Self-supervised Learning with Local Contrastive Loss for Detection and Semantic Segmentation

Ashraful Islam*
Nvidia
aislam@nvidia.com

Ben Lundell, Harpreet Sawhney, Sudipta N. Sinha
Microsoft Mixed Reality
{Benjamin.Lundell,Harpreet.Sawhney,Sudipta.Sinha}@microsoft.com

Peter Morales
io.peter.morales@gmail.com

Richard J. Radke
Rensselaer Polytechnic Institute
rjradke@rpi.edu

Abstract

We present a self-supervised learning (SSL) method suitable for semi-global tasks such as object detection and semantic segmentation. We enforce local consistency between self-learned features that represent corresponding image locations of transformed versions of the same image, by minimizing a pixel-level local contrastive (LC) loss during training. LC-loss can be added to existing self-supervised learning methods with minimal overhead. We evaluate our SSL approach on two downstream tasks – object detection and semantic segmentation, using COCO, PASCAL VOC, and CityScapes datasets. Our method outperforms the existing state-of-the-art SSL approaches by 1.9% on COCO object detection, 1.4% on PASCAL VOC detection, and 0.6% on CityScapes segmentation.

1. Introduction

Self-supervised learning (SSL) approaches learn generic feature representations from data in the absence of any external supervision. These approaches often solve an *instance discrimination* pretext task in which multiple transformations of the same image are required to generate similar learned features. Recent SSL methods have shown remarkable promise in *global* tasks such as classifying images by training simple classifiers on the features learned via instance discrimination [1, 2, 4, 17, 18]. However, global feature-learning SSL approaches do not explicitly retain spatial information thus rendering them ill-suited for semi-global tasks such as object detection, and instance and semantic segmentation [37, 43].

This work focuses on extending SSL to incorporate spatial locality by using a local contrastive (LC) loss function at a dense and fine-grained pixel level. The main idea is illustrated in Figure 1. Specifically, we encourage corresponding *local pixels* in the two transformed images to produce similar features. The true pixel correspondences are known since the input image pairs are generated by applying two distinct transformations to a single image. Note that this approach can be used along side *any* conventional global SSL objective with minimal overhead.

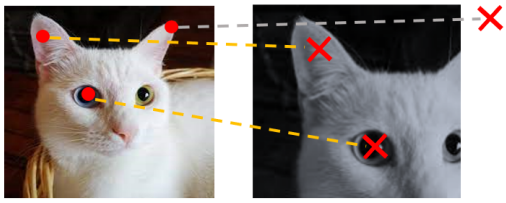


Figure 1: Our framework encourages local regions of two transformed images to learn similar features. The first image (left) uses color augmentation only, and another image (right) employs both spatial (random resize crop) and color transformations. With known corresponding pixels a consistency loss enforces maximal similarity between the corresponding learned features.

We evaluate the impact of LC-loss on several downstream tasks, namely object detection, and instance and semantic segmentation and report promising improvements over previous spatially-aware SSL methods [37, 38, 41, 43] on Pascal VOC, COCO and Cityscapes datasets.

Contributions. Our main contribution is in demonstrating that adding a pixel-level contrastive loss to the BYOL [17] training procedure for the instance discrimination pretext task is sufficient to produce excellent results on many downstream dense prediction tasks. A similar, pixel-level contrastive loss formulation was presented in PixPro [43], but a more complicated pixel-to-propagation consistency pretext task was required to achieve state of the art results in dense prediction tasks. We show that no additional pre-text is necessary, and our simpler local contrastive loss formula-

*This work was done while the author was an intern at Microsoft.

tion achieves superior performance. Specifically, our key technical contributions are: **(1)** a simple framework that computes local contrastive loss (LC-loss) to make the corresponding pixels of two augmented versions of the same image similar, that can be added to any self-supervised learning method, such as BYOL; and **(2)** state-of-the-art transfer learning results in several dense labeling tasks. Using ResNet-50 backbones pretrained on ImageNet, our BYOL variant achieves 40.6 AP for COCO object detection (+1.9 vs SSL SOTA[1]), 60.1 AP for VOC object detection (+1.4 vs SOTA), 72.1 mIoU for VOC segmentation (+1 vs SOTA), and 77.8 mIoU for CityScapes segmentation (+0.6 vs SOTA) for full-network fine-tuning setting. Our performance improvement is even more significant in the frozen backbone setting discussed in Section 4.4.

