-train	mIoU
random init.	40.7
super. IN	67.7
MoCo-v2 CC	64.5
DenseCL CC	67.5
SimCLR IN	64.3
BYOL IN	63.3
MoCo-v2 IN	67.5
DenseCL IN	69.4

(a) PASCAL VOC

pre-train	mIoU
random init.	63.5
super. IN	73.7
MoCo-v2 CC	73.8
DenseCL CC	75.6
SimCLR IN	73.1
BYOL IN	71.6
MoCo-v2 IN	74.5
DenseCL IN	75.7

(b) Cityscapes

Table 3 – Semantic segmentation on PASCAL VOC and Cityscapes. 'CC' and 'IN' indicate the pre-training models trained on COCO and ImageNet respectively. The metric is the commonly used mean IoU (mIoU). Results are averaged over 5 independent trials.

Loss weight λ. The hyper-parameter λ in Equation (3) serves as the weight to balance the two contrastive loss terms, *i.e.*, the global term and the dense term. We report the results of different λ in Table 4. It shows a trend that the detection performance improves when we increase the λ. For the baseline method, *i.e.*, $\lambda = 0$, the result is 54.7% AP. The AP is 56.2% when $\lambda = 0.3$, which improves the baseline by 1.5% AP. Increasing λ from 0.3 to 0.5 brings another 0.5% AP gains. Although further increasing it to 0.7 still gives minor improvements (0.1% AP) on detection performance, the classification result drops from 82.9% to 81.0%. Considering the trade-off, we use $\lambda = 0.5$ as our default setting in other experiments. It should be noted that when $\lambda = 0.9$, compared to the MoCo-v2 baseline, the classification performance rapidly drops (-4.8% mAP) while the detection performance improves for 0.8% AP. It is in accordance with our intention that DenseCL is specifically designed for dense prediction tasks.

	Detection			Classification
λ	AP	AP_{50}	AP_{75}	mAP
0.0	54.7	81.0	60.6	82.6
0.1	55.2	81.4	61.4	82.9
0.3	56.2	81.5	62.6	83.3
0.5	56.7	81.7	63.0	82.9
0.7	56.8	81.9	63.1	81.0
0.9	55.5	80.9	61.3	77.8
1.0*	53.5	79.5	58.8	68.9

Table 4 – Ablation study of weight λ. $\lambda = 0$ is the MoCo-v2 baseline. $\lambda = 0.5$ shows the best trade-off between detection and classification. '*' indicates training with warm-up, as discussed in Section 3.4.

strategy	Detection AP	Detection AP_{50}	Detection AP_{75}	Classification mAP
random	56.0	81.3	62.0	81.7
max-sim $\mathbf{\Theta}$	56.0	81.5	62.1	81.8
max-sim $\mathbf{F}$	56.7	81.7	63.0	82.9

Table 5 – Ablation study of matching strategy. To extract the dense correspondence according to the backbone features $\mathbf{F}_1$ and $\mathbf{F}_2$ shows the best results.

Matching strategy. In Table 5, we compare three different matching strategies used to extract correspondence across views. 1) 'random': the dense feature vectors from two views are randomly matched; 2) 'max-sim $\mathbf{\Theta}$': the dense correspondence is extracted using the dense feature vectors $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$ generated by the dense projection head; (3) 'max-sim $\mathbf{F}$': the dense correspondence is extracted according to the backbone features $\mathbf{F}_1$ and $\mathbf{F}_2$, as in Equation 4. The random matching strategy can also achieve 1.3% AP improvements compared to MoCo-v2, meanwhile the classification performance drops by 0.9% mAP. It may be because 1) the dense output format itself helps, and 2) part of the random matches are somewhat correct. Matching by the outputs of dense projection head, *i.e.*, $\mathbf{\Theta}_1$ and $\mathbf{\Theta}_2$, brings no clear improvement. The best results are obtained by extracting the dense correspondence according to the backbone features $\mathbf{F}_1$ and $\mathbf{F}_2$.

Grid size. In the default setting, the adopted ResNet backbone outputs features with stride 32. For a 224×224-pixel crop, the backbone features $\mathbf{F}$ has the spatial size of 7×7. We set the spatial size of the dense feature vectors $\mathbf{\Theta}$ to 7×7 by default, *i.e.*, $S = 7$. However, S can be flexibly adjusted and $\mathbf{F}$ will be pooled to the designated spatial size by an adaptive average pooling, as introduced in Section 2.4. We report the results of using different numbers of grid in Table 6. For $S = 1$, it is the same as the MoCo-v2 baseline except for two differences. 1) The parameters of dense projection head are independent with those of global projection head. 2) The dense contrastive learning maintains an independent dictionary. The results are similar to those of MoCo-v2 baseline. It indicates that the extra parameters and dictionary do not bring improvements. The performance improves as the grid size increases. We use grid size being 7 as the default setting, as the performance becomes stable when the S grows beyond 7.

Negative samples. We use the global average pooled features as negatives because it's conceptually simpler. Besides pooling, sampling is an alternative strategy. For keeping the same number of negatives, one can randomly sample a local feature from a different image. The COCO pre-trained model with sampling strategy achieves 56.7% AP on VOC detection, which is the same as the adopted pooling strategy.

grid size	Detection AP	Detection AP_{50}	Detection AP_{75}	Classification mAP
1	54.6	80.8	60.5	82.2
3	55.6	81.3	61.5	81.6
5	56.1	81.4	62.2	82.6
7	56.7	81.7	63.0	82.9
9	56.7	82.1	63.2	82.9

Table 6 – Ablation study of grid size S. The results increase as the S gets larger. We use grid size being 7 in other experiments, as the performance becomes stable when the S grows beyond 7.

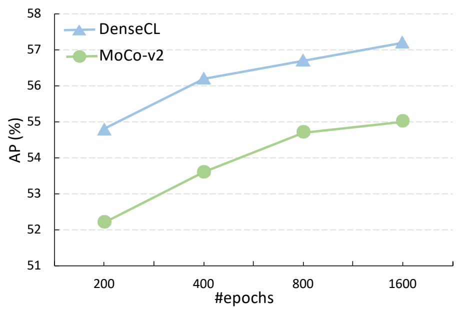


Figure 3 – Different pre-training schedules on COCO. For each pre-trained model, a Faster R-CNN detector is fine-tuned on VOC `trainval07+12` for 24k iterations and evaluated on `test2007`. The metric is the COCO-style AP. Results are averaged over 5 independent trials.

Training schedule. We show the results of using different training schedules in Table 7. The performance consistently improves as the training schedule gets longer, from 200 epochs to 1600 epochs. Note that the 1600-epoch COCO pre-trained DenseCL even surpasses the 200-epoch ImageNet pre-trained MoCO-v2 (57.2% AP vs. 57.0% AP). Compared to 200-epoch ImageNet pre-training, 1600-epoch COCO pre-training only uses $\sim 1/10$ images and $\sim 7/10$ iterations. In Figure 3, we further provide an intuitive comparison with the baseline as the training schedule gets longer. It shows that DenseCL consistently outperforms the MoCo-v2 by at least 2% AP.

#epochs	Detection AP	Detection AP_{50}	Detection AP_{75}	Classification mAP
200	54.8	80.5	60.7	77.6
400	56.2	81.5	62.3	81.3
800	56.7	81.7	63.0	82.9
1600	57.2	82.2	63.6	83.0

Table 7 – Ablation study of training schedule. The results consistently improve as the training schedule gets longer. Although 1600-epoch training schedule is 0.5% AP better, we use 800-epoch schedule in other experiments for faster training.

Pre-training time. In Table 8, we compare DenseCL with MoCo-v2 in terms of training time. DenseCL is only 1s

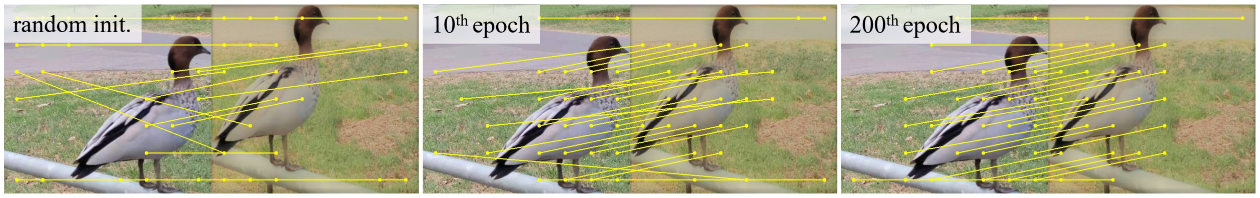


Figure 4 – Comparison of dense correspondence extracted from random initialization to well trained DenseCL. The correspondence is extracted between two views of the same image, using the ImageNet pre-trained model. All the matches are visualized without thresholding.

time/epoch	COCO	ImageNet
MoCo-v2	$1'45''$	$16'48''$
DenseCL	$1'46''$	$16'54''$

Table 8 – **Pre-training time comparison.** The training time per epoch is reported. We measure the results on the same 8-GPU machine. The training time overhead introduced by DenseCL is less than 1%.

and 6s slower per epoch when pre-trained on COCO and ImageNet respectively. The overhead is less than 1%. It strongly demonstrates the efficiency of our method.

3.4. Discussions on DenseCL

To further study how DenseCL works, in this section, we visualize the learned dense correspondence in DenseCL. The issue of chicken-and-egg during the training is also discussed.

Dense correspondence visualization. We visualize the dense correspondence from two aspects: comparison of the final correspondence extracted from different pre-training methods, *i.e.*, MoCo-v2 vs. DenseCL, and the comparison of different training status, *i.e.*, from the random initialization to well trained DenseCL. Given two views of the same image, we use the pre-trained backbone to extract the features $\mathbf{F}_1$ and $\mathbf{F}_2$. For each feature vector in $\mathbf{F}_1$, we find the corresponding feature vector in $\mathbf{F}_2$ which has the highest cosine similarity. The match is kept if the same match holds from $\mathbf{F}_2$ to $\mathbf{F}_1$. Each match is assigned an averaged similarity. In [39], we visualize the high-similarity matches (*i.e.*, similarity ≥ 0.9). DenseCL extracts many more high-similarity matches than its baseline. It is in accordance with our intention that the local features extracted from the two views of the same image should be similar.

Figure 4 shows how the correspondence changes over training time. The randomly initialized model extracts some random noisy matches. The matches get more accurate as the training time increases.

Chicken-and-egg issue. In our pilot experiments, we observe that the training loss does not converge if we set λ to 1.0, *i.e.*, removing the global contrastive learning, and only applying the dense contrastive learning. It may be because at the beginning of the training, the randomly initialized model is not able to generate correct correspondence across views. It is thus a chicken-and-egg issue that good features will not be learned if incorrect correspondence is extracted, and the correct correspondence will not be available if the features are not sufficiently good. As shown in Figure 4, most of the matches are incorrect with the random initialization. The core solution is to provide a guide when training starts, to break the deadlock. We introduce three different solutions to tackle this problem. 1) To initialize the model with the weights of a pre-trained model; 2) To set a warm-up period at the beginning during which the λ is set to 0; 3) To set $\lambda \in (0, 1)$ during the whole training. They all solve this issue well. The second one is reported in Table 4, with λ being changed from 0 to 1.0 after the first 10k iterations. We adopt the last one as the default setting for its simplicity.

4. Conclusion

In this work we have developed a simple and effective self-supervised learning framework DenseCL, which is designed and optimized for dense prediction tasks. A new contrastive learning paradigm is proposed to perform dense pairwise contrastive learning at the level of pixels (or local features). Our method largely closes the gap between self-supervised pre-training and dense prediction tasks, and shows significant improvements in a variety of tasks and datasets, including PASCAL VOC object detection, COCO object detection, COCO instance segmentation, PASCAL VOC semantic segmentation and Cityscapes semantic segmentation. We expect the proposed effective and efficient self-supervised pre-training techniques could be applied to larger-scale data to fully realize its potential, as well as hoping that DenseCL pre-trained models would completely replace the supervised pre-trained models in many of those dense prediction tasks in computer vision.

Acknowledgement and Declaration of Conflicting Interests The authors would like to thank Wanxuan Lu for her helpful discussion. CS and his employer received no financial support for the research, authorship, and/or publication of this article.

References

[1] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. Contrastive learning of global and local features for medical image segmentation with limited annotations. *arXiv preprint arXiv:2006.10511*, 2020. 2

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, 2020. 1, 2, 3, 4, 6

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv: Comp. Res. Repository*, 2020. 1, 2, 3, 4, 5, 6

[4] Christopher Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2414–2422, 2016. 3

[5] 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 *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3213–3223, 2016. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 248–255, 2009. 5

[7] Carl Doersch, Abhinav Gupta, and Alexei Efros. Unsupervised visual representation learning by context prediction. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1422–1430, 2015. 2

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2758–2766, 2015. 3

[9] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *Int. J. Comput. Vision*, 88(2):303–338, 2010. 1

[10] Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symp.*, pages 963–968, 2011. 3

[11] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. Int. Conf. Learn. Representations*, 2018. 2

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 2672–2680, 2014. 2

[13] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2019. 6

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv: Comp. Res. Repository*, 2020. 1, 2, 6

[15] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4370–4379, 2019. 3

[16] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *Proc. Advances in Neural Inf. Process. Syst.*, 33, 2020. 2

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020. 1, 2, 3, 4, 5, 6

[18] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 4918–4927, 2019. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016. 4, 5

[20] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for rgb-d cameras. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots & Systems*, pages 2100–2106, 2013. 3

[21] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. FCSS: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6560–6569, 2017. 3

[22] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. HyperNet: Towards accurate region proposal generation and joint object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 845–853, 2016. 2

[23] Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S. Davis. An analysis of pre-training on object detection. *arXiv: Comp. Res. Repository*, 2019. 2

[24] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. DetNet: Design backbone for object detection. In *Proc. Eur. Conf. Comp. Vis.*, pages 334–350, 2018. 2

[25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755, 2014. 1, 2, 5

[26] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015. 2, 4, 5

[27] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proc. Eur. Conf. Comp. Vis.*, pages 181–196, 2018. 2

[28] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. Eur. Conf. Comp. Vis.*, pages 69–84, 2016. 2

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv: Comp. Res. Repository*, 2018. 2, 3

[30] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2536–2544, 2016. 2

[31] Pedro Pinheiro, Amjad Almahairi, Ryan Y Benmaleck, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. 2020. 2

[32] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, real-time object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 779–788, 2016. 2

[33] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7263–7271, 2017. 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Inf. Process. Syst.*, pages 91–99, 2015. 2

[35] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4104–4113, 2016. 3

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5693–5703, 2019. 2

[37] Mingxing Tan, Ruoming Pang, and Quoc V Le. EfficientDet: Scalable and efficient object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10781–10790, 2020. 2

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

[39] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. *arXiv preprint arXiv:2011.09157*, 2020. 8

[40] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019. 5

[41] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018. 2, 3

[42] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas J Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3d point cloud understanding. *arXiv preprint arXiv:2007.10985*, 2020. 2

[43] Ramin Zabih and John Woodfill. Non-parametric local transforms for computing visual correspondence. In *Proc. Eur. Conf. Comp. Vis.*, pages 151–158, 1994. 3

[44] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2020. 6

[45] Chao Zhang, Chunhua Shen, and Tingzhi Shen. Unsupervised feature learning for dense correspondences across scenes. *Int. J. Comput. Vision*, 116(1):90–107, 2016. 3

[46] Richard Zhang, Phillip Isola, and Alexei Efros. Colorful image colorization. In *Proc. Eur. Conf. Comp. Vis.*, pages 649–666, 2016. 2

[47] Nanxuan Zhao, Zhirong Wu, Rynson Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 2

[48] Dongzhan Zhou, Xinchi Zhou, Hongwen Zhang, Shuai Yi, and Wanli Ouyang. Cheaper Pre-training Lunch: An efficient paradigm for object detection. *arXiv preprint arXiv:2004.12178*, 2020. 2