OBJECT-AWARE CROPPING FOR SELF-SUPERVISED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

A core component of the recent success of self-supervised learning is cropping data augmentation, which selects sub-regions of an image to be used as positive views in the self-supervised loss. The underlying assumption is that randomly cropped and resized regions of a given image share information about the objects of interest, which the learned representation will capture. This assumption is mostly satisfied in datasets such as ImageNet where there is a large, centered object, which is highly likely to be present in random crops of the full image. However, in other datasets such as OpenImages or COCO, which are more representative of real world uncurated data, there are typically multiple small objects in an image. In this work, we show that self-supervised learning based on the usual random cropping performs poorly on such datasets. We propose replacing one or both of the random crops with crops obtained from an object proposal algorithm. This encourages the model to learn both object and scene level semantic representations. Using this approach, which we call *object-aware cropping*, results in significant improvements over scene cropping on classification and object detection benchmarks. For example, on OpenImages, our approach achieves an improvement of 8.8% mAP over random scene-level cropping using MoCo-v2 based pre-training. We also show significant improvements on COCO and PASCAL-VOC object detection and segmentation tasks over the state-of-the-art self-supervised learning approaches. Our approach is efficient, simple and general, and can be used in most existing contrastive and non-contrastive self-supervised learning frameworks.

1 INTRODUCTION

In recent works on self-supervised learning (SSL) of image representations, the most successful approaches have used data augmentation as a crucial tool (Chen et al., 2020a; He et al., 2019; Grill et al., 2020; Tian et al., 2019; Caron et al., 2020). Given a randomly chosen image sample, augmentations of the image are generated using common image transformations such as cropping and resizing a smaller region of the image, color transformations (hue, saturation, contrast), rotations etc. (Chen et al., 2020a; Gidaris et al., 2018). Of these augmentations, the use of cropping seems to be clearly the most powerful (see Chen et al. (2020a), Fig. 5). This makes intuitive sense: cropping followed by resizing forces the representation to focus on different parts of an object with varying aspect ratios. This makes the representation robust to such natural transformations as scale and occlusion. The implicit assumption in this scheme is that the object of interest (classification or detection target) occupies most of the image and is fairly centered in the image, so that random crops of the image mostly result in (most of) the object still being present in the cropped image. Such an assumption holds for "iconic" datasets such as ImageNet (Krizhevsky et al., 2012). Forcing the resulting representations to be closer together maximizes the mutual information between the crops (also called *views*) (van den Oord et al., 2018; Tian et al., 2019).

However, in the case of "non-iconic" datasets such as OpenImages (Kuznetsova et al., 2020) and COCO (Lin et al., 2014), the objects of interest are small relative to the image size and rarely centered, see Fig. 1. These datasets are more representative of real-world uncurated data. We find that the default random cropping approach (which we call *scene* cropping) leads to a significant reduction in performance for self-supervised contrastive learning approaches. For example, using the default pipeline of MoCo-v2 (Chen et al., 2020b), there is a gap of 16.5% mean average precision (mAP) compared to fully supervised learning. Other state of the art methods such as BYOL (Grill et al.,

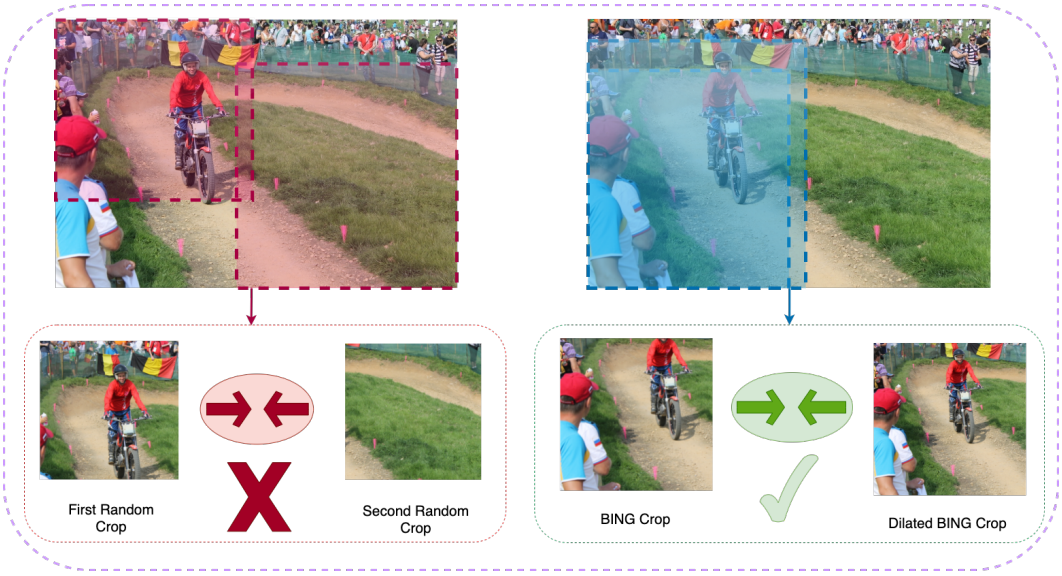

Figure 1: Illustration of object aware cropping. Top-Left: We show the original image with random crops overlaid. Bottom (red panel): Overlap between random crops tend to miss the object of interest. Top-Right: We show crops generated from the BING (Cheng et al., 2014) algorithm and also the dilated BING crop. Bottom-Right (green panel): Instead, we use BING-based object-aware crops. This incorporates both object and scene information into the MoCo-v2 (or other SSL frameworks).

2020), SwAV (Caron et al., 2020) and CMC (Tian et al., 2019) perform poorly as well (see Table 1). As we show, the core problem here is that random scene crops do not contain enough information about objects, causing degraded representation quality.

However, merely switching from scene-level crops to purely object-level crops does not exploit the correlations that exist between scenes and objects in most natural images. These correlations are helpful for downstream tasks (Xiao et al., 2020). Keeping this in mind, we introduce *object-aware* cropping, which applies a simple pre-processing step using the BING algorithm (Cheng et al., 2014). BING outputs multiple object proposals, one of which we pick at random as the first view for the SSL loss (most variants of SSL use at least two views of the same sample as "positives"). For the second view, we experiment with the following variants to incorporate both object and scene information into the representation: (1) scene-level random crop to incorporate both object and scene context (we call this setting *obj-scene*); (2) a dilated version of the BING proposal followed by random crops applied to both views (*obj-obj+dilate*) (3) a second crop which applies a random shift to the BING proposal (*obj-obj+shift*). The baseline applies scene-level random crops to both views (*scene-scene*).

We conduct a number of experiments incorporating object-aware cropping, finding consistent improvements on state of the art self-supervised methods such as MoCo-v2 (Chen et al., 2020b), BYOL (Grill et al., 2020) and Dense-CL Wang et al. (2021) across varied datasets and tasks (see Fig.2 and Section 4). We find that *Obj-Obj+Dilate* outperforms other object-aware crop methods. This approach is fast (>125 fps on an NVIDIA P100 GPU), adds minimial overhead to the baseline computation time (<1%), and is simple to implement. We will release our models and code.

2 Analysis of Self-Supervised Learning methods on the OpenImages Dataset

In this section, we first identify some limitations of state-of-the-art SSL methods such as MoCo-v2 (He et al., 2019; Chen et al., 2020b), SwAV (Caron et al., 2020) and BYOL (Grill et al., 2020) on non-iconic datasets. These methods have nearly closed the performance gap with supervised learning methods when pre-trained and linear probed on ImageNet (Deng et al., 2009). However, the performance of these methods on non-iconic datasets (where images contain multiple small objects) has not been extensively studied. OpenImages (Kuznetsova et al., 2020) is such a dataset, with images

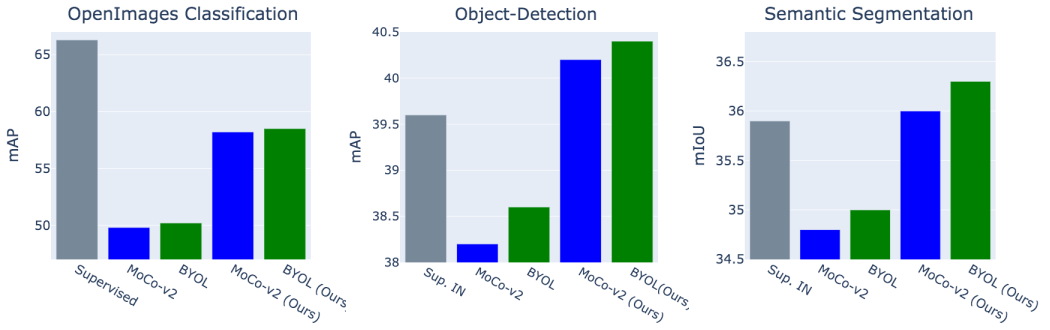

Figure 2: Our object-aware cropping approach can be easily plugged into self-supervised learning pipelines and achieves excellent results for classification on OpenImages (left), COCO object detection (middle) and COCO semantic segmentation (right). Using object-aware cropping instead of scene-level cropping provides a consistent boost on BYOL (Richemond et al., 2020) and MoCo-v2 (Chen et al., 2020b), two of the top SSL methods. In the case of COCO object detection and semantic segmentation, this boost allows us to beat pre-training on supervised ImageNet (denoted "Sup. IN"). In the case of classification on OpenImages, we reduce the gap to supervised training by nearly 50%.

of complex scenes and several objects (containing, on average, 8 annotated objects per image). It consists of a total of 9.1 million images. To perform controlled experiments on the effectiveness of cropping on SSL method performance, we construct a subset of the OpenImages dataset. We sample images that have labelled bounding boxes to enable comparisons with fully supervised learning. Secondly, we sample images with objects from at least 2 distinct classes to create a dataset that better reflects real-world uncurated data. Finally, we only consider class categories with at least 900 images to mitigate effects of imbalanced class distribution. After this processing, we have $212,753$ images present across 208 classes and approximately 12 objects per image on average. We provide further details in the supplementary Sec A. For convenience, we still refer to this dataset as OpenImages through this paper.