2. Related Work

Self-supervised learning (SSL). In self-supervisd learning, the supervisory signal is automatically generated from a pair of input images and a pretext task. The input pair is generated by applying two distinct transformations. The pretext task is comparison between the learned representations of each pair of input images . Various pretext tasks have been explored, such as, patch position [12], image colorization [45], image inpainting [31], rotation [15], and predictive coding [24]. The pretext task that has shown the most promise is the instance discrimination task, in which each image is considered as a single class. SimCLR [4], the first to propose this pretext task, adopts contrastive learning in which features from augmented versions of the same image are made closer in the feature space than all the other images in a mini-batch. SimCLR requires a large mini-batch to make contrastive learning feasible. MoCo [18] solves the issue of large batch size using a momentum queue and moving average encoder. Despite impressive results in image classification tasks, contrastive learning requires careful handling of negative pairs. Recent approaches like BYOL [17], SwaV [1], and DINO [2] do not require any negative pairs or a memory bank. They also achieve impressive performance on the ImageNet1k linear evaluation task and downstream image classification-based transfer learning tasks.

SSL for Detection and Segmentation. For dense prediction tasks, SSL methods use an ImageNet-pretrained backbone within a larger architecture designed for a detection or segmentation task [29, 33, 40], and fine tune the network on the downstream task dataset. He et al. [19]reported that ImageNet pretrained models might be less helpful if the target task is localization sensitive than classification . One potential solution is to increase the target dataset size [19], or to impose local consistency during the ImageNet pretraining. We adopt the latter strategy.

Broadly, there are two approaches in the literature to ensuring local consistency during self-supervised pretraining: pixel-based and region-based. In region-based methods, first region proposals are generated - either during the self-supervised training [38, 34, 44], or before training starts [41], and then local consistency is applied between pooled features of the proposed regions. Our approach is pixel-based; local consistency is applied between the local features for corresponding pixels of transformed versions of the same image [37, 43, 42].

DenseCL [37] proposed dense contrastive learning for self-supervised visual pretraining. It follows the MoCo [18] framework to formulate the dense loss. However, DenseCL does not use known pixel-correspondences to generate positive pairs of local feature between two images. Instead, it extracts the correspondence across views. This creates a chicken-and-egg problem where DenseCL first requires learning a good feature representation to generate correct correspondences.

Our local contrastive loss is more similar to the PixContrast loss in the PixPro paper [43]. Given an image I, both methods operate on two distinct transforms $J_1 = \mathcal{T}_1(I)$ and $J_2 = \mathcal{T}_2(I)$ of I to produce two low-resolution, spatial feature maps. Both methods use a contrastive loss using pixel correspondences to generate positive and negative samples. The difference comes in how these samples are selected.

In PixContrast, pixels in the low-resolution feature map are warped back to the original image space using $\mathcal{T}_1^{-1}$ and $\mathcal{T}_1^{-1}$. Positive samples are then determined by all pairs of pixels that are sufficiently close after warping. Our method generates positive samples using the correspondences in J_1 and J_2 derived directly from $\mathcal{T}_1$ and $\mathcal{T}_2$. While similar to ours, the method proposed for PixContrast does not work. Instead, an additional pixel-propagation module is introduced (PixPro) to measure the feature similarity between corresponding pixels. We show that no pixel propagation or feature warping is required in our simpler formulation.

In summary, our framework does not require: (1) an encoder-decoder architecture for local correspondence loss [32]; (2) contrastive learning that needs carefully tuned negative pairs [32, 37, 41]; (3) a good local feature extractor to find local feature correspondences [37]; and (4) an additional propagation module to measure local contrastive loss [43]. A simple local correspondence loss obtained from matching pixel pairs achieves state-of-the-art results in detection and segmentation tasks.

3. Methodology

Figure 2 depicts our LC-loss framework. We use BYOL for the global self-supervised loss function, and apply a local contrastive loss on dense feature representations obtained from the backbone networks. We adopted BYOL

[1] State-Of-The-Art

framework because it achieves higher performance than contrastive learning without using any negative pairs, and it is more resilient to changes in hyper-parameters like batch size and image transformations. In the following, we briefly describe the BYOL framework, and introduce our approach.

Instance Discrimination from Global Features. BYOL consists of two neural networks: an online network with parameters θ and a target network with parameters ξ. The online network has three sub-networks: an encoder, a projector and a predictor. The target network has the same architecture as the online network except the predictor. The online network is updated by gradient descent, and the parameters of the target network are exponential moving averages of the parameters of the online network. Given an input image I, two transformed views I_c and I_t of I are obtained. One view I_c is passed through the online network with parameters θ to obtain local features F_θ, average pooled encoder output $\mathbf{f}_\theta$, a projection $\mathbf{g}_\theta$ and prediction $\mathbf{q}_\theta$. View I_t is passed through the target network with parameters ξ to obtain local features F_ξ, average pooled encoder output f_ξ and a projection $\mathbf{g}_\xi$. Check Fig. 2. There is no predictor in the target network. This asymmetric design is adopted to prevent collapse during self-supervised training [17]. We follow the original BYOL network to set the dimension of encoder outputs (2048 dim), projections (4096 dim), and predictions (256 dim). The global self-supervised loss function for a single input is defined as:

$$\mathcal{L}_{\mathrm{G}} = 2 - 2\frac{\mathbf{q}_\theta^T \mathbf{g}_\xi}{||\mathbf{q}_\theta||_2 ||\mathbf{g}_\xi||_2} \tag{1}$$

where $\mathbf{q}_\theta$ and $\mathbf{g}_\xi$ are learned global representations for two transformed images that are forced to be similar under cosine similarity.

Local Contrastive Loss. Given two transformed versions of the same image, namely I_c and I_t, let an image point p_c in I_c, correspond to another image point p_t in I_t. We can determine p_t for every p_c given the known image transformations. The correspondence map $C_{p_c} \in \mathbb{R}^{H \times W}$ for source point p_c is calculated, where $C_{p_c}(p_l)$ denotes the similarity score between p_c of I_c and every pixel p_l of I_t. We define $C_{p_c}(p_l)$ to be the similarity score that p_l is the corresponding pixel of p_c. As we know p_t is the actual corresponding pixel, we want $C_{p_c}(p_t)$ to be maximized. The local loss for p_c is the negative log likelihood at p_t which encourages maximizing the likelihood estimate for the target locations $-\log C_{p_c}(p_t)$.

We now describe how this is incorporated in our pipeline. We employ learned feature-level correspondence as a measure of pixel-level correspondence. Given an image I, we apply image transformation $\mathcal{T}_{\mathrm{c}}$ to I to obtain I_{c}. $\mathcal{T}_{\mathrm{c}}$ contains strong color transformations (for example, Gaussian blur, solarization, color distortion), but does not include spatial transformations like image flipping or random crop. We apply normal resize operation to resize the image to $H \times W$ shape. Another transformation $\mathcal{T}_{\mathrm{sc}}$ is applied to I to obtain I_{sc}, where $\mathcal{T}_{\mathrm{sc}}$ contains both spatial and color transformations. We obtain dense feature representations of I_{sc} from the backbone of the online network, and denote it as $F_\theta \in \mathbb{R}^{h \times w}$ where $h = {}^{H}/_{p}$ and $w = {}^{W}/_{p}$ with p as the stride size for the feature representation. We get a similar feature representation $F_\xi \in \mathbb{R}^{h \times w}$ by passing I_{c} through the target network.

Next we select $h \times w$ image points on a 2D uniform grid in I_{c}. Each point p_{c} of I_{c} has the local feature representation $F_\xi(p_{\mathrm{c}})$ obtained from the target network. For the point p_{c}, there is a corresponding point p_{sc} in I_{sc}. Note that because of random crop and flipping in $\mathcal{T}_{\mathrm{sc}}$, the feature $F_\theta(p_{\mathrm{sc}})$ may not be at an integer pixel coordinate p_{sc}. Instead of adopting an expensive high-dimensional warping of F_θ to obtain corresponding features, we compute the negative log-likelihood from correspondence map and resample the negative log-likelihood using bilinear interpolation to deal 2D points with subpixel coordinates, which is described below [14].

We compute a dense correspondence map $C' \in \mathbb{R}^{(h \times w) \times (h \times w)}$ between I_{c} and I_{sc} as,

$$C'(p_c, p_t) = \frac{F_\xi(p_c)^T F_\theta(p_{\mathrm{sc}})}{||F_\xi(p_c)||_2 ||F_\theta(p_{\mathrm{sc}})||_2} \tag{2}$$

where p_c and p_t are image points from I_{c} and I_{sc} respectively. Then, we calculate the negative log-likelihood

$$\mathrm{NLL}(p_{\mathrm{c}}, p_{\mathrm{t}}) = -\log \frac{\exp(C'(p_c, p_{\mathrm{sc}})/\tau)}{\sum_{k \in \Omega_{sc}} (\exp(C'(p_c, p_k)/\tau))} \tag{3}$$

where Ω are set of locations in the I_{sc}. If p_{sc} is not an integer location, we obtain the negative log-likelihood $\mathrm{NLL}(p_{\mathrm{c}}, p_{\mathrm{sc}})$ by bilinearly interpolating $\mathrm{NLL}(p_{\mathrm{c}}, :)$. The contrastive loss, LC-loss, is defined as

$$\mathcal{L}_{\mathrm{LC}} = \frac{1}{|\mathcal{P}|} \sum_{(p_{\mathrm{c}}, p_{\mathrm{sc}}) \in \mathcal{P}} \mathrm{NLL}(p_{\mathrm{c}}, p_{\mathrm{sc}}) \tag{4}$$

where $\mathcal{P}$ contains all corresponding pairs $\{(p_{\mathrm{c}}, p_{\mathrm{sc}})\}$ for I_{c} and I_{sc} such that p_{sc} does not fall outside of the boundary of I_{sc}, and $|\mathcal{P}| \leq h \times w$.

Our total loss is defined as:

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{\mathrm{G}} + \alpha \mathcal{L}_{\mathrm{LC}} \tag{5}$$

where α is the multiplicative factor that balances the two loss components. See Section 4.5 for a study on the impact of α.

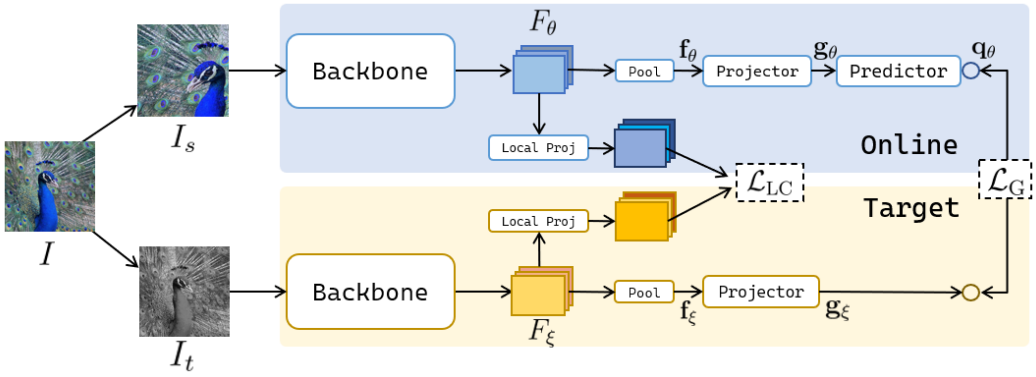


Figure 2: **Proposed Framework.** We use BYOL as the self-supervised learning framework. BYOL consists of an online network with parameters θ and a target network with parameters ξ which are the exponential moving average of θ. Given an image, we create two transformed versions. We apply mean squared error loss between the L2-normalized global feature representations from the online and target networks.. We also calculate a local contrastive loss from the dense feature representations of the image pairs.

4. Experiments

4.1. Implementation Details

Pretraining Setup. We use ResNet-50 [21] as the backbone network and BYOL [17] as the self-supervision architecture. We use identical architectures for projection and prediction networks as in BYOL. For extracting local features, we add a local projection branch on the online network and another branch of similar architecture on the target network. As with the global branches, only the online local projection branch is updated through optimization while the target local projection branch is the exponential moving average of the online one. The local projection branch consists of two convolution layers. The first convolution layer consists of a 1×1 convolution kernel with input dimension 2048 and target dimension 2048 following by a BatchNorm layer. The second convolution layer contains a 1×1 kernel with output dimension 256. The input to the local projection branch is the local feature representation from the final stage of ResNet-50 (before the global average pooling layer). For image transformations during pretraining, following BYOL [17], we use random resize crop (resize to 224×224), random horizontal flip, color distortion, blurring, and solarization. We do not apply random crop for the image that is used to obtain local contrastive loss.

Dataset. We use the ImageNet [11] dataset for pretraining the networks. ImageNet contains $\sim$1.28M training images, mostly with a single foreground object.

Optimization. The default model is trained with 400 epochs if not specified in the results. See Sec. 4.5 for details on the effect of pretraining epoch to the transfer performance. The LARS optimizer is used with a base learning rate of 0.3 for batch-size 256, momentum 0.9, weight-decay 1e-6, and with cosine learning rate decay schedule for with learning rate warm-up for 10 epochs. We use 16 GPUs with 256 batch-size on each GPU, hence, the effective batch-size is 4096. We linearly scale the learning rate with the effective batch size. The weight parameter α is set to 0.1 (Eqn. 5). For the momentum encoder, the momentum value starts from 0.996 and ends at 1. We use 16 bit mixed-precision during pre-training.

4.2. Results on Object Detection and Instance Segmentation

We use Detectron2 framework [39] for evaluation of downstream object detection and segmentation results on COCO and PASCAL VOC dataset.

COCO Object Detection. For object detection on COCO, we adopt RetinaNet [27] following [38, 34, 42]. We finetune all layers with Sync BatchNorm for 90k iterations on COCO `train2017` set and evaluate on COCO `val2017` set. Table 1 shows object detection results on COCO for our method and other approaches in the literature with full-network finetuning. Note that ReSim, SoCo, and SCRL use region proposal networks during pretraining on ImageNet, hence, these approaches are not exactly comparable to ours. Our model is more similar with methods like DetCo, DenseCL, and PixPro. We achieve 40.6 AP for object detection tasks outperforming the second best method PixPro [43] by a significant 1.9%. We also report results on COCO detection using Mask R-CNN + FPN. We again outperform PixPro (our mAP is higher by 1.4), when using Mask R-CNN + FPN for the detector.

COCO Instance Segmentation. We use the Mask-RCNN framework [20] with ResNet50-FPN backbone. We follow the $1\times$ schedule. Table 1 depicts that we achieve 38.3% AP for COCO instance segmentation, which is comparable with the SoCo [38]. Note that SoCo performs selective search on the input image to find object proposals, and uses additional feature pyramid networks during pre-training.

Method	Pretrain Epochs	COCO Object Detection RetinaNet + FPN AP^b	AP^b_{50}	AP^b_{75}	COCO Object Detection Mask-RCNN + FPN AP^b	AP^b_{50}	AP^b_{75}	COCO Instance Segmentation Mask-RCNN + FPN AP^{mk}	AP^{mk}_{50}	AP^{mk}_{75}
Supervised [21]	90	37.7	57.2	40.4	38.9	59.6	42.7	35.4	56.5	38.1
Moco v2 [6]	200	37.3	56.2	40.4	40.4	60.2	44.2	36.4	57.2	38.9
BYOL [17]	300	35.4	54.7	37.4	40.4	61.6	44.1	37.2	58.8	39.8
DetCo [42]	800	38.4	57.8	41.2	40.1	61.0	43.9	36.4	58.0	38.9
ReSim-FPN [41]	200	38.6	57.6	41.6	39.8	60.2	43.5	36.0	57.1	38.6
SCRL [34]	800	39.0	58.7	41.9	-	-	-	37.7	59.6	40.7
SoCo [38]	400	38.3	57.2	41.2	**43.0**	63.3	47.1	38.2	60.2	41.0
DenseCL [37]	200	37.6	56.6	40.2	40.3	59.9	44.3	36.4	57.0	39.2
PixPro [43]	400	38.7	57.5	41.6	41.4	61.6	45.4	-	-	-
OURS	400	**40.6**	60.4	43.6	42.5	62.9	46.7	**38.3**	60.0	41.1

Table 1: **Main Results.** We use faster-RCNN with RetinaNet for COCO object detection, Mask-RCNN with FPN for COCO instance segmentation, Faster RCNN with FPN for VOC object detection.

PASCAL VOC Object Detection [13]. We use the Faster-RCNN [33] object detector with ResNet50-FPN backbone following [34]. For training, we use images from both `trainval07+12` sets and we evaluate only on the VOC07 `test07` set. We use the pre-trained checkpoints released by the authors for the backbone network, and fine tune the full networks on the VOC dataset. Table 2 shows that we achieve 60.1 AP for VOC detection. Our method improves mean AP by a significant 3.2% over baseline BYOL, and outperforms the current SOTA PixPro by 1.4% AP. The improvement is even more significant in AP75, where we outperform BYOL by 3.6% and PixPro by 1.9%.

Method	Pretrain Epochs	PASCAL VOC Object Detection FRCNN + FPN AP^b	AP^b_{50}	AP^b_{75}
Supervised [21]	90	53.2	81.7	58.2
BYOL [17]	300	55.0	83.1	61.1
SCRL [34]	800	57.2	83.8	63.9
DenseCL† [37]	200	56.6	81.8	62.9
PixPro† [43]	400	58.7	82.9	65.9
OURS	400	**60.1**	**84.2**	**67.8**

Table 2: **Main Results.** We use Faster RCNN with FPN for VOC object detection. Supervised and BYOL results are from [34]. (†): We use pre-trained checkpoint released by the authors and fine tune on the VOC dataset.

4.3. Results on Semantic Segmentation

We show semantic segmentation evaluation in Table 3 on PASCAL VOC and CityScapes [10] datasets for both fine tuning and frozen backbone settings. We use FCN backbone [29] following the settings in `mmsegmentation` [9].

PASCAL VOC Segmentation. We train on VOC `train-aug2012` set for 20k iterations and evