

# Coarse Is Better? A New Pipeline Towards Self-Supervised Learning with Uncurated Images

Ke Zhu, Yin-Yin He and Jianxin Wu

State Key Laboratory for Novel Software Technology

Nanjing University, Nanjing, China

{zhuk, heyy}@lamda.nju.edu.cn, wujx2001@nju.edu.cn

## Abstract

Most self-supervised learning (SSL) methods often work on curated datasets where the object-centric assumption holds. This assumption breaks down in uncurated images. Existing scene image SSL methods try to find the two views from original scene images that are well matched or dense, which is both complex and computationally heavy. This paper proposes a conceptually different pipeline: first find regions that are coarse objects (with adequate objectness), crop them out as pseudo object-centric images, then any SSL method can be directly applied as in a real object-centric dataset. That is, coarse crops benefits scene images SSL. A novel cropping strategy that produces coarse object box is proposed. The new pipeline and cropping strategy successfully learn quality features from uncurated datasets without ImageNet. Experiments show that our pipeline outperforms existing SSL methods (MoCo-v2, DenseCL and MAE) on classification, detection and segmentation tasks. We further conduct extensively ablations to verify that: 1) the pipeline do not rely on pretrained models; 2) the cropping strategy is better than existing object discovery methods; 3) our method is not sensitive to hyperparameters and data augmentations.

## 1. Introduction

Self-supervised learning (SSL) of visual representation has boosted the accuracy of various downstream tasks such as image classification [38] and object detection [37]. Recently, contrastive learning based SSL methods have become a popular paradigm. Representative methods in this family, such as MoCo [6], BYOL [15] and SwAV [4], are based on the so-called *object-centric* assumption: the images used for SSL are supposed to be object-centric, and hence two different *views* augmented from the same image will share similar visual semantics. This assumption holds true in object-centric (also called curated) datasets

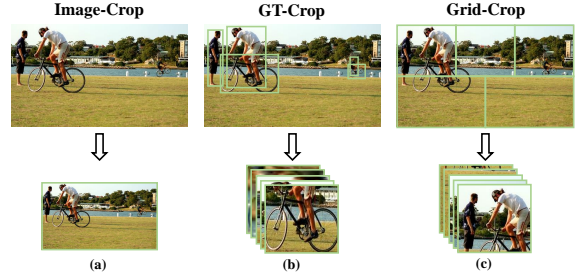


Figure 1. Three strategies for cropping pseudo object-centric images, using the entire image (a), groundtruth (b), and regular grid (c), respectively. This figure is best viewed in color.

like ImageNet [38], in which objects are usually centered and occupy large areas. But, for downstream tasks like detection and segmentation, it simply breaks down. The images in such scene (or multi-object, or uncurated) datasets, e.g., VOC [13] and COCO [27], often contain many (small) objects per image. Hence, it is a natural requirement that we need new methods to handle SSL with scene images. In this paper, we aim to design a SSL method that *works directly on scene images*, without *relying on object-centric datasets like ImageNet*. In SSL pretraining facing uncurated datasets, previous efforts are generally based on *dense comparison*, such as DenseCL [43], Self-EMD [28] and ReSim [46]. They [28, 47, 22, 1] introduce region-level or feature-level pretext tasks, with dedicated pipelines to formulate the loss functions. Another line of work [44, 48] (e.g., SoCo [44]) try to find pairs of precise boxes with similar semantics from images to apply the InfoNCE loss [35], but they involve heavy matching procedures [48] on top of box generation (e.g., selective search [40]). Besides, their downstream improvements are modest (e.g., considering SoCo pretrained on MS-COCO [27]).

One common characteristic of these prior arts is that they need *manually* matched pairs on dense feature level or image level, and they *generate the views directly from the raw input image*. The input image is uncurated, but the gener-

Table 1. Downstream tasks’ results using different cropping strategies for object-centric SSL pretraining on VOC2007. All models (ResNet50) were pretrained for 800 epochs on the trainval set using BYOL or MoCo-v2.  $AP_{50}^{bbox}$  evaluates object detection accuracy, and  $mAP$  and  $mAP^l$  are for multi-label classification with end-to-end finetuning and linear evaluation, respectively.

Method	Cropping	$AP_{50}^{bbox}$	$mAP$	$mAP^l$
BYOL	Image-Crop	63.3	63.0	39.8
	GT-Crop	66.9	69.2	33.1
	Grid-Crop	68.1	70.3	41.6
	Our-Crop	<b>69.5</b>	<b>70.7</b>	<b>42.5</b>
MoCo-v2	Image-Crop	61.8	62.2	26.9
	GT-Crop	62.3	66.2	28.7
	Grid-Crop	65.6	67.7	37.2
	Our-Crop	<b>67.2</b>	<b>69.6</b>	<b>44.9</b>

ated views are assumed to be good matches for each other and correspond to objects. This assumption is tremendously difficult to entertain. Besides, it is believed that the (unavailable) *GT-box/GT-mask* could lead to the best pretrained features (as shown in previous SSL method [22]). The usefulness of coarse boxes (which contain partial objects) to SSL pretraining, however, remains unexplored.

In this paper, we propose a completely different *scene image SSL paradigm without matching boxes*, and for the first time argue that *coarse object boxes are comparable to GT boxes* for SSL pretraining. Our method can be summarized as follows: we first crop a few (e.g., 5) boxes from each uncured image, *only requiring the cropped boxes to coarsely contain objects* (e.g., do contain object(s), but not necessarily a single or a full object); then, the cropped boxes are treated as *pseudo object-centric images*, which are directly fed to existing object-centric SSL methods (like MoCo-v2 or BYOL), and the two views are extracted from the same pseudo object-centric image for SSL learning.

Our approach is motivated by the following simple experiment. We consider 3 different cropping strategies as in Fig. 1. ‘Image-Crop’ treats the entire input image as the only crop. ‘GT-Crop’ utilizes the groundtruth bounding box annotations, and each bounding box becomes a crop. ‘Grid-Crop’ splits the input image into 2 rows, with each row being split into 3 or 2 crops with equal size. These crops form pseudo object-centric images *with dramatically different crop quality*, where ‘GT’ is perfect but ‘Grid’ is almost random in objectness. By pretraining an SSL model using these pseudo images to initialize downstream models, Table 1 shows their object detection and multi-label recognition accuracy on the VOC2007 dataset (more details in Sec. 3). The results demonstrate that all cropping strategies (after SSL) are beneficial to downstream tasks. More importantly, the almost random ‘Grid’ crop is even consistently better than the ‘GT’ strategy. The ‘Grid’ crops indeed contain objects, but the objects may be partial or multiple, i.e., only coarsely contain objects. Hence, we hypothesize that

Table 2. Empirical analysis of cropping strategies. ‘Objectness’ is the average ratio of object pixels within each crop. We selected a best padding ratio for ‘GTPad-Crop’ and generated ‘Poor-Crop’ objectness around 20%. Models were pretrained with BYOL. We used the pixel-level groundtruth annotations only in calculating the objectness.

Cropping	$AP_{50}^{bbox}$	$mAP$	$mAP^l$	Objectness
GT-Crop	66.9	69.2	33.1	100.0%
GTPad-Crop	<b>69.7</b>	<b>72.3</b>	<b>40.2</b>	48.2%
Poor-Crop	60.4	62.2	21.4	20.1%
Image-Crop	63.3	63.0	39.8	36.4%
Grid-Crop	68.1	70.3	41.6	39.8%
Our-Crop	<b>69.5</b>	<b>70.7</b>	<b>42.5</b>	52.6%

we do not need precisely cropped images as views, nor do we need matched crops. Instead, *crops with coarse objects are what we want*. Treating these crops as pseudo object-centric images and then *generate two views from the same pseudo image* is both much simpler and more beneficial.

We have then calculated the objectness of the crops, and consider two new crops: ‘Poor-Crop’ (low objectness regions) and ‘GTPad-Crop’ (GT boxes with padding). As shown in Table 2 (cf. the visualization in Fig. 5), too low objectness (‘Poor-Crop’) is detrimental for SSL, while too high objectness (‘GT-Crop’) leads to suboptimal results compared with medium objectness (‘GTPad-Crop’ and ‘Our-Crop’). These results again support our hypothesis: *objectness should not be too high, and coarse crops are enough*. On the other hand, *a filtering algorithm is required to filter away crops with low objectness*. For example, roughly 40% of the crops in ‘Grid-Crop’ have lower than 20% objectness (see appendix for more details), which hinders the model from learning quality representation.

Consequently, we propose a novel crop generation and filtering method to obtain bounding boxes with adequate objectness, and demonstrate its superiority over existing unsupervised object discovery methods from both qualitative and quantitative results. The effectiveness of our new pipeline and the cropping strategy are extensively verified on uncured datasets like MS-COCO and VOC for detection, segmentation and classification, and are further studied by abundant ablations. Our contributions can be summarized as:

1. We conduct an empirical study to argue that cropping benefits scene images SSL—just treat these local crops as pseudo object-centric images.
2. We verify that we do not need precisely matched boxes for scene SSL pretraining, and propose a novel box generation and filtering strategy to obtain crops with coarse objectness (i.e., ‘Our-Crop’).
3. Feeding the cropped pseudo images to existing SSL methods (e.g., BYOL, DenseCL or MAE), we achieve

consistent improvements on VOC and MS-COCO for recognition, detection and segmentation. Note that we do *not* rely on the ImageNet pretraining data.

## 2. Related Work

**SSL on object-centric images.** Self-supervised learning has become a powerful tool to learn good representations from unlabeled data [7, 8]. In particular, contrastive SSL methods [21, 29, 25, 30] become the most popular and have largely boosted the performance of various downstream tasks. Previous works formulated the InfoNCE [35] loss using contrastive predictive coding, and InfoMin [39] argued that the mutual information of two views should be reduced as long as the task-relevant information are kept intact. Recent studies based on bootstrapping [15, 49], memory bank [6, 47, 1, 22] and online learning [4] tried to pull positive pairs closer while pushing negative pairs apart. The underline assumption of these methods is that two positive views generated from the same object-centric image should share similar content, which may cease to be true in uncurated datasets. [14] attempted to apply SSL in the wild, and [41] tried to improve SSL of scene images by introducing constrained multi-crop [4]. Still, a specialized method for uncurated scene images is of high importance.

**SSL on uncurated scene images.** Recent works [43, 52, 28, 48] started to focus on the usage of uncurated datasets like MS-COCO [27], aiming to get better representation for dense prediction tasks. DenseCL [43], MaskCo [52], Self-EMD [28] tried to form positive pairs with dense features. But, they need manually matched pairs from feature level or image level, which can be difficult to entertain, especially in scene images with multiple different objects.

Another SSL paradigm [48, 44, 22] resorted to external unsupervised object discovery methods such as selective search [40] to obtain precise object boxes in scene images. They tried to learn a good representation based on the global image and local object parts. These methods usually involved complex filtering algorithms like K-Nearest Neighbors [48], so as to select object bounding boxes as precisely as possible. They assumed that the more precise the object proposals, the better the SSL. In this paper, however, we call into question the validity of this assumption. Instead, we argue that bounding boxes that are only roughly objects (with adequate objectness) are enough, and we need to crop them as pseudo object-centric images. Then, we advocate that the two positive views need to be cropped within the same one such pseudo image, *not from two original input images*.

A recent work that is close to ours is [32], which try to obtain views from object-aware boxes. However, we clarify that Our-Crop and [32] is much different. First, [32] directly rely on existing object discovery methods (some even need groundtruth annotations), while our cropping strategy

is done in a fully unsupervised manner. More importantly, *quantitative analysis* of how different quality (objectness, cf. Table 1-2) of boxes affect pretraining are not told in their paper, while we found perfect GT are *not* possibly the best and argued *coarse* is enough. Technically, they are very different, too: they add extra augment like dilation, shifting to the BING boxes, adjust many augmentation parameters, and adopt different projection heads, while we *directly* fed our pseudo-images into SSL without *any* modification. In a word, our pipeline, cropping strategy and the analysis shares minimal similarity with that of [32].

## 3. Method

We first briefly introduce the formulation of two SSL methods for object-centric images, as they are both our baseline and component techniques. Then, we present the overall pipeline of the proposed method in detail. Finally, we present the key component of our pipeline: a novel cropping strategy to generate pseudo object-centric images.

### 3.1. MoCo-v2 and BYOL

Both MoCo-v2 and BYOL implicitly follow the object-centric assumption, but in practice we can feed any type of images (*e.g.*, uncurated ones) to them. MoCo-v2 [6] is based on the InfoNCE [35] loss:

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

where  $\tau$  is the temperature, and  $q$  is the encoded query that computes the loss over one positive key  $k_+$  and many negatives  $k_-$ . Note that  $q$  and  $k_+$  are *two views generated from the same image by different data augmentations*.

BYOL [15] devises an online network  $\theta$  and a slow moving averaged target network  $\xi$ , which was inspired by several earlier reinforcement learning methods [16]. The loss function of BYOL is designed to only pull positive views together:

$$\mathcal{L}_{\theta, \xi} = \left\| \frac{q_\theta}{\|q_\theta\|} - \frac{z'_\xi}{\|z'_\xi\|} \right\|_2^2 = 2 \left( 1 - \frac{\langle q_\theta, z'_\xi \rangle}{\|q_\theta\| \cdot \|z'_\xi\|} \right), \quad (2)$$

in which  $q_\theta$  and  $z'_\xi$  are feature representations for the online and the target network, respectively. Note that  $q_\theta$  and  $z'_\xi$  are also *two views generated from the same image*. Many recent work [28, 48, 52] have followed BYOL for its efficacy.

### 3.2. A new pipeline for SSL with scene images

It is easy to accept that two views generated out of the same object-centric image share enough semantic similarity. But, as suggested by previous work [43, 52, 48], two views generated from one non-object-centric image potentially have insufficient semantic overlap and will lead to



suboptimal accuracy in downstream tasks. Hence, we argue that instead of following the commonly used strategy to find semantically or visually matched object regions as the two views, it is better to *crop out regions that coarsely contain objects* (i.e., *with adequate objectness*) as *pseudo object-centric images*, then *generate two views from the same one such pseudo image*. The two views generated in this manner are naturally similar to each other, and will contain coarse object information, too.

The proposed pipeline is as follows:

1. Given a scene image dataset like COCO, use any object-centric SSL methods to learn a backbone model (the  $\phi(\cdot)$  shown in Fig. 2) by simply pretending these images as object-centric ones (though Table 11 proved that this backbone learning stage is *not* necessary).
2. For each uncurated image, use our proposed cropping strategy and this backbone model  $\phi(\cdot)$  to find a few (e.g., 5) crops that coarsely contain objects: filter away crops with low objectness and the remaining ones contain adequate objectness. These crops are treated as pseudo object-centric images.
3. Feed these pseudo object-centric images to any object-centric SSL method to train a randomly initialized backbone model from scratch (that is, we do *not* inherit the weights of  $\phi(\cdot)$  from stage 1) so as to finish unsupervised visual representation learning with uncurated images.

Our pipeline (and its feasibility) is motivated by the experiments in Table 1, in which we tried 3 cropping strategies and 2 SSL methods (BYOL and MoCo-v2) on VOC2007. Following [3], we set batch size to 256, and trained 800 epochs on VOC2007 trainval [13]. For the downstream tasks, we trained Faster R-CNN [37] for object detection and finetuned multi-label classification models.

As shown in Table 1, directly cropping two views from the uncurated images (‘Image-Crop’) works fairly poorly, since it has a big chance to generate two views containing different semantics. ‘GT-Crop’ is much better, showing the importance of cropping two views from objects locally. What is surprising is that cropping two views from one fixed-grid pseudo image (‘Grid-Crop’) achieves even higher mAP than ‘GT’! In fact, the five pseudo images of ‘Grid’ contain random contents and are at best coarse objects, which suggests that we do *not need precise object locations* in a cropping strategy. Then, it is natural to hypothesize that *constructing two views within one locally cropped image with coarse object(s) is the key*—we do not need precise object cropping, and it is both easier and better than finding matching regions as the views [43, 22, 48] either from the image-level or the feature-level.

Meanwhile, as Table 2 suggests, the pseudo images in ‘Grid’ are too coarse such that a large proportion of them have extremely low quality (about 40% of the crops in ‘Grid’ have lower than 20% objectness), and may contain no objects at all (e.g., the one contains only grass in Fig. 1). The diversity of the boxes is not guaranteed, either. These drawbacks necessitate a novel strategy that filters away cropped pseudo images with low objectness scores and to make the cropped pseudo images diverse (e.g., it is harmful if all 5 crops are around one same region).

### 3.3. Generating coarse object crops

The proposed cropping strategy (Fig. 2) differs from previous unsupervised object discovery methods [40, 9, 53], since these object discovery methods focus on cropping highly accurate objects (as later verified in Fig. 3), which may not suit SSL pretraining well (as indicated by Table 2 and compared in Table 13). Instead, we argue that crops that coarsely contain objects are enough. As Fig. 2 illustrates, we compute both the per-pixel objectness score and a set of predefined anchors, then filter the anchors with the objectness scores.

**Anchor generation.** We borrow the anchors from object detection [37, 19], since it *does not* need a real forward pass and is highly efficient. Besides, the diversity of the anchors are guaranteed, as shown by the detection literature. For an image  $I$  with height  $H$  and width  $W$ , a feature tensor  $\mathbf{x} \in \mathbb{R}^{d \times h \times w}$  would be obtained *if it were* to be sent to the backbone network, where  $d$ ,  $w$  and  $h$  are the dimensionality, height and width of the feature tensor, respectively. For each spatial location out of the  $h \times w$  in the feature tensor, we predefine 12 anchors with four different sizes (32, 64, 128, 256) and three aspect ratios (0.5, 1.0, 2.0). Then the spatial coordinates of these anchors are mapped from the feature space back into the image space according to the down-sampling ratio  $H/h$ . Thus, a total of  $12 \times h \times w$  anchor boxes are generated for an image.

**Objectness scores and crops.** The key idea to filter the generated boxes is to assign each box an objectness score. For simplicity, we generate a shared per-pixel score map, and the objectness score of a box is the average score of all pixels within it. Following SCDA [45], an image  $I \in \mathbb{R}^{H \times W}$  is first sent to a feature extractor  $\phi$  (the backbone network) to obtain the feature tensor  $\mathbf{x} \in \mathbb{R}^{d \times h \times w}$ :  $\mathbf{x} = \phi(I; \theta)$ , where  $\theta$  are the backbone’s parameters. Then, a sum operation is done on the feature dimension to obtain a corresponding score map  $A \in \mathbb{R}^{h \times w}$ :

$$A_{i,j} = \sum_d x_{i,j,d}, \quad (3)$$

in which  $i \in \{1, 2, \dots, h\}$  and  $j \in \{1, 2, \dots, w\}$ . As pointed out by SCDA [45], the score map  $A$  is indicative of the objectness at the corresponding position. We then linearly nor-

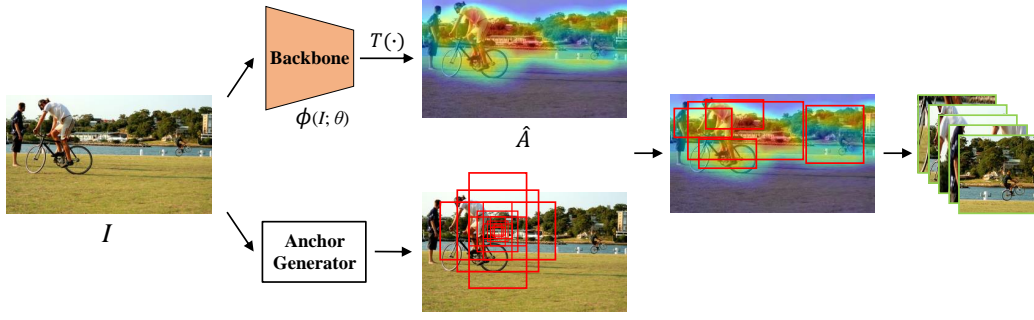


Figure 2. The proposed cropping strategy. In the top branch, an image  $I \in \mathbb{R}^{H \times W \times C}$  first passes through a self-supervised pretrained backbone  $\phi(\cdot)$  (e.g., ResNet-50) to obtain a feature tensor  $\mathbf{x} \in \mathbb{R}^{h \times w \times d}$ , which is then transformed by  $T(\cdot)$  (sum, normalize and resize) to an objectness score map  $\hat{A} \in \mathbb{R}^{H \times W}$  (color coded on top of the input image). Second, in the bottom branch, a large number of anchor boxes with different sizes and aspect ratios are generated. Third, anchors are filtered according to the score map  $\hat{A}$ . Finally, the top-5 anchors are our selected crops (pseudo object-centric images). Best viewed in color.

malize the score map  $A$  to the  $[0, 1]$  range and upsample it to the original image size by bilinear interpolation to get the final score map  $\hat{A} \in \mathbb{R}^{H \times W}$ .

Suppose we have generated  $N$  anchors in an image, namely,  $c_1, c_2, \dots, c_N$ . We calculate the mean score within each anchor by using the final score map  $\hat{A}$ . Specifically, for a given anchor  $c_k$ , the score  $s_k$  for it is:

$$s_k = \frac{1}{h' \times w'} \sum_{i=x_1}^{y_1} \sum_{j=y_1}^{y_2} \hat{A}_{i,j}, \quad (4)$$

where  $h'$  and  $w'$  are the height and width of the anchor  $c_k$ , and  $(x_1, y_1)$  and  $(x_2, y_2)$  are the top-left and bottom-right coordinates of the anchor  $c_k$ , respectively. The NMS [33] operation is then applied to merge heavily-overlapped anchors. Finally, anchors with the top-5 scores are cropped.

**The backbone and the final learning.** This box filtering process needs a feature extractor  $\phi$  to produce an objectness score map  $A$ . Take Table 1 for example, we use MoCo-v2 and uncured images (i.e., ‘Image-Crop’) to learn the backbone  $\phi$ . After all the pseudo object-centric images are cropped, we *randomly initialize* a model, and then perform SSL using any object-centric SSL method and these pseudo images. As shown in Table 1, ‘Our-Crop’ significantly outperforms other cropping strategies, showing the effectiveness of our cropping strategy.

## 4. Experiments

For almost all of our experiments, we choose MS-COCO [27] as the pretraining dataset and evaluate the effectiveness of our cropping strategy on various downstream benchmarks: MS-COCO detection and segmentation, VOC0712 [13] detection, CityScapes [11] segmentation and 7 small classification [42] datasets. We also explore the effectiveness of Our-Crop pretrained on truly object-centric dataset (e.g., ImageNet [38] only in Table 8).

### 4.1. Experimental settings

**Datasets.** MS-COCO is a large uncured dataset. We use 118k training images to pretrain our backbone with different self-supervised methods (MoCo-v2, BYOL, DenseCL and MAE). VOC2007 contains 5,011 trainval and 4,952 test images, while VOC2012 [13] has 11k trainval images. The Cityscapes dataset is a scene dataset containing 3475 images (2975 for training and 500 for validation). Since most previous [43, 41] SSL works utilize these datasets for evaluation, we evaluate the object detection and instance segmentation tasks on MS-COCO, object detection and multi-label classification tasks on VOC, and semantic segmentation on Cityscapes. Besides, we also validate our effectiveness on various single-label classification datasets, namely CUB200 [42], Flowers [34], Cars [24], Aircraft [31] and DTD [10]. On top of that, we explore ImageNet [38] (128k train images) pretraining to evaluate the generalization ability of our cropping strategy.

**Training details.** We pretrain all models from scratch for 400 epochs using ResNet50 [20] (except ViT-B [12] for MAE [17]) as our backbone with synchronized batch normalization [18] during pretraining. Specifically, for MoCo-v2 [6], batch size and learning rate are 256 and 0.3, respectively. The temperature  $\tau$  is 0.2, and MLP’s hidden dimension is 2048. For data augmentation, we follow the default setting of [6]. For BYOL, we set the batch size as 512 and follow the base learning rate scheduler of [49, 44]. The momentum grows from 0.99 to 1 using the cosine scheduler, and data augmentation follows the original paper [15]. DenseCL [43] is suitable for uncured images, which follows most of the settings from MoCo-v2 except for a different (dense) projection head. For MAE, we strictly follow its original pre-training settings [17].

For simplicity, we use MoCo-v2 pretrained with ‘Image-Crop’ as the backbone model  $\phi(\cdot)$  to generate our crops

Table 3. Downstream results of COCO detection and segmentation using the Mask R-CNN R50-FPN structure. Models were pretrained 400 epochs from scratch using BYOL, MoCo-v2 and DenseCL on MS-COCO with cropped pseudo images ('our', 5 crops per image by default). 'IN' and 'CC' stand for ImageNet and COCO, respectively. The SSL method SoCo\* is our reproduced result.

Method	Data	Detection			Segmentation		
		AP <sup>bbox</sup>	AP <sup>bbox</sup> <sub>50</sub>	AP <sup>bbox</sup> <sub>75</sub>	AP <sup>seg</sup>	AP <sup>seg</sup> <sub>50</sub>	AP <sup>seg</sup> <sub>75</sub>
Supervised	IN	38.9	59.6	42.7	35.4	56.5	38.1
ReSim-C4 [46]	IN	39.3	59.7	43.1	35.7	56.7	38.1
LEWEL <sub>M</sub> [23]	IN	40.0	59.8	43.7	36.1	57.0	38.7
SimCLR [5]	CC	37.0	56.8	40.3	33.7	53.8	36.1
Self-EMD [28]	CC	39.3	60.1	42.8	-	-	-
SoCo* [44]	CC	39.1	59.1	42.7	35.4	56.0	37.8
MoCo-v2	CC	38.2	58.0	41.9	34.7	55.1	37.2
MoCo-v2 <sup>our</sup>	CC	<b>39.3</b>	<b>59.1</b>	<b>42.7</b>	<b>35.6</b>	<b>56.4</b>	<b>38.0</b>
DenseCL	CC	39.3	58.9	42.9	35.4	56.0	37.8
DenseCL <sup>our</sup>	CC	<b>39.9</b>	<b>59.8</b>	<b>43.5</b>	<b>35.9</b>	<b>56.8</b>	<b>38.5</b>
BYOL	CC	38.8	58.5	42.2	35.0	55.9	38.1
BYOL <sup>our</sup>	CC	<b>40.2</b>	<b>60.4</b>	<b>43.9</b>	<b>36.4</b>	<b>57.3</b>	<b>39.0</b>

(5 per image, cf. Fig. 2 for a full illustration) on the MS-COCO dataset. We then train MoCo-v2, BYOL, DenseCL and MAE *from scratch* by generating two views from one of these crops (pseudo images), referred to as 'MoCo-v2<sup>our</sup>', 'BYOL<sup>our</sup>', 'DenseCL<sup>our</sup>' or 'MAE<sup>our</sup>', respectively.

**Downstream finetuning.** After the MoCo-v2, BYOL, DenseCL or MAE model is pretrained on our cropped pseudo object-centric images, we finetune them for the MS-COCO object detection or instance segmentation, VOC detection and Cityscapes segmentation tasks. Specifically, we finetune for 90k iterations on MS-COCO using Mask R-CNN R50-FPN and Mask R-CNN R50-C4, and finetune for 24k iterations on the VOC07+12 trainval set using Faster R-CNN R50-C4. On Cityscapes, we finetune for 40k iterations using PSANet. The learning rate and batch size are 0.02 and 16, respectively. We strictly follow the settings in ViTDet [26] for MAE pretrained models. Following previous works [15, 50], we adopt AP (average precision) for detection and instance segmentation, and adopt mAP (mean average precision) for multi-label classification. The mIoU (mean IoU), mAcc (mean accuracy) and aAcc are used for semantic segmentation. We take 3 runs on VOC detection and CityScapes segmentation tasks since these results have larger variances. All our experiments were conducted using PyTorch [36] and we used 8 GeForce RTX 3090 for our experiments. More details can be found in the appendix.

## 4.2. COCO detection and segmentation.

**ResNet models.** We first evaluate on MS-COCO object detection and instance segmentation using ResNet-50 as backbone. The results of MoCo-v2 and BYOL pretrained models using the Mask R-CNN R50-FPN structure

Table 4. Downstream results on COCO with Mask R-CNN R50-C4. All models were pretrained on MS-COCO for 400 epochs.

Method	Data	Detection			Segmentation		
		AP <sup>bbox</sup>	AP <sup>bbox</sup> <sub>50</sub>	AP <sup>bbox</sup> <sub>75</sub>	AP <sup>seg</sup>	AP <sup>seg</sup> <sub>50</sub>	AP <sup>seg</sup> <sub>75</sub>
Supervised	IN	38.1	58.1	41.1	33.2	54.8	35.0
BYOL	CC	36.9	56.7	39.4	32.4	53.5	34.3
BYOL <sup>our</sup>	CC	<b>37.9</b>	<b>57.8</b>	<b>40.7</b>	<b>33.1</b>	<b>54.3</b>	<b>35.2</b>
MoCo-v2	CC	37.3	56.7	40.4	32.8	53.5	34.9
MoCo-v2 <sup>our</sup>	CC	<b>38.1</b>	<b>57.4</b>	<b>41.4</b>	<b>33.4</b>	<b>54.2</b>	<b>35.8</b>
DenseCL	CC	38.3	57.9	41.4	33.5	54.4	35.7
DenseCL <sup>our</sup>	CC	<b>39.3</b>	<b>58.8</b>	<b>42.4</b>	<b>34.3</b>	<b>55.6</b>	<b>36.6</b>

Table 5. ViT-Base pretrained (800ep) on COCO with MAE [17], and finetuned on COCO using Mask RCNN FPN ('Mask') and Cascade RCNN FPN ('Cas.') according to ViTDet [26].

Method	Detector	Detection			Segmentation		
		AP <sup>bbox</sup>	AP <sup>bbox</sup> <sub>50</sub>	AP <sup>bbox</sup> <sub>75</sub>	AP <sup>seg</sup>	AP <sup>seg</sup> <sub>50</sub>	AP <sup>seg</sup> <sub>75</sub>
Random	Mask	27.8	45.4	29.4	26.0	43.0	27.0
MAE	Mask	38.0	57.8	41.1	34.6	55.0	37.0
MAE <sup>our</sup>	Mask	<b>43.2</b>	<b>63.8</b>	<b>47.5</b>	<b>39.0</b>	<b>60.8</b>	<b>41.8</b>
Random	Cas.	31.9	47.0	34.1	28.0	44.7	29.8
MAE	Cas.	41.7	58.8	45.1	36.3	56.2	39.1
MAE <sup>our</sup>	Cas.	<b>46.4</b>	<b>64.2</b>	<b>50.4</b>	<b>40.3</b>	<b>61.8</b>	<b>43.8</b>

are in Table 3. Specifically, our method has 1.4%, 1.1% and 0.6% AP<sup>bbox</sup> gains over the BYOL, MoCo-v2, and DenseCL baseline, respectively. Our method also achieves state-of-the-art performance within self-supervised training methods on uncropped images (i.e., ImageNet free). It outperforms ReSim [46] and SoCo [44] consistently. It also surpasses its supervised counterpart by 1.3% AP<sup>bbox</sup> and 1.0% AP<sup>seg</sup> for detection and segmentation, respectively. We also tested our method using the Mask R-CNN R50-C4 structure. As shown in Table 4, our cropping strategy has roughly 1% AP<sup>bbox</sup> gain consistently when compared with the baseline MoCo-v2, BYOL and DenseCL (i.e., 'Image-Crop'). More importantly, DenseCL with our cropping strategy has surpassed supervised pretraining by a large margin (1.2% AP<sup>bbox</sup> and 1.1% AP<sup>seg</sup> for detection and segmentation, respectively).

**ViT structure with MAE.** Since our cropping strategy is orthogonal to SSL methods, we now validate its adaptability to the non-contrastive SSL method MAE [17]. We pretrain ViT-B on COCO with original image and Our-Crop ('our') using MAE for 800 epochs and finetune COCO detection. Our cropping strategy surpasses the original Image-Crop by a significant margin for both Mask RCNN and Cascade RCNN [2] detectors, as Table 5 shows. We argue that decoupling original image to a set of image patches (Our-Crop) could *relieve the difficulty of the MAE's reconstruction*, since the pixels in one of these pseudo object-centric images will have higher correlation to each other that jointly describe a certain object content.



Table 6. Transfer learning of VOC0712 object detection using Faster-RCNN R50-C4 and CityScapes (‘City’) segmentation using PSANet [51]. Our models were pretrained using original image and ‘Our-Crop’ (*our*) on MS-COCO for the default 400 epochs.

Method	Data	VOC detection			City segmentation		
		AP	AP <sub>50</sub>	AP <sub>75</sub>	mIoU	mAcc	aAcc
Supervised	IN	53.3	81.0	58.8	77.5	86.5	95.9
Self-EMD [28]	CC	53.0	80.0	58.6	-	-	-
BYOL	CC	51.7	80.2	56.4	77.6	86.6	95.8
BYOL <sub>our</sub>	CC	<b>52.1</b>	<b>80.7</b>	<b>56.9</b>	<b>78.1</b>	<b>86.8</b>	<b>96.0</b>
MoCo-v2	CC	53.7	80.0	59.5	76.8	85.7	95.8
MoCo-v2 <sub>our</sub>	CC	<b>54.8</b>	<b>81.0</b>	<b>61.2</b>	<b>77.6</b>	<b>85.9</b>	<b>96.0</b>
DenseCL	CC	56.0	81.5	62.5	77.6	86.6	96.0
DenseCL <sub>our</sub>	CC	<b>57.2</b>	<b>82.2</b>	<b>63.4</b>	<b>78.6</b>	<b>86.9</b>	<b>96.2</b>

Table 7. Transfer learning results on 7 small classification datasets. All the models are pretrained on MS-COCO dataset for 400 epochs with BYOL SSL method using original image and our cropping strategies (*our*), respectively.

Method	CUB	Flowers	Cars	Aircraft	Indoor	Pets	DTD
BYOL	74.0	92.9	89.7	84.6	70.3	83.8	65.3
BYOL <sub>our</sub>	<b>75.4</b>	<b>94.5</b>	<b>90.4</b>	<b>85.2</b>	<b>71.0</b>	<b>84.7</b>	<b>66.2</b>

### 4.3. Transfer Learning

**VOC and CityScapes results.** We then transfer our pretrained models to VOC detection using Faster R-CNN R50-C4. We run MoCo-v2, BYOL and DenseCL with or without ‘Our-Crop’ during the pretraining stage. As shown in Table 6, the improvement of our cropping strategy over MoCo-v2 and BYOL is consistent. Specifically for MoCo-v2, we get 1.1% and 1.7% gain over its baseline for *AP* and *AP*<sub>75</sub>, respectively. Our method also surpasses the ImageNet supervised baseline and the SSL method Self-EMD [28] by a large margin. Finally, we validate our cropping method on Cityscapes semantic segmentation. As shown in Table 6, using ‘Our-Crop’ during the pretraining stage leads to consistent improvement over 3 SSL baseline methods, surpassing its supervised counterpart for all of them.

**Classification results.** Besides dense prediction tasks, we also verify the effectiveness of Our-Crop on various classification benchmarks. The results can be found in Table 7, which again clearly demonstrates the adaptability of our cropping strategy to different downstream tasks.

**ImageNet pretraining.** Now we further validate on object-centric images. We first compare Our-Crop with Selective Search [40] (SS) under the SoCo pretraining pipeline on ImageNet. Number of boxes in SS and ours is 41 and 22 per image, respectively. As shown in Table 8, larger dataset pretraining boosts downstream performance. Although the number of crops in Our-Crop is only 22 (about half of that in SS), SoCo<sub>our</sub> consistently surpassed SoCo<sup>ss</sup>. We also

Table 8. ImageNet pretraining results. SoCo [44] were pretrained with its original boxes from selective search (*ss*) or Our-Crop (*our*). BYOL and MoCo-v2 were pretrained for 200 epochs on ImageNet subset (‘IN100’, with 100 sampled images per class) using Our-Crop (3 crops per image). All pretrained models were then finetuned on COCO with Mask R-CNN R50-FPN.

Method	Data	Detection			Segmentation		
		AP <sup>bbox</sup>	AP <sub>50</sub> <sup>bbox</sup>	AP <sub>75</sub> <sup>bbox</sup>	AP <sup>seg</sup>	AP <sub>50</sub> <sup>seg</sup>	AP <sub>75</sub> <sup>seg</sup>
Supervised	IN	38.9	59.6	42.7	35.4	56.5	38.1
SoCo <sup>ss</sup>	IN	41.7	62.2	45.7	37.3	59.1	39.9
SoCo <sub>our</sub>	IN	<b>42.0</b>	<b>62.5</b>	<b>46.1</b>	<b>37.5</b>	<b>59.4</b>	<b>40.0</b>
BYOL	IN100	37.9	57.4	41.3	34.3	54.5	36.7
BYOL <sub>our</sub>		<b>38.8</b>	<b>58.5</b>	<b>42.4</b>	<b>35.2</b>	<b>55.7</b>	<b>37.5</b>
MoCo-v2	IN100	36.6	55.7	39.9	33.3	53.0	35.7
MoCo-v2 <sub>our</sub>		<b>37.8</b>	<b>57.3</b>	<b>41.2</b>	<b>34.3</b>	<b>54.5</b>	<b>36.8</b>
DenseCL	IN100	37.9	57.4	41.4	34.5	54.5	37.0
DenseCL <sub>our</sub>		<b>39.1</b>	<b>58.7</b>	<b>42.9</b>	<b>35.4</b>	<b>55.8</b>	<b>38.2</b>

sample 100 images per ImageNet class to form an ImageNet subset (‘IN100’), and adopt our cropping pipeline (cf. Sec. 3.2) to generate 3 crops per image. As Table 8 shows, training BYOL and MoCo-v2 with Our-Crop is significantly better than training with original image on IN100, demonstrating the flexibility of our cropping pipeline to not only non-curated datasets (*e.g.*, MS-COCO), but to object-centric datasets (*e.g.*, ImageNet) as well!

## 5. Ablations

In this section, we will first explore various components in our cropping pipeline (Sec. 5.1), then compare with other box generation methods in Sec. 5.2 to verify of our hypothesis: *crops that have coarse objectness scores work the best*. Besides, we do a series of hyper-parameter tuning and visualization in Sec. 5.3 to show the robustness of our method.

### 5.1. Efficiency and validity analysis

**More or less training epochs.** Since there are 5 (‘Our-Crop’) crops per image, we explored a 5x baseline for ‘Image-Crop’. As shown in Table 9, more training epochs will *not* increase the accuracy for ‘Image-Crop’, and that pretraining with ‘Our-Crop’ is consistently better than ‘Image-Crop’ with both 1x and 5x schedule. Besides, we downscale the pre-training epochs on MS-COCO from the *default* 400 to 200 and 100. As shown in Table 10, even with one fourth (100ep) of the default scheduler, Our-Crop is still better than Image-Crop. It is also viable for Our-Crop to adopt half scheduler since 400 and 200 epochs lead to quite similar results.

**Components of Our-Crop.** We pretrain BYOL (Table 11) with Image-Crop (‘baseline’), GTPad-Crop (‘s2<sup>GTPad</sup>’), Grid-Crop (‘s2<sup>grid</sup>’), Multi-Crop (‘s3<sup>multi</sup>’) and three variants of Our-Crop (‘s4-s6’). Specifically, ‘s4’ means *randomly* generated boxes (pseudo-centric crops, 5

Table 9. More training epochs (2000) of ‘Image-Crop’ pretrained on COCO, and finetuned on COCO with Mask R-CNN R50-C4.

Arch	Epochs	Detection			Segmentation		
		AP <sup>bbox</sup>	AP <sub>50</sub> <sup>bbox</sup>	AP <sub>75</sub> <sup>bbox</sup>	AP <sup>seg</sup>	AP <sub>50</sub> <sup>seg</sup>	AP <sub>75</sub> <sup>seg</sup>
BYOL	400	36.9	56.7	39.4	32.4	53.5	34.3
BYOL	2000	36.1	56.3	38.4	31.7	52.6	33.3
BYOL <sup>our</sup>	400	<b>37.9</b>	<b>57.8</b>	<b>40.7</b>	<b>33.1</b>	<b>54.3</b>	<b>35.2</b>
MoCo-v2	400	37.3	56.7	40.4	32.8	53.5	34.9
MoCo-v2	2000	37.4	56.6	40.3	32.8	53.3	34.8
MoCo-v2 <sup>our</sup>	400	<b>38.1</b>	<b>57.4</b>	<b>41.4</b>	<b>33.4</b>	<b>54.2</b>	<b>35.8</b>

Table 10. Less pre-training epochs of BYOL and MoCo-v2 with ‘Our-Crop’ on COCO and finetuned with Mask R-CNN R50-FPN.

Crop types	Epochs	Schedule	BYOL		MoCo-v2	
			AP <sup>bbox</sup>	AP <sup>seg</sup>	AP <sup>bbox</sup>	AP <sup>seg</sup>
Image-Crop	400	1.0x	38.8	35.0	38.2	34.7
	100	0.25x	39.2	35.4	38.5	34.9
Our-Crop	200	0.5x	40.1	36.2	39.2	35.5
	400	1.0x	<b>40.2</b>	<b>36.4</b>	<b>39.3</b>	<b>35.6</b>

Table 11. Comparison of different cropping strategies illustrated in Fig. 1 and the analysis of 3 components (stages) in our cropping pipeline. We consider Image-Crop (‘s0’), GTpad-Crop (‘s1<sup>GTpad</sup>’), Grid-Crop (‘s2<sup>grid</sup>’), Multi-Crop [4, 41] (‘s3<sup>multi</sup>’) and three strategies (‘s4-s6’) of Our-Crop. The three stages are detailed in Sec. 3.2, which means learning backbone weights for SCDA scores (cf. Fig. 2), anchors generation & filtering, and the final SSL learning with pseudo object-centric images, respectively.

Strategy	Stage1	Stage2	Stage3	GT box	AP <sup>bbox</sup>	AP <sup>seg</sup>
<i>different cropping</i>						
s0					38.8	35.0
s1 <sup>GTpad</sup>			✓	✓	39.6	35.9
s2 <sup>grid</sup>			✓		39.0	35.2
s3 <sup>multi</sup>			✓		39.5	35.7
<i>our cropping</i>						
s4			✓		39.1	35.5
s5		✓	✓		40.0	36.2
s6	✓	✓	✓		<b>40.2</b>	<b>36.4</b>

per image) for SSL learning, ‘s5’ using *randomly* initialized network weights for box filtering, and ‘s6’ is the default Our-Crop pipeline. It is clear that ‘s1<sup>GTpad</sup>’ and ‘s2<sup>grid</sup>’ both surpassed baseline, showing that cropping is the *key* (generating views from local object contents). And our cropping pipeline is effective than the pure multi-crop strategy. However, *GTpad* (39.6% AP<sup>bbox</sup>) is still inferior to Our-Crop (40.2% AP<sup>bbox</sup>), suggesting *groundtruth bounding boxes are not necessarily the most important information in SSL*. Surprisingly, we can also obtain similar results without ‘Stage1’ pretraining (the strategy ‘s5’), showing that our cropping strategy *does not* rely on pretrained SSL model!

Table 12. Compare with other box generation methods (Selective Search ‘ss’ and EdgeBox ‘edge’) on MS-COCO object detection and segmentation. All models were SSL pretrained on MS-COCO using BYOL and MoCo-v2 for 400 epochs.

Method	Detection			Segmentation		
	AP <sup>bbox</sup>	AP <sub>50</sub> <sup>bbox</sup>	AP <sub>75</sub> <sup>bbox</sup>	AP <sup>seg</sup>	AP <sub>50</sub> <sup>seg</sup>	AP <sub>75</sub> <sup>seg</sup>
BYOL	38.8	58.5	42.2	35.0	55.9	38.1
BYOL <sup>ss</sup>	39.3	59.4	42.9	35.6	56.3	38.3
BYOL <sup>edge</sup>	39.6	59.8	43.6	35.9	56.7	38.5
BYOL <sup>our</sup>	<b>40.2</b>	<b>60.4</b>	<b>43.9</b>	<b>36.4</b>	<b>57.3</b>	<b>39.0</b>
MoCo	38.2	58.0	41.9	34.7	55.1	37.2
MoCo <sup>ss</sup>	38.1	57.4	41.7	34.3	54.4	36.9
MoCo <sup>edge</sup>	38.5	58.2	42.0	35.0	55.4	37.5
MoCo <sup>our</sup>	<b>39.3</b>	<b>59.1</b>	<b>42.7</b>	<b>35.6</b>	<b>56.4</b>	<b>38.0</b>

Table 13. Compare with other box generation methods on VOC2007 detection and multi-label classification tasks. Models were SSL pretrained on the VOC2007 trainval set.

Method	Crop types	AP <sub>50</sub> <sup>bbox</sup>	mAP <sup>l</sup>
BYOL	Image-Crop	63.3	39.8
	Selective search	68.0	36.1
	EdgeBox	68.1	30.5
	Our-Crop	<b>69.5</b>	<b>42.5</b>
MoCo-v2	Image-Crop	61.8	26.9
	Selective search	66.4	39.7
	EdgeBox	67.0	43.3
	Our-Crop	<b>67.2</b>	<b>44.9</b>

## 5.2. Comparing with existing object discovery methods

**Quantitative comparison.** We also compare our cropping strategy with two unsupervised object discovery methods, selective search [40] (SS) and EdgeBox [53]. We use them to generate 5 top boxes for each image (because we have 5 crops in ‘Our-Crop’), and then crop them out as pseudo images. We first pretrain all models on the MS-COCO dataset. As can be seen in Table 12, all cropping strategy (EdgeBox, SS and Our) lead to consistent improvement over the Image-Crop baseline, showing local object contents suits SSL better. Besides, our cropping strategy achieves the highest results among all of them. The same is true for VOC pretraining results (shown in Table 13), which clearly demonstrate the superiority of our cropping strategy over selective search and EdgeBox. Meanwhile, it is obvious that generating two views from a cropped pseudo image (SS and EdgeBox) significantly outperforms ‘Image-Crop’. This observation further shows that our pipeline is effective: it’s better to obtain two views from a cropped pseudo image rather than from the entire scene images.

**Why Our-Crop is better?** We then dig deeper into the phenomenon demonstrated in Tables 12 and 13, that is, *why our cropping strategy is better than existing object discovery methods?*. We analyze this question from the objectness aspect. Since previous findings in Table 2 suggest that



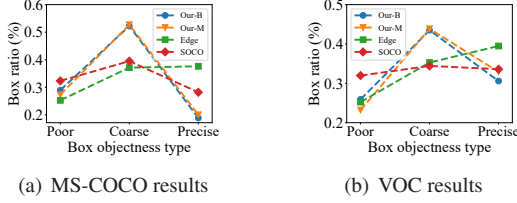


Figure 3. Objectness scores of object crops generated by our method and other object discovery methods. ‘Our-B’ and ‘Our-M’ means the backbone weights (used for box filtering, cf. Fig. 2) are obtained with BYOL and MoCo-v2 pretraining methods, respectively.

boxes with too high (precise crop) or too low (poor crop) objectness score are *both* detrimental for SSL pretraining, we calculate the ratio of boxes that fall into each category: Poor-Crop (objectness score  $< 20\%$ ), Precise-Crop (objectness score  $> 80\%$ ) and Coarse-Crop (the remaining) for EdgeBox, Selective Search and Our-Crop. Fig. 3 shows the result on MS-COCO and VOC2007, where ‘B’ and ‘M’ stands for crops filtered with BYOL and MoCo-v2 pretrained backbone, respectively. It is clear that existing object discovery seek highly accurate objects since their ratio of ‘Precise’ box are both higher than our cropping strategy. These results support that our conjecture in Sec. 3.3 and our motivation in Sec. 1 are valid: *crops with either too high or too low objectness scores are sub-optimal, and coarse crops are generally better for SSL.*

### 5.3. Hyper-parameter & Visualization

**Number of crops.** Now we study how our single hyper-parameter (number of crops per image, denoted as  $N$ ) influences SSL quality. We perform SSL pretraining on the VOC2007 trainval set, then finetune on VOC multi-label recognition and object detection benchmarks. As shown in Table 14, it consistently improves as  $N$  grows larger. When  $N = 1$ , our strategy performs fairly poorly, since  $N = 1$  means the only pseudo image contains merely a small part of an uncurated image and the global context has been lost. Bigger  $N$  can guarantee enough objects of interest and indeed boost finetuning accuracy. However, it also means higher training cost after pseudo images have been cropped out. Due to constraints on our computing power, we chose  $N = 5$  in our experiments.

**Augmentation sensitivity.** Since the pseudo object-centric images (the proposed Our-Crop) might already contain object of interest, we now validate whether the most important SSL data augmentation RandomResizedCrop is necessary for Our-Crop. We adopt ResNet-50 models pretrained on VOC2007 trainval with Image-Crop and Our-Crop, and separately remove the RandomResizedCrop operation for each pipeline. The VOC07 downstream object detection results can be found in Table 15. It is ob-

Table 14. Object detection and multi-label classification results with different  $N$  (number of cropped pseudo object-centric images per image) pretrained and finetuned on VOC2007.

Method	Top-N anchors	$AP_{50}^{bbox}$	mAP	$mAP^l$
BYOL	Image-Crop	63.3	63.0	39.8
	$N = 1$	60.7	61.2	31.1
	$N = 3$	67.7	69.3	36.8
	$N = 5$	69.5	70.7	42.5
	$N = 7$	<b>71.5</b>	<b>72.7</b>	<b>48.2</b>
MoCo-v2	Image-Crop	61.8	62.2	26.9
	$N = 1$	60.1	60.2	22.6
	$N = 3$	65.6	68.7	32.4
	$N = 5$	67.2	69.6	44.9
	$N = 7$	<b>69.3</b>	<b>72.1</b>	<b>51.2</b>

Table 15. VOC2007 object detection results. We pretrained BYOL on VOC2007 trainval with (w/) or without (w/o) the standard RandomResizedCrop augmentation (‘RRC-aug’) on original image (‘Image-Crop’) or the pseudo-centric images (Our-Crop).

Method	Crop types	RRC-aug	$AP_{50}^{bbox}$	AP drop
BYOL	Image-Crop	w/	<b>63.3</b>	(-0.0)
		w/o	54.2	(-9.1)
BYOL	Our-Crop	w/	<b>69.5</b>	(-0.0)
		w/o	65.0	(-4.5)



Figure 4. Visualization of our objectness score map and the cropped boxes on COCO (1st row) and VOC (2nd row) images. The backbone that generates the objectness scores were pretrained using MoCo-v2 on COCO & VOC ‘Image-Crop’, respectively. Best viewed in color.

vious that models pretrained with Our-Crop (4.5%  $AP_{50}^{bbox}$  drop) is less sensitive to that with Image-Crop (9.1%  $AP_{50}^{bbox}$  drop). Our-Crop without RandomResizedCrop even surpasses Image-Crop with this augmentation by 1.7%  $AP_{50}^{bbox}$ !

**Visualization.** At last, we randomly sampled a few images from the VOC2007 trainval set and the MS-COCO train2017 set, then calculated their normalized score maps  $\hat{A}$  (cf. Fig 2). The left image of each pair is the normalized map  $\hat{A}$ , while the right one shows the final generated coarse cropping boxes with top five scores. Note that the score maps are color coded (red and blue are high and low scores, respectively). As can be seen in Fig 4, our normalized score map effectively captures objects, and filters out unnecessary background. These attention maps again demonstrate the power of current SSL methods: an SSL pretrained model’s weights are already indicative of objects [3].

We also plot the relation between objectness of each cropping strategy and its corresponding results in Fig. 5

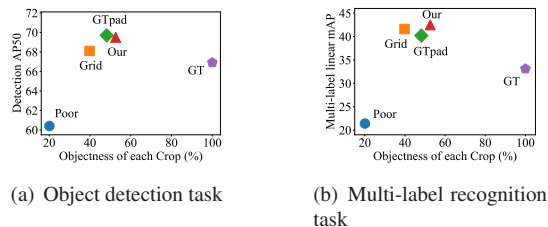


Figure 5. Results of VOC07 downstream tasks with different types of pretraining crops and their (averaged) objectness scores. Models are SSL pretrained with BYOL on VOC07 trainval.

(also shown in Table 2), which clearly supports our motivation: *crops with coarse objects work the best*.

## 6. Conclusions

In this paper, we argued that successfully cropping coarse objects out benefits self-supervised learning (SSL) with uncurated scene images. Based on this finding, we designed a novel and efficient crop generation strategy, which utilizes the unsupervised model weights and anchor generator to obtain a few coarse crops with varying shapes and sizes. The crops (pseudo object-centric images) are then fed into object-centric SSL methods. Experiments on detection and segmentation benchmarks have clearly verified the effectiveness of our pipeline over other SSL methods on uncurated scene images. We have also carefully designed a series of ablations to verify the efficiency and validity of our cropping strategy from both empirical and theoretical aspects.

As for the limitations, it remains unclear what’s the best objectness score (a certain number) of the pseudo-centric images that benefits *SSL pretraining*. We argued that mid-level objectness works better since it contains both objects and contexts. There might be other statistical metrics besides objectness that could also evaluate the quality of each cropping strategy. In the future, we will fully explore these questions that are helpful for various downstream tasks.

## References

- [1] Yutong Bai, Xinlei Chen, Alexander Kirillov, Alan Yuille, and Alexander C. Berg. Point-level region contrast for object detection pre-training. In *CVPR*, pages 16061–16070, 2022. 1, 3
- [2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018. 6
- [3] Yun-Hao Cao and Jianxin Wu. A random CNN sees objects: One inductive bias of CNN and its applications. In *AAAI*, pages 194–202, 2021. 4, 9
- [4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, pages 9912–9924, 2020. 1, 3, 8
- [5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 6
- [6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 3, 5
- [7] Yingyi Chen, Xi Shen, Yahui Liu, Qinghua Tao, and Johan AK Suykens. Jigsaw-vit: Learning jigsaw puzzles in vision transformer. *Pattern Recognition Letters*, 166:53–60, 2023. 3
- [8] Zihan Chen, Hongyuan Zhu, Hao Cheng, Siya Mi, Yu Zhang, and Xin Geng. Lpcl: Localized prominence contrastive learning for self-supervised dense visual pre-training. *Pattern Recognition*, 135:109185, 2023. 3
- [9] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, pages 3286–3293, 2014. 4
- [10] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, pages 3606–3613, 2014. 5
- [11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 5
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 5
- [13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 1, 4, 5
- [14] Priya Goyal, Mathilde Caron, Benjamin Lefaudeaux, Min Xu, Pengchao Wang, Vivek Pai, Mannat Singh, Vitaliy Liptchinsky, Ishan Misra, Armand Joulin, and Piotr Bojanowski. Self-supervised pretraining of visual features in the wild. *arXiv preprint arXiv:2103.01988*, 2021. 3
- [15] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent—a new approach to self-supervised learning. In *NeurIPS*, pages 21271–21284, 2020. 1, 3, 5, 6
- [16] Zhaohan Daniel Guo, Bernardo Avila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *ICML*, pages 3875–3886, 2020. 3
- [17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, June 2022. 5, 6

- [18] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, pages 4918–4927, 2019. 5
- [19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, pages 2961–2969, 2017. 4
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5
- [21] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, pages 4182–4192, 2020. 3
- [22] Olivier J Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *ICCV*, pages 10086–10096, 2021. 1, 2, 3, 4
- [23] Lang Huang, Shan You, Mingkai Zheng, Fei Wang, Chen Qian, and Toshihiko Yamasaki. Learning where to learn in cross-view self-supervised learning. In *CVPR*, pages 14451–14460, 2022. 6
- [24] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013. 5
- [25] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 3
- [26] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022. 6
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014. 1, 3, 5
- [28] Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet. *arXiv preprint arXiv:2011.13677*, 2020. 1, 3, 6, 7
- [29] Zhenfei Luo, Yixiang Dong, Qinghua Zheng, Huan Liu, and Minnan Luo. Dual-channel graph contrastive learning for self-supervised graph-level representation learning. *Pattern Recognition*, 139:109448, 2023. 3
- [30] Xin Ma, Xiaoqiang Zhou, Huaibo Huang, Gengyun Jia, Zhenhua Chai, and Xiaolin Wei. Contrastive attention network with dense field estimation for face completion. *Pattern Recognition*, 124:108465, 2022. 3
- [31] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5
- [32] Shlok Kumar Mishra, Anshul Shah, Ankan Bansal, Janit K Anjaria, Abhyuday Narayan Jagannatha, Abhishek Sharma, David Jacobs, and Dilip Krishnan. Object-aware cropping for self-supervised learning. *Transactions on Machine Learning Research*, 2022. 3
- [33] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *ICPR*, pages 850–855, 2006. 5
- [34] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *CVPR*, pages 1447–1454, 2006. 5
- [35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 3
- [36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019. 6
- [37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 1, 4
- [38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1, 5
- [39] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In *NeurIPS*, pages 6827–6839, 2020. 3
- [40] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 1, 3, 4, 7, 8
- [41] Wouter Van Gansbeke, Simon Vandenhende, Stamatis Georgoulis, and Luc V Gool. Revisiting contrastive methods for unsupervised learning of visual representations. In *NeurIPS*, pages 16238–16250, 2021. 3, 5, 8
- [42] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5
- [43] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *CVPR*, pages 3024–3033, 2021. 1, 3, 4, 5
- [44] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. In *NeurIPS*, pages 22682–22694, 2021. 1, 3, 5, 6, 7
- [45] Xiu-Shen Wei, Jian-Hao Luo, Jianxin Wu, and Zhi-Hua Zhou. Selective convolutional descriptor aggregation for fine-grained image retrieval. *IEEE TIP*, 26(6):2868–2881, 2017. 4
- [46] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *ICCV*, pages 10539–10548, 2021. 1, 6
- [47] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. DetCo: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021. 1, 3
- [48] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. In *NeurIPS*, pages 28864–28876, 2021. 1, 3, 4

- [49] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, pages 16684–16693, 2021. 3, 5
- [50] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 6
- [51] Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *ECCV*, pages 267–283, 2018. 7
- [52] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *ICCV*, pages 10160–10169, 2021. 3
- [53] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, volume 8693 of *Lecture Notes in Computer Science*, pages 391–405. Springer, 2014. 4, 8