2.1 Performance of SSL methods

We pretrain several SSL methods MoCo-v2, CMC (Tian et al., 2019), SwAV (Caron et al., 2020) and BYOL (Grill et al., 2020) on our OpenImages dataset. MoCo-v2, BYOL, and other recent state of the art SSL approaches all relying on *scene-scene* cropping of the same image to generate positive samples. This cropping strategy has also been used as a default data augmentation in supervised learning (He et al., 2015; Krizhevsky et al., 2012; Cubuk et al., 2018). The most common workflow is: choose a random scale (default range 0.2 to 1.0) and a random aspect ratio (default range 0.75 to 1.33). A crop is made from the original image using these two values (scale and aspect ratio). Finally, the crop is resized to the final size of 224×224 pixels.

We use a ResNet-50 (He et al., 2015) deep network as the backbone for all experiments. After pre-training, we freeze the backbone weights, and train a linear classifier or fine-tune on the downstream datasets and tasks. We also train a randomly initialized ResNet-50 network in a fully supervised manner using ground truth labels and a multi-class logistic regression loss. We follow the mAP metric as described in section 4.2 of paper(Veit et al., 2017) (Eqn.(4)). Table 1 shows our results. We see a significant difference in performance between fully supervised training and SSL approaches on the OpenImages dataset, with a gap of 16.3 mAP points on average across the 4 SSL approaches considered. On ImageNet, the top-1 accuracy gap is considerably smaller with an average gap of only 8.5, nearly half that of OpenImages. The last row shows the significant boost obtained by using our object-aware cropping approach, which we describe in the next section, with MOCO-v2. The gap between SSL and supervised training on OpenImages is now the same as ImageNet.

2.2 Analysis and Motivation

We conduct further experiments to analyze better the results seen in Table 1, and to motivate our proposed approach. Our experiments help to narrow down scene cropping as one main cause of

Model	OpenImages (mAP)	ImageNet (Top-1 %)
Supervised Performance	66.3	76.2
CMC (Tian et al., 2019)	48.7 (-17.6)	60.0 (-16.2)
BYOL (Grill et al., 2020)	50.2 (-16.1)	70.7 (-5.5)
SwAV (Caron et al., 2020)	51.3 (-15.0)	72.7 (-3.5)
MoCo-v2 (Scene-Scene crop)	49.8 (-16.5)	67.5 (-8.7)
MoCo-v2 (Object-Object+Dilate crop) (Ours)	58.6 (-7.7)	68.0 (-8.2)

Table 1: Classification results on OpenImages and Imagenet. For each SSL method, we show in parentheses the gap to fully supervised training (same number of epochs). The last row shows that our proposed approach using *obj-obj+dilate* cropping reduces the gap on OpenImages by nearly half compared to the baselines, improving over the *scene-scene* cropping based SSL methods by between 6.8 to 9.4 mAP points. We also observe improvements on ImageNet as well.

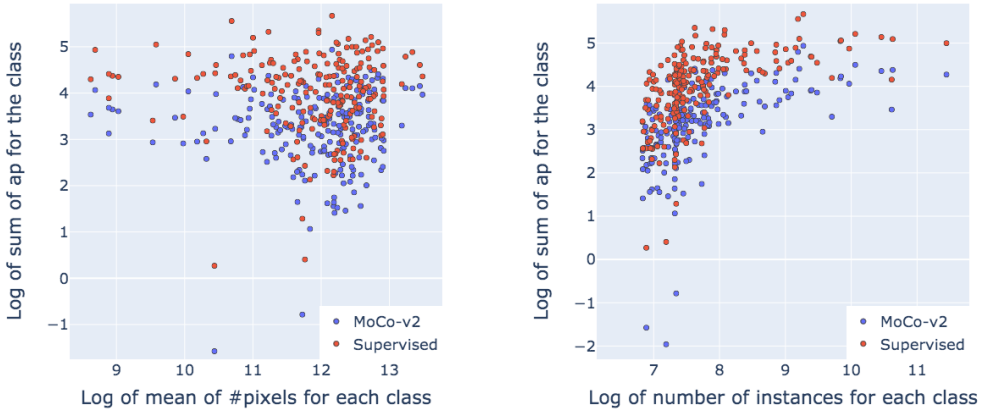

Figure 3: Analysis of OpenImages data distribution. Left: Performance of supervised and MoCo-v2 pre-training as a function of the scale of the objects; we plot the log of average of pixels against the sum of AP for each class. We see no discernible pattern of performance of MoCo-v2 or supervised learning as a function of object scale. Right: Performance of supervised learning and MoCo-v2 as a function of the number of instances in a class; we plot the log number of instances in a class against the sum of AP for that class. We do not see any discernible pattern of performance difference as a function of class size.

the poor performance of SSL on OpenImages, rather than other differences with ImageNet, such as object size, class distributions or image resolution.

Object Size: We compare MoCo-v2 performance to that of fully supervised learning, with both methods using scene-based cropping. Fig. 3 (left) shows that the performance gap between supervised learning and SSL methods does not vary significantly for objects of different sizes in OpenImages. This suggests that once object sizes are below a threshold where scene cropping tends to ignore object information, MoCo-v2 performance is mostly independent of object scale.

Long-tail Distribution: Even after selecting at least 900 images per class, our OpenImages subset has a significant variation in the number of images per class (from around 1000 to 60000). Fig. 3 (right) plots the performance of MoCo-v2 and supervised training as a function of the number of instances in each class. We do not see a significant change in relative performance as the number of instances in a class changes. This rules out long tails of the distribution as a cause for the poor absolute performance of MoCo-v2 on OpenImages.

Can ImageNet pre-training help? We pre-trained a supervised model on ImageNet and then fine-tuned the final fully-connected layer on the OpenImages dataset. We can see from the second column

				Scene-Scene Crop					
Supervised	IN Pretraining	Resize to (384×384)	GT-Crop	0.8-1.0	0.6-1.0	0.4-1.0	0.2-1.0	0.1-1.0	0.05-1.0
66.3	28.3	45.9	45.3	26.5	37.6	45.6	49.8	46.1	43.1

Table 2: Linear evaluation on our OpenImages dataset with different pre-training strategies with MoCo-v2 (see Section 2 for details). Column 1 uses fully supervised learning on OpenImages. We see that for self-supervised pre-training, no specific range of scene crops helps to close the large gap between SSL and supervised training. However, there is a sweet spot of scene crop range where MoCo-v2 performance is highest.

in Table 2 that this pre-training does not help to close the performance gap. One of the reasons that ImageNet pre-training does not help is that OpenImages and ImageNet have significantly different class distributions (e.g. see Li et al. (2019) for a detailed analysis).

Can resizing the images help? We also experimented with resizing the images in OpenImages to the same approximate size (384×384) and aspect ratio as ImageNet. The result is shown in the third column of Table 2 confirming that controlling for image size does not help to close the gap.

Can cropping on ground truth objects help? We see from column 4 of Table 2 that using random cropping on ground truth object boxes does not help reduce the performance gap either. As we show later in Section3 that learning from both object and context is important for learning semantic information on multi-object datasets.

Varying the lower scale of random resized crop: MoCo-v2 (Chen et al. (2020b)) used scene crops whose size was chosen from a uniform distribution ranging from 20% to 100% of the ImageNet image size (384×384). Since OpenImages images are bigger and objects generally occupy a smaller fraction compared to ImageNet, we vary the lower bound for scene crops to measure the impact. The last six columns of Table 2 shows that varying the range of scene crop is insufficient to close the performance gap.

We conclude that the performance gap between supervised training and SSL training is likely due to the data augmentation, rather than characteristics of the image distribution. Further analysis experiments are provided in the supplementary (Sec C).

3 Proposed Approach

Our analysis shows that SSL methods based on either purely object crops, or purely scene crops, both perform worse than supervised learning. This suggests a third option: incorporating *both* object and scene information in the crops. Our hypothesis is that context around an object is helpful to learn robust representations due to the natural correlations between scene and object (Hinton, 2021). However, it is unclear what mix of object and scene information is most helpful in learning robust representations. Our main contribution is a simple and effective approach to achieving this cropping mechanism. We also experiment with a number of plausible baselines to incorporate scene and object information.

To enable object-awareness, we consider three object proposal models: BING (Cheng et al., 2014) and Edge-Boxes (Zitnick & Dollár, 2014), both of which are trained with boxes from Pascal-VOC; and an unsupervised object proposal method (Vo et al., 2019). BING (Cheng et al., 2014) uses the norm of gradients within a fixed window size as a simple feature that is fed into a cascaded SVM framework to make object proposals. This method is trained on the ground-truth bounding boxes of 2501 images of the PASCAL-VOC (Everingham et al., 2010) training set. Note that BING does not use any class label, but only label independent bounding box information. BING has the advantage of extremely fast performance (>125 fps on an image of size 300×300), generates many proposals, and generalizes well to many datasets, as shown in Cheng et al. (2014) and seen in our empirical results. This makes BING well suited as an object proposal algorithm. In our experiments, we generate up to 10 proposals per image and select one object uniformly at random among these proposals. The details of other object proposals are present in supplementary Section B. Our results in Table 6 show that other object proposal methods provide sufficient quality and the choice may be mostly dependent on the speed of the method. We experiment with four types of cropping in the rest of the paper:

(a) Dilated object proposals (Obj-Obj+Dilate Crop): Scene pixels spatially close to the object are more likely to have a positive correlation with the object. With this intuition, we generate the second view by dilating the BING proposal. We dilate the box by 10% or 20% of the image size, followed by a random crop. Changing δ gives us control over how much scene information is incorporated (a value of 10% works well in most cases). Note that the original and dilated boxes are both followed by a random crop, ensuring that the first view is not trivially included in the second view. The choice of which crop to use as query or key is arbitrary and both object and dilated object crops can be used as either key and query. We also consider other baselines listed below which are other plausible approaches to incorporating scene information. However, as shown in the results Section 4, dilated crops are the best performing method.

(b) Scene-Scene Crop: We take two random crops in an image and treat them as positive views. This is the default approach used in MoCo-v2 and other SSL approaches. From our analysis in Section 2, this approach performs poorly on datasets such as OpenImages.

(c) Shifted object proposals (Obj-Obj+Shift Crop): Unlike Obj-Obj+Dilate, here, the second view is a box selected at random within a pre-specified distance range of the first box (the BING proposal). To choose a pre-specified distance, we choose a random value for the offset from a few ranges: 80-100 pixels; 100-120 pixels etc.

(d) Random crop (Obj-Scene Crop): The first view is the BING proposal; and a regular random crop at scene level is the second view. This method provides information at both object and scene levels, and we use it as a baseline for adding context information to the model.

Discussion on lower scale of random resized crop: As stated above, for both the Obj-Obj+Dilate and Obj-Obj+Shift crops, we use a random crop on the BING object proposal or its shifted or dilated version, to generate the final views. Since the object proposal itself is a small fraction of the image (e.g. in COCO, an object crop typically covers about 39% of the image), using the usual default lower value for the random crop range (usually 0.2) works poorly as it results in extremely small crops from the image. Therefore, we set the lower limit such that it matches the minimum sized crop in case of the usual scene crop ($s_{\text{min}} = \frac{0.2}{\text{average BING proposal size}}$) .

Different projection heads for Object and Scene crops: The projection head, introduced in Chen et al. (2020a) is an important component of most SSL methods. This is a deep network that maps representations from the encoder backbone to a lower-dimensional space where the loss function is applied. In Chen & He (2020), a single projection head is used for both views. We find it beneficial for performance to use two different projection heads: one for the first view (BING crop) and another for the second view for obj-scene, obj-obj+dilate and obj-obj+shift crops. For scene-scene crops we always use a single projection head. We hypothesize that the different projection networks specialize to either an object-specific representation, or to a representation that incorporates scene information.

Other than the above change to the cropping methodology, we leave the rest of the SSL pipeline unchanged: other data augmentations such as color shifts are applied in the usual manner, and the loss functions and training regimes are also unchanged. Our approach therefore involves really simple changes to most SSL pipelines, involving only a few lines of code.

4 RESULTS

We created a subset of the OpenImages dataset with 212k images, as described in Section 2. We also experiment with the complete OpenImages ($\sim$1.9 million images.) In addition, we perform pre-training on ImageNet (Deng et al., 2009) and MS-COCO (Lin et al., 2014). ImageNet (with 1.2M training images) has been extensively used and is the standard dataset used for benchmarking of SSL methods. MS-COCO has $\sim$ 118k training images and 896k labelled objects which is approximately 7 objects per image. For pre-training on MoCo-v2, we closely follow the standard protocol described in Chen et al. (2020b). We randomly select from 10 object proposals provided by BING, Edge-Boxes or the unsupervised proposal method of (Vo et al., 2019). All our training and evaluation is performed on a ResNet-50 (He et al., 2015).

We evaluate the pre-trained models on classification (linear evaluation), object detection and semantic segmentation. For VOC object detection, COCO object detection and COCO semantic segmentation,

Model	Crops	Obj-Obj+Dilate	Obj-Scene	Scene-Scene	mAP
Supervised	-	-	-	-	66.3
MoCo-v2	Ground Truth boxes	-	✓	-	58.9
MoCo-v2	Ground Truth boxes	✓	-	-	60.2
MoCo-v2	-	-	-	✓	49.8
BYOL	-	-	-	✓	50.2
MoCo-v2	Unsupervised proposal boxes	-	✓	-	58.0
MoCo-v2	EdgeBoxes crops	-	✓	-	57.1
MoCo-v2	BING crops	-	✓	-	58.1
BYOL	BING crops	-	✓	-	58.5
MoCo-v2	BING crops	✓	-	-	58.6
BYOL	BING crops	✓	-	-	59.1

Table 3: Crop approaches on OpenImages: using BING crops to generate one view, and a dilated crop or a scene crop for the other positive, we are able to reduce the difference between SSL and Supervised Learning by close to 50% (compare the last two rows to the first row). Using ground-truth boxes to generate crops from OpenImages improves the pre-training performance marginally compared to BING crops. Obj-Obj+Dilate (last two rows) have the best performance, although Obj-Scene also does well compared to Scene-Scene.

Description	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
Supervised (Random Initialization)	32.8	50.9	35.3	29.9	47.9	32.0
Supervised (ImageNet Pre-trained)	39.7	59.5	43.3	35.9	56.6	38.6
MoCo-v2 (Chen et al., 2020b)	38.2	58.9	41.6	34.8	55.3	37.8
BYOL (Hénaff et al., 2021)	38.8	58.5	42.2	35.0	55.9	38.1
Dense-CL (Wang et al., 2021)	39.6	59.3	43.3	35.7	56.5	38.4
CAST (Selvaraju et al., 2020) (180K steps)	39.4	60.0	42.8	35.8	57.1	37.6
Self-EMD (Liu et al., 2021a) (Uses BYOL)	39.8	60.0	43.4	-	-	-
MoCo-V2 (Obj-Scene) (Ours)	39.4	59.8	42.9	35.8	57.8	38.7
MoCo-v2 (Obj-Obj+Dilate) (Ours)	**39.7**	**60.1**	**43.4**	**36.0**	**57.3**	**38.8**
MoCo-v2 (Obj-Obj+Dilate) (180k steps)	**40.2**	**60.6**	**43.6**	**36.3**	**57.4**	**39.0**
BYOL (Obj-Obj+Dilate) (Ours)	**40.1**	**60.8**	**43.6**	**36.4**	**58.4**	**39.5**
Dense-CL (Obj-Obj+Dilate) (Ours)	**40.4**	**60.4**	**44.0**	**36.6**	**57.9**	**39.5**

Table 4: Object detection (first 3 columns) and Semantic Segmentation (last 3 columns) results on COCO dataset. All SSL models have been pre-trained on COCO dataset and then finetuned on COCO. Note that for the same number of finetuning iterations (180K), we outperform CAST (Selvaraju et al., 2020) which also relies on localized crops. All other methods are run for 90K, finetuning iterations. For any SSL method, we compare (BYOL, Moco-v2, Dense-CL) adding Obj-Obj+Dilate crop improves performance.

we closely follow the common protocols listed in Detectron2 (Wu et al., 2019). For VOC object detection, we evaluate on the Faster-RCNN(C4-backbone) (Ren et al., 2015) detector on VOC `trainval07+12` dataset using the standard 1 $\times$ standard protocol. For COCO-Object detection and semantic segmentation, we fine tune on the MaskRCNN detector (FPN-backbone) (He et al., 2018) on COCO `train2017` split (118k images) with the standard 1 $\times$ schedule, evaluating on the COCO 5k `val2017 split`. We compare to the state of the art SSL methods, including Self-EMD (Liu et al., 2021a), DetCon (Hénaff et al., 2021), BYOL (Richemond et al., 2020), DenseCL (Wang et al., 2021) and the default MoCo-v2 (Chen et al., 2020b).

Description	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
COCO: MoCo-v2 (Scene-Scene crop)	38.2	58.9	41.6	34.8	55.3	37.8
COCO: MoCo-v2 (Obj-Obj+Dilate crop)	**40.2**	**60.6**	**43.6**	**36.3**	**57.4**	**39.0**
VOC: MoCo-v2 (Scene-Scene crop)	56.1	81.3	61.3	-	-	-
VOC: MoCo-v2 (Obj-Obj+Dilate crop)	**57.6**	**82.5**	**63.8**	-	-	-

Table 5: Object detection (first 3 columns) and semantic segmentation (last 3 columns) results on COCO (first 2 rows) and VOC (last 2 rows). All SSL models have been pre-trained on complete OpenImages dataset(1.9 million images) for 75 epochs and then finetuned on COCO and VOC dataset.

Table 1 and Table 3 shows results on OpenImages dataset. We perform ablation studies on 3 different types of crops. We can see that the best performing Obj-Obj+Dilate crops outperform the baseline by 8.2 mAP, closing the gap between supervised learning and MoCo-v2 baseline by almost 50%. Obj-Scene crops also outperform the baseline by 8.1 mAP. Obj-Obj+Shift crops (54.1 mAP) do not perform as well as Obj-Obj+Dilate or Obj-Scene. We believe this is because the second object crop is chosen at a certain minimum radius from the first (otherwise, the overlap between the crops is too large to produce meaningful learning). The second object crop, therefore, contains only part of the object. Conversely, a dilated object crop would potentially contain the entire object and more scene information; therefore, the representation from the dilated object-crop contains complementary information from both the object and the scene. We also show in Table 3 an ablation with ground truth bounding boxes being used to guide the object cropping. This performs marginally better than the use of BING crops, suggesting that a tight fit around the object is not necessary for improved representations.

Table 4 shows results on object detection and semantic segmentation for COCO (by pre-training on COCO `trainval2017` datasets and finetuning on COCO). We train MoCo-v2 and BYOL models. Our MoCo-v2 Obj-Obj+Dilate cropping outperforms MoCo-v2 Scene-Scene baseline and Obj-Scene cropping. For semantic segmentation, we outperform the MoCo-v2 baseline. Our proposed cropping is agnostic to the pre-training SSL method; we show state-of-art results by adding our approach to Dense-CL (Wang et al., 2021). We also outperform the CAST model (Selvaraju et al., 2020) which also uses localized crops based on saliency maps: our approach is simpler and performs better by around 0.5 mAP. Table 6 (supplementary) shows results of object detection on PASCAL-VOC. We pre-train on COCO and then fine-tune on VOC. Obj-Obj+Dilate cropping outperforms the MoCo-v2 baseline by 3.2 mAP; Obj-Scene cropping outperforms the MoCo-v2 baseline by 2.4 mAP and the BYOL baseline by 2.5 mAP. Improved results on iconic datasets like Aircraft, Birds and Cars can be found in Supplementary (Table 10). Additional results on varying number of proposals used can be found in Supplementary (Table 8).

Table 5 shows results on object detection and semantic segmentation for COCO and object-detection on VOC by pre-training on full OpenImages dataset (Kuznetsova et al., 2020) (all 1.9 million images) for 75 epochs. We show improved performance over the baseline on both object detection and semantic segmentation tasks by using Obj-Obj+Dilate crops. Our proposed dilation method works not only for small multi-object datasets like COCO but also for datasets like OpenImages and performs well under a transfer learning setup.

Results on ImageNet: We also show improved results on ImageNet pre-training using object-aware cropping (Table 13 Supplementary) and MoCo-v2. For object detection on VOC2007, we see an improvement of 1.0 mAP; and a 0.5 mAP improvements for object detection on COCO. Our approach is thus adaptable to the pre-training dataset and SSL algorithm.

Use of Multiple Projection Heads: The use of different projection heads for each view on OpenImages classification gives us a boost of 1.1 mAP on Obj-Obj+Dilate crop. Pre-training on COCO and finetuning on VOC dataset for object-detection task gives a boost of 0.4 mAP. Hence using multiple projection heads results in a consistent improvement.

Varying Dilation Parameter: Table 9 (supplementary) shows the effect of varying the dilation parameter. No dilation reduces to an Obj-Obj cropping and a large dilation parameter corresponds to Obj-Scene. A sweet spot exists at a moderate dilation value of $\delta = 0.1$ for COCO object detection.

Computational Cost: BING adds negligible time to the pre-training. Generating object proposals takes 29 mins for the full OpenImages dataset (one-time cost) and 16 mins for COCO. Instead of pre-generating, adding the BING operator to the data loader pipeline has a trivial overhead (+0.1%).

Analysis of Obj-Scene and Obj-Obj+Shift crops: As shown in Tian et al. (2019), the performance of contrastive self-supervised methods is have "sweet spot" which correspond to an optimal amount of MI between views. We perform a similar analysis on the Obj-Scene crop approach by creating crops with progressively higher overlaps. Fig.7(left) (supplementary) shows the performance peaks around an overlap of 57.2%. This finding is consistent with that of (Tian et al., 2019). Scene-Scene crop has a higher optimal overlap of 65.2%. Varying the radius of Obj-Obj+Shift crop (Fig 7(right) for OpenImages; Fig.6(right) for COCO shows similar results. We show additional analysis on overlap between crops and object of interest in Section D.

5 Related Work

Recent progress in self-supervised learning, based on contrastive and non-contrastive approaches, has achieved excellent performance on various domains, datasets and tasks (He et al., 2019; Chen et al., 2020b;a; van den Oord et al., 2018; Tian et al., 2019; Gidaris et al., 2020; Misra & van der Maaten, 2019; Tian et al., 2020; Wu et al., 2018; Grill et al., 2020; Gidaris et al., 2018; Larsson et al., 2017; Noroozi et al., 2018; Pathak et al., 2016). The top-performing methods have all used related ideas of pulling together "views" of a sample in representation space. Some of these approaches, in addition, use negative samples to add a "push" factor, and this is termed contrastive self-supervised learning. Theoretical and empirical studies have been published to better understand the behavior and limitations of these approaches (Arora et al., 2019; Xiao et al., 2021; Purushwalkam & Gupta, 2020; Tosh et al., 2021; Wang & Isola, 2020; Yang et al., 2020; Chuang et al., 2020; Liu et al., 2021b; Kalantidis et al., 2020; Newell & Deng, 2020; Cai et al., 2020).

A number of papers have observed that the default SSL approaches above (whether contrastive or not) perform poorly on uncurated datasets such as OpenImages (Kuznetsova et al., 2020). To address this, recent works have used different workarounds such as knowledge distillation (Tian et al., 2021), clustering (Goyal et al., 2021), localization (Selvaraju et al., 2020), unsupervised semantic segmentation masks (Hénaff et al., 2021), pixel-level pretext tasks(Xie et al., 2021), Instance localization (Yang et al., 2021) and local contrastive learning (Liu et al., 2021a). The common element among top-performing image-based SSL approaches, regardless of dataset, task or architecture, is their reliance on strong data augmentations such as random cropping, gaussian blurring, color jittering or rotations. These augmentations create meaningful positive views, and other randomly sampled images in the dataset are used to create negative views in the case of contrastive SSL methods. SSL data augmentation pipelines are adapted from the supervised learning literature (Cubuk et al., 2018; Zoph et al., 2020; Krizhevsky et al., 2012; Simard et al., 2003; DeVries & Taylor, 2017; Cubuk et al., 2018; Zhang et al., 2018; Cubuk et al., 2019; Wu et al., 2019; Yun et al., 2019; Lim et al., 2019; Hataya et al., 2019).

The closest work to our object cropping work is (Selvaraju et al., 2020), which introduces a technique to choose crops around objects based on saliency maps (Selvaraju et al., 2016), showing good improvements over the baseline of random crops for COCO datasets (see Table 4). As shown in our results, Obj-Obj+Dilate crop consistently performs better than (Selvaraju et al., 2020) (Table 4). Our approach is also significantly simpler to incorporate into existing pipelines, requiring no change to the training, architecture or loss. Gansbeke et al. (2021) show that constrained multi-cropping improves performance of SSL methods: our approach can be incorporated into their pipeline to further improve performance.

6 Conclusion

We have introduced object-aware cropping, a simple, fast and highly effective data augmentation alternative to random scene cropping. We conducted numerous experiments to show that object cropping significantly improves performance over scene cropping for self-supervised pre-training for classification, object detection and semantic segmentation on a number of datasets. The approach can be incorporated into most self-supervised learning pipelines in a seamless manner.

7 ETHICS

Since our work is a technical contribution, it's ethical concerns will be dependent on the underlying models, which we hope are used in a positive manner for the good of society.

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019.

Tiffany Tianhui Cai, Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Are all negatives created equal in contrastive instance discrimination?, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020b.

Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3286–3293, 2014.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8765–8775. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/63c3ddcc7b23daa1e42dc41f9a44a873-Paper.pdf.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *ArXiv*, abs/1805.09501, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space, 2017.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Revisiting contrastive methods for unsupervised learning of visual representations, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words, 2020.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, 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, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019.

Geoffrey E. Hinton. How to represent part-whole hierarchies in a neural network. *ArXiv*, abs/2102.12627, 2021.

Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection, 2021.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pp. 1–26, 2020.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 840–849, 2017.

Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S. Davis. An analysis of pretraining on object detection, 2019.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment, 2019.

Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet, 2021a.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive, 2021b.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.00737. URL http://dx.doi.org/10.1109/CVPR42600.2020.00737.

Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases, 2020.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2016.

Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. *arXiv preprint arXiv:2012.04630*, 2020.

Patrice Simard, David Steinkraus, and John Platt. Best practices for convolutional neural networks applied to visual document analysis. pp. 958–962, 01 2003. doi: 10.1109/ICDAR.2003.1227801.

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

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Yonglong Tian, Olivier J. Henaff, and Aaron van den Oord. Divide and contrast: Self-supervised learning from uncurated data, 2021.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models, 2021.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision, 2017.

Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Perez, and Jean Ponce. Unsupervised image matching and object discovery as optimization, 2019.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training, 2021.

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

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. doi: 10.1109/CVPR.2018.00393.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16684–16693, 2021.

Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3987–3996, 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

C. L. Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. Rethinking pre-training and self-training, 2020.

A Details of OpenImages-Subset

We show in Fig. 4(left) the plot of the number of images and selected class ID's. The mean of the number of images per class is 3762.13. We show additional statistics by removing the classes with more than 10k instances present. We can see this in Fig. 4(middle). The mean of number of images per class also comes down from 3762 to 2203. We plot the width and height of all the images present in OpenImages-Subset. We can see from Fig. 4(right) that most images are in around 1000×1000 pixels. with some being significantly larger.

We created OpenImages-Subset by only keeping classes from OpenImages which had at least 900 images, resulting in a total of 208 classes with 3762 images per class on average. For each image in OpenImages-Subset we computed how many images have an extra class that is not present in these 208 classes: resulting in a very small number 0.002: mist objects in the subset are from the 208 classes. The total number of images is 212k. Finally we computed the number of images with N objects, where N varies from 2 to 7. These numbers are: $2/51008; 3/56131; 4/43426; 5/26011; 6/13693; 7/6593$.

B Other object proposal methods

We also experiment with two object proposal methods, i.e Edge-Boxes (Zitnick & Dollár, 2014) and an unsupervised object proposal method (Vo et al., 2019). Here we discuss their methodology in more detail.

Edge-Boxes (Zitnick & Dollár, 2014) is based on a simple heuristic: an object is more likely to be contained in a box if the number of contours wholly inside the box exceed the contours that cross the boundary of the box. In our experiments, we found that EdgeBoxes was slower than BING, although the performance on the generated boxes was similar to that of BING.

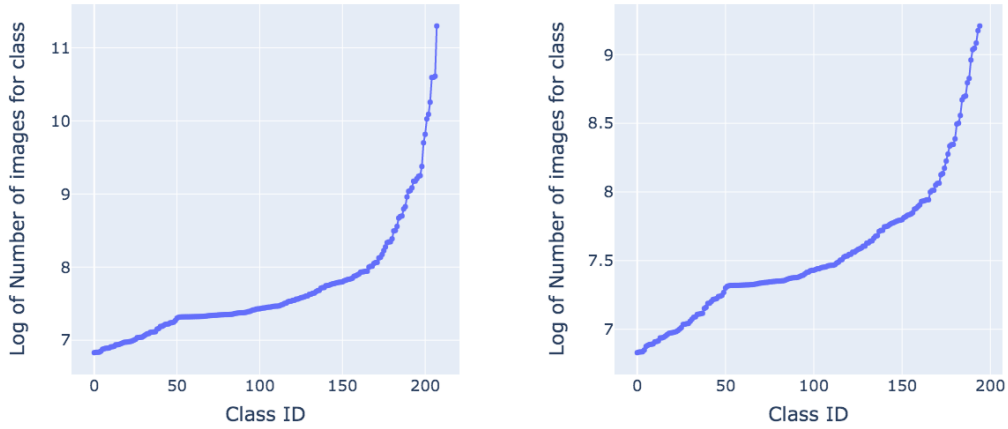

Figure 4: **Left:** distribution of number (log) of images in each class for each of 208 selected classes. **Middle:** distribution after removing classes which have more than 10k images. **Right:** scatter plot of the height and width.

We also consider an alternative unsupervised object proposal method (Vo et al., 2019), which uses a robust matching technique that relies on appearance and geometric consistency constraints to assign confidence scores to region proposals. While fully unsupervised, this approach performs similar to BING as shown in Table 3 and Table 6, but is significantly slower than BING. Our results therefore suggest that many object proposal methods provide sufficient quality and the choice may be mostly dependent on the speed of the method.

C More Analysis on OpenImages

Varying the temperature parameter: Some self-supervised methods are highly dependent on the temperature hyperparameter, which can can be dataset dependent. The default temp parameter in MoCo-v2 (Chen et al., 2020b) (tested on ImageNet) is 0.07. To ensure that this temperature setting was not a concern for OpenImages, we trained MoCo-v2 models with a number of temperature hyperparameters i.e $0.05, 0.7, 0.1, 0.2$ and 0.3. We found that a temperature setting of 0.2 gives the best result which is an increase of 1.2 mAP over the default value, which is significantly lower than the increase of 8mAP we get with object cropping.

MoCo-v2 training dynamics: Another potential concern with OpenImages is poor training dynamics of MoCo-v2 on OpenInmages. To verify that this was not the case, we computed the mean inner product positive samples and of negative samples after training MoCo-v2 for 200 epochs. We find that the mean of the score for positive samples is 0.81 and the mean of scores of negative samples is 0.005, very close to the ideal of 1 and 0 respectively. This indicates that MoCo-v2 trains properly, and the difference between the MoCo-v2 and supervised training is unlikely to be due to poor training dynamics.

D Analysis on Object-Scene crops:

Overlap between crops and object of interest: In expectation, any two crops of a scene will overlap to some extent. If this intersection overlaps an object, the crops are more likely to contribute to representations that capture object properties. We measure the fraction of pixels in this intersection that belong to a ground truth object box, to get a sense of how this varies with different cropping strategies and data sets. We find that this fraction for COCO is 99% for object-scene crops and 92.1% for the scene-scene crop. In the case of OpenImages, the numbers are, respectively, 99.1%

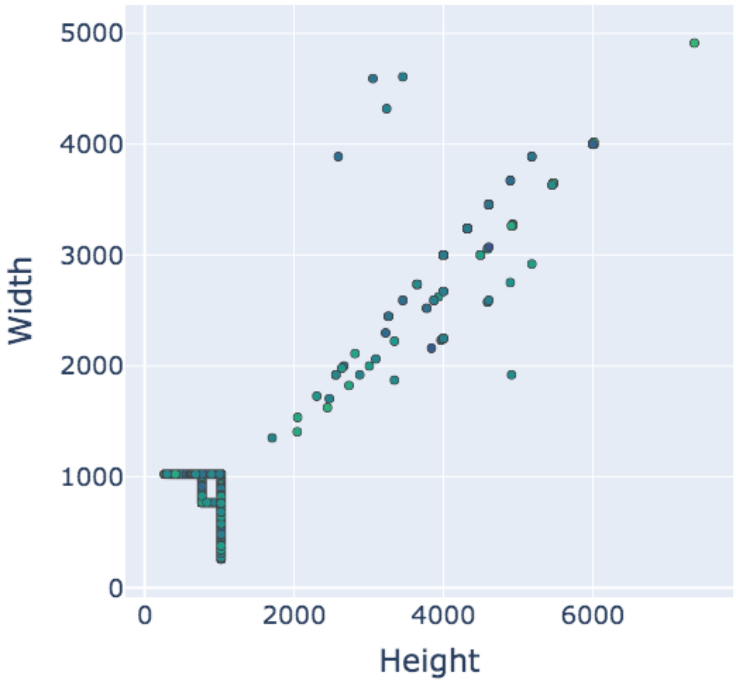

Figure 5: Scatter plot of the height and width of the OpenImages Dataset..

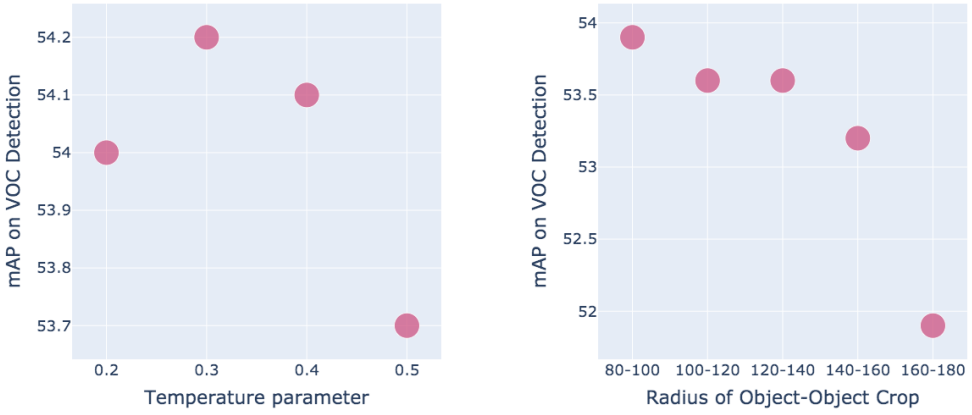

Figure 6: Results on COCO then transferred to VOC. **Left**: Varying the temperature parameter. **Right**: Varying radius for object-object crop.

and 87.3%. This demonstrates that object-scene crops capture information about objects much more than scene-scene crops, especially in the case of OpenImages.

Varying the radius in OpenImages and COCO: We also vary the radius, which is the distance of the second object-object crop in any random direction from the center of the object crop. We vary

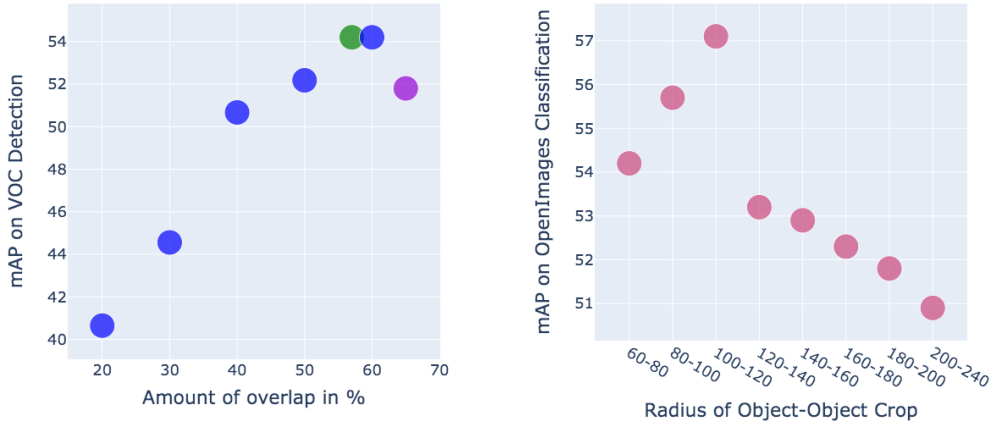

Figure 7: Left: We vary the amount of overlap between object and scene crops, as shown. Optimal VOC detection performance is achieved around 58% overlap, decreasing on either side. The purple dot indicates overlap for scene-scene crops (66%). Right: We vary the radius of the object-object crops and observe a similar phenomenon of a "sweet spot" distance between the object crops at which downstream performance is maximized.

this radius under object-object crop when combining all three type of crops. Fig 7(right) shows that, for OpenImages, as we increase the distance of the two object crops, performance first increases and then decreases, again suggesting a sweet spot. We report similar results for COCO dataset in the supplementary.

Use of scene information: To test whether scene information is used by our models, we pre-train two models on the OpenImages dataset: one with object-object cropping and the other with object-scene cropping. Next, we create a new dataset by cropping ground truth bounding boxes from the OpenImages images. Each resulting image contains exactly one object and minimal scene information. We fine-tune the two models on this dataset, and we find that they perform comparably (object-object fine-tuned gives 37 mAP and object-scene fine-tuned gives 36.4 mAP). However if we fine-tuned instead on the OpenImages dataset, the object-scene model outperforms by close to 4 mAP, thus verifying that object-scene model cropping learns features that rely on both object and scene information.

Varying radius: Fig. 6(right) shows the performance on VOC (pre-training on COCO) for MoCo-v2 as the radius object-object crops is changed. We see that there is a "sweet spot" for the radius: neither too large or too small is optimal.

E Additional Results

Generating positive samples by randomly cropping on Dilated BING boxes: We also try to generate both the positive samples by randomly cropping on the Dilated BING boxes. We find that this method gives us a performance of 54.5 mAP for OpenImages classification. Hence cropping on object boxes and dilated crops are crucial to success of our method.

Varying temperature parameter: We also varied the temperature hyper-parameter for training MoCo-v2 on COCO (transfering to VOC) as seen in Fig. 6(left). We find that temperature of 0.3 performs best in our setting on the COCO dataset.

Varying number of proposals used: We vary the number of proposals from BING as shown in Table 8. We find that using more proposals results in better performance and we tested upto 10 proposals per image, finding a consistent boost.

Model	AP	AP_{50}	AP_{75}
Supervised	56.8	83.2	63.7
BYOL Scene-Scene crop (Grill et al., 2020)	50.0	75.8	54.8
MoCo-v2 Scene-Scene crop (Chen et al., 2020b)	51.8	77.6	55.4
BYOL Obj-Scene crop using BING crops (Ours)	52.5	77.1	58.1
MoCo-v2 Obj-Scene crop using Unsupervised crops (Ours)	**53.9**	**79.8**	**59.8**
MoCo-v2 Obj-Scene crop using BING crops (Ours)	**54.2**	**80.1**	**60.0**
MoCo-v2 - Obj-Obj+Dilate crop ($\delta = 0.1$) (Ours)	**55.0**	**80.9**	**60.7**
Dene-CL (Wang et al., 2021)	56.7	82.5	63.8
Dense-CL with Obj-Obj+Dilate crop (Ours)	**57.6**	**82.5**	**63.8**

Table 6: Object detection results on VOC dataset (COCO pre-training). All models have been pre-trained on COCO and then fine-tuned on VOC. For both MoCo-v2 and BYOL, replacing the default scene crops with object-scene crops results in a consistent improvement.

Model	AP	AP_{50}	AP_{75}
Supervised	56.8	83.2	63.7
Baseline (Scene-Scene) (Chen et al., 2020b)	51.5	79.4	56.1
Ours (Obj-Scene) (Chen et al., 2020b)	53.0	79.3	58.3

Table 7: Object detection results on VOC dataset (OpenImages pre-training). All models have been pre-trained on OpenImages and then fine-tuned on VOC. Replacing the default scene crops with Obj-Scene crops results in a consistent improvement.

Impact of varying scene crop range in the object-scene crop: We show the effect of varying scene crop range in object-scene crop in Table 11. We can see that as we again reach a sweet spot when the lower crop range is 0.2.

Results after 200 epochs training: We show results after training 200 epochs of training in Table 12. We compare these results with 800 epochs and we can see their a good co-relation between results after 200 epochs and after 800 epochs. This could be useful in low compute regime settings, since training 200 epochs is highly indicative of performance after longer training schedules.

F VISUALIZATIONS

We show additional examples of object crops and scene (random) crops in Figures 8, 9, 10, 11, 12. We can see in all these images random scene crops often miss out objects of interest while BING tends to capture them. We also show visual comparison between OpenImages and ImageNet as shown in Fig 13. We can see that ImageNet images are centered and mostly occupy the center part of the image, which is not the case with OpenImages.

Model	Proposals	AP	AP_{50}	AP_{75}
Supervised	-	56.8	83.2	63.7
MoCo-v2 Obj-Scene crop (Chen et al., 2020b)	1	51.9	77.6	55.4
MoCo-v2 Obj-Scene crop (Chen et al., 2020b)	5	53.5	79.4	56.4
MoCo-v2 Obj-Scene crops using BING crop (Ours)	10	**54.2**	**80.1**	**60.0**

Table 8: Varying the number of proposals generated by BING. All models are pre-trained on COCO and then fine-tuned on VOC Object Detection. Increasing the number of proposals provides a consistent boost. We used 10 proposals as this was close to the average number of objects per image in OpenImages.

Model	δ	AP	AP_{50}	AP_{75}
MoCo-v2	0	54.3	77.1	5484
MoCo-v2	0.1	55.1	81.0	60.7
MoCo-v2	0.2	54.5	80.7	60.3
MoCo-v2	0.3	54.2	80.2	60.1

Table 9: Varying dilation i.e δ on the COCO dataset for Obj-Obj+Dilate crop strategy. All models have been pre-trained on COCO and then fine-tuned on VOC Object Detection. $\delta = 0$ corresponds to Obj-Obj cropping and larger δ is very similar to Obj-Scene. A sweet spot exists between the extremes.

Dataset	Obj-Scene (Top-1)	Obj-Obj+Dilate (Top-1)
Aircraft	87.8	88.7
Stanford Cars	89.1	90.7
Caltech-UCSD Birds	87.2	88.8

Table 10: Obj-Obj+Dilate crop pre-training consistently outperforms Obj-Scene for transfer to various downsteam datasets for classification (COCO pre-training).

Model	mAP (Classification)
Supervised OpenImages	66.3
MoCo-v2: Scene Crop Range (Object-Scene) 0.8-1.0	34.2
MoCo-v2: Scene Crop Range (Object-Scene) 0.6-1.0	46.1
MoCo-v2: Scene Crop Range (Object-Scene) 0.4-1.0	54.3
MoCo-v2: Scene Crop Range (Object-Scene) 0.2-1.0	58.1
MoCo-v2: Scene Crop Range (Object-Scene) 0.1-1.0	56.3
MoCo-v2: Scene Crop Range (Object-Scene) 0.05-1.0	53.5

Table 11: Impact of varying scene crop range in the object-scene crop.

Description	epochs	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
MoCo-v2 Scene-Scene crop	200	34.6	53.5	37.0	30.4	50.1	32.3
MoCo-v2 Obj-scene crop (Ours)	200	**35.6**	**54.3**	**38.3**	**31.3**	**51.2**	**33.4**
MoCo-v2	800	38.2	58.9	41.6	34.8	55.3	37.8
MoCo-v2 Obj-Scene crop (Ours)	800	**39.4**	**59.8**	**42.9**	**35.8**	**57.8**	**38.7**

Table 12: Results after pre-training for 200 epochs. These results are highly indicative of results we get after longer training and can be useful for comparison in less compute settings.

Description	AP	AP_{50}	AP_{75}	AP_s	AP_l	AP_m
MoCo-v2	39.8	59.8	43.6	36.1	56.9	38.7
MoCo-v2 (Obj-Obj+Dilate $\delta = 0$) (Ours)	**40.2**	**60.6**	**43.6**	**36.4**	**57.4**	**39.0**
MoCo-v2	57.0	82.2	63.4	-	-	-
MoCo-v2 (Obj-Obj+Dilate $\delta = 0$)	**57.8**	**82.9**	**64.2**	-	-	-

Table 13: Object detection results on COCO (top 2 rows) and VOC (bottom 2 rows). All SSL models have been pre-trained on ImageNet for 200 epochs and then fine-tuned on COCO and VOC.

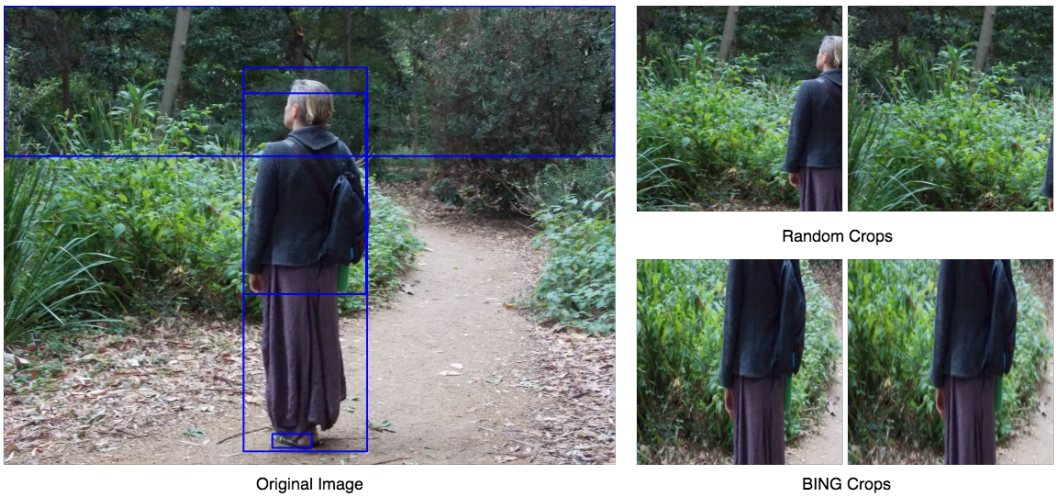

Figure 8: Visualization of BING and random crops.

Figure 9: Visualization of BING and random crops.

G Implementation Details

One additional change that we make to object crops is if the size of the object proposal is small, i.e less than 224×224, we use a larger crop from the original image so that the size of the crop is at least 224×224. To achieve this, we pick the center of the object-crop in the original image and cut a center crop of size 224×224 from original object-crop center. Hence the object-crops for images smaller than 224×224 are not tight bounding boxes but bigger crops with objects in it. Additionaly we find that $delta$ = 0.1 works well for COCO and $delta$ = 0.2 works well for OpenImages.

G.1 Hyperparameters for different methods

We train all the SSL methods for 200 epochs on OpenImages-Subset unless mentioned otherwise.

MoCo-v2: For MoCo-v2 (Chen et al., 2020b) we closely follow the standard hyper-parameters (given in the main paper). We used a learning rate of 0.03 using SGD as our optimizer, weight decay of 10^{-4} and initial learning rate of 0.03. To traine OpenImages-Subset, we trained for 200 epochs and for COCO, following DenseCL (Wang et al., 2021), we trained for 800 epochs.

Figure 10: Visualization of BING and random crops.

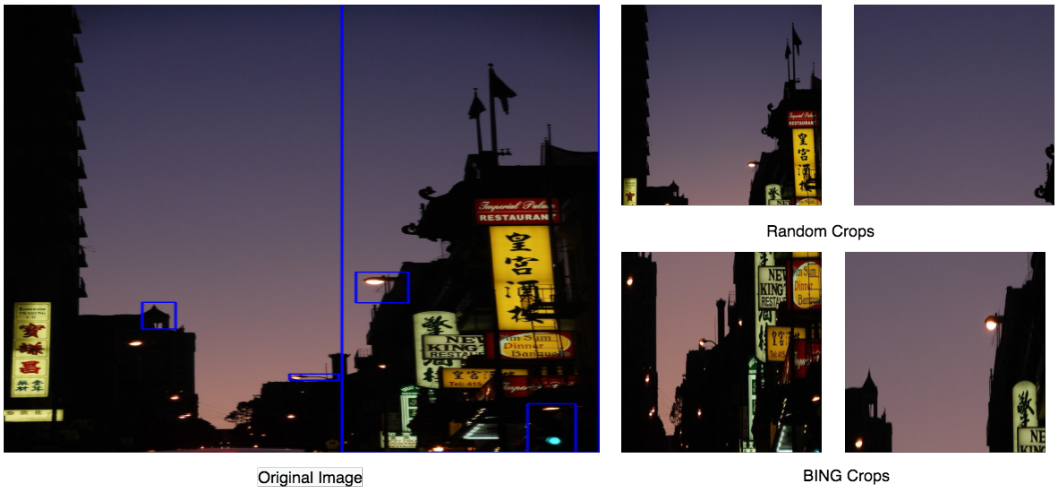

Figure 11: Visualization of BING and random crops.

Supervised Training on OpenImages: For supervised training on OpenImages, we use standard Adam optimizer with learning rate of 10^{-3}, patience of 5 epochs and trained the network for 90 epochs.

CMC: We follow the standard hyperparameters following CMC(Tian et al., 2019): temperature 0.07, momentum 0.5 and learning rate 0.03.

BYOL: For BYOL, we used LARS optimizer with learning rate of 0.2, following the standard hyperparameters listed in (Grill et al., 2020).

SWaV: For SWaV the standard hyperparameters listed in SWaV (Caron et al., 2020) do not work well for OpenImages-Subset (the authors do not show results on this dataset in their paper). So following their FAQ's in the offical github repository (`https://github.com/facebookresearch/swav`) we tried few different hyperparameters. Intially we started with epsilon of 0.05 and then decreased it to 0.03. We also deceased the LR to half and froze the prototypes during the first few iterations. We trained to network for 200 epochs.

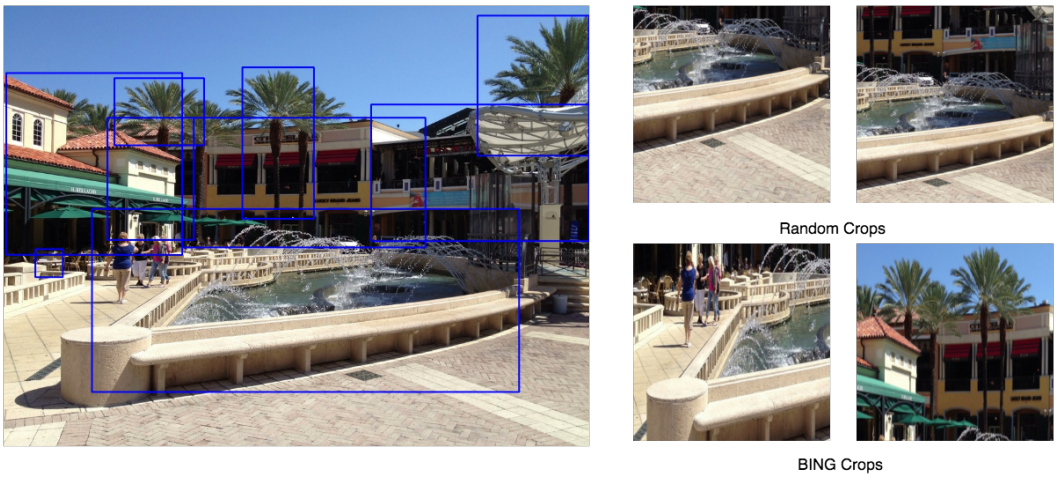

Figure 12: Visualization of BING and random crops.

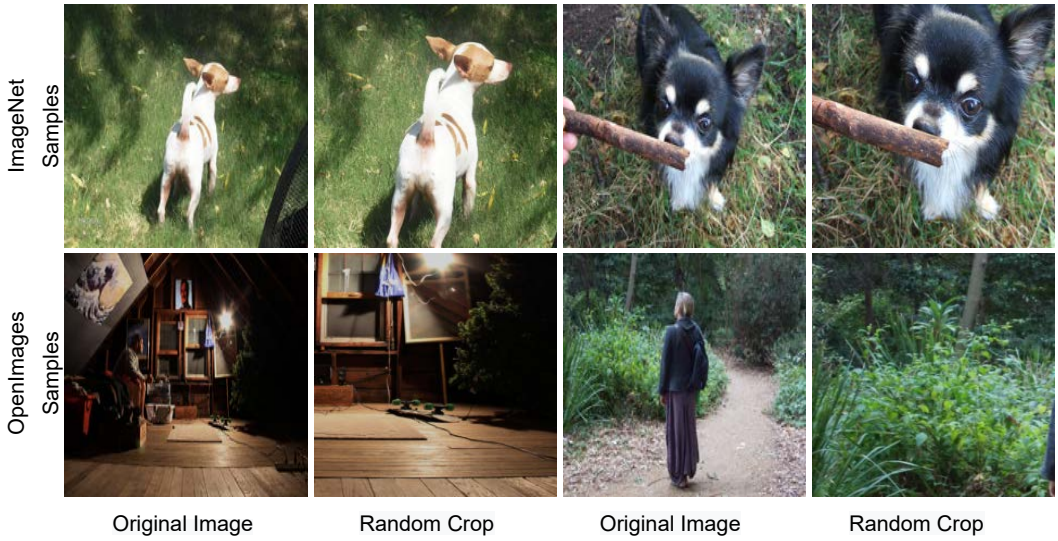

Figure 13: First column: Two samples from OpenImages dataset. Second and Third Columns: Random scene crops; Fourth and fifth columns: Random object crops generated by the BING (Cheng et al., 2014) algorithm. We see that the object crop tends to center on objects in the image that are often missed by the scene crops.

H THEORETICAL MOTIVATION

Our hypothesis is that uniformly random image cropping has a high likelihood of missing relevant objects or having other noisy signals. This is especially true when objects are small relative to the size of the image. This leads to suboptimal representations for multi-object datasets such as OpenImages due to the inclusion of data that is weakly correlated with the objects of interest. To overcome this, we propose using semantic information in the cropping to ensure a higher probability of selecting crops around relevant semantic objects in the scene. While we apply our techniques to several self-supervised training approaches, our intuition is best understood in the context of contrastive learning frameworks.

It has been shown(van den Oord et al., 2018) that objectives such as InfoNCE estimate a lower bound on $I(X;C)$, where I(.;.) is the mutual information, X is the "image signal of interest" (X) and the correlated context (C). InfoNCE approximates the lower bound by maximizing the correlation between encoded image inputs $g_\theta(X)$ and $g_\theta(C)$, while using uncorrelated contexts to prevent code

collapse. Here g_θ is the neural network parameterized by θ, and θ is optimized using stochastic gradient descent based optimization.

$I(X;C)$ can be rewritten as $\int_{x,c} p(x,c)\log(p(x|c)/p(x))dxdc$. The statement for our approach is that *we plug in a new p'(x) that concentrates the probability mass around regions of X that are strongly correlated with artefacts of interest in the downstream tasks*. Intuitively, this ensures that the training procedure for g_θ is more focused on X samples that are strongly correlated with objects of interest. For downstream tasks such as object recognition and detection, we can approximate "strong correlation with artefacts of interest" in a crop with the presence of objects in that crop. Therefore our approach can be interpreted as plugging in a new distribution p'(x) in the place of p(x) in the following InfoNCE equation.

$$I(X;C) = \int_{x,c} p(x,c)\log\frac{p(x|c)}{p(x)}dxdc \geq \int_{(x,c,c'_{1:k})} -\log\frac{f(x,c)}{f(x,c)+\sum_k f(x,c'_k)}d(x,c,c'_{1:k}) \tag{1}$$

Here f is the function used to compute the density ratio (van den Oord et al., 2018) and $c'_{1:k}$ are the k negative samples. Our proposed p'(x) should produce a set of samples that are more enriched with object-centric crops. To produce this enriched sample set, we use three object proposal models: BING (Cheng et al., 2014) and Edge-Boxes (Zitnick & Dollár, 2014), both of which are trained with boxes from Pascal-VOC; and an unsupervised object proposal method (Vo et al., 2019).

In the later sections of this paper, we provide experimental result that validate the hypothesis that our plugged in p'(x) distribution leads to improved downstream performance. Additionally, our schema also shows improvement over self-supervised approaches, such as BYOL and self-EMD, which cannot be decomposed into a gradient descent procedure for an objective function. Our approach also seems to complement methods like self-EMD that are already designed to improve performance on non-iconic image datasets such as COCO.

I PSEUDO-CODE USING MOCO

REFERENCES

Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019.

Tiffany Tianhui Cai, Jonathan Frankle, David J. Schwab, and Ari S. Morcos. Are all negatives created equal in contrastive instance discrimination?, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 9912–9924. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/70feb62b69f16e0238f741fab228fec2-Paper.pdf.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020a.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020b.

Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3286–3293, 2014.

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

`bmm`: batch matrix multiplication; `mm`: matrix multiplication; `cat`: concatenation.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature
max_objects = 10
f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   view_1 = aug(x) # a randomly augmented version
   frame_num = random(0,max_objects)
   view_2 = BING(x)[frame_num]
   view_2 = aug(view_2) # BING object augmented version

   if randn() > 0:
      x_q = view_1
      x_k = view_2
   else:
      x_q = view_2
      x_k = view_1

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 8765–8775. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/63c3ddcc7b23daa1e42dc41f9a44a873-Paper.pdf.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *ArXiv*, abs/1805.09501, 2018.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Terrance DeVries and Graham W. Taylor. Dataset augmentation in feature space, 2017.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338, 2010.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Revisiting contrastive methods for unsupervised learning of visual representations, 2021.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words, 2020.

Priya Goyal, Mathilde Caron, Benjamin Lefaudeux, 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, 2021.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

Ryuichiro Hataya, Jan Zdenek, Kazuki Yoshizoe, and Hideki Nakayama. Faster autoaugment: Learning augmentation strategies using backpropagation, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019.

Geoffrey E. Hinton. How to represent part-whole hierarchies in a neural network. *ArXiv*, abs/2102.12627, 2021.

Olivier J. Hénaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection, 2021.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning, 2020.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4. *International Journal of Computer Vision*, pp. 1–26, 2020.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 840–849, 2017.

Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S. Davis. An analysis of pre-training on object detection, 2019.

Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment, 2019.

Tsung-Yi Lin, M. Maire, Serge J. Belongie, James Hays, P. Perona, D. Ramanan, Piotr Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

Songtao Liu, Zeming Li, and Jian Sun. Self-emd: Self-supervised object detection without imagenet, 2021a.

Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised learning: Generative or contrastive, 2021b.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations, 2019.

Alejandro Newell and Jia Deng. How useful is self-supervised pretraining for visual tasks? *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020. doi: 10.1109/cvpr42600.2020.00737. URL http://dx.doi.org/10.1109/CVPR42600.2020.00737.

Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2536–2544, 2016.

Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases, 2020.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

Pierre H Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel Smith, Soham De, Razvan Pascanu, Bilal Piot, et al. Byol works even without batch statistics. *arXiv preprint arXiv:2010.10241*, 2020.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2016.

Ramprasaath R Selvaraju, Karan Desai, Justin Johnson, and Nikhil Naik. Casting your model: Learning to localize improves self-supervised representations. *arXiv preprint arXiv:2012.04630*, 2020.

Patrice Simard, David Steinkraus, and John Platt. Best practices for convolutional neural networks applied to visual document analysis. pp. 958–962, 01 2003. doi: 10.1109/ICDAR.2003.1227801.

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

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6827–6839. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/4c2e5eaae9152079b9e95845750bb9ab-Paper.pdf.

Yonglong Tian, Olivier J. Henaff, and Aaron van den Oord. Divide and contrast: Self-supervised learning from uncurated data, 2021.

Christopher Tosh, Akshay Krishnamurthy, and Daniel Hsu. Contrastive learning, multi-view redundancy, and linear models, 2021.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision, 2017.

Huy V. Vo, Francis Bach, Minsu Cho, Kai Han, Yann LeCun, Patrick Perez, and Jean Ponce. Unsupervised image matching and object discovery as optimization, 2019.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere, 2020.

Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training, 2021.

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

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018. doi: 10.1109/CVPR.2018.00393.

Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020.

Tete Xiao, Xiaolong Wang, Alexei A. Efros, and Trevor Darrell. What should not be contrastive in contrastive learning, 2021.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16684–16693, 2021.

Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3987–3996, 2021.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.

C. L. Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. Rethinking pre-training and self-training, 2020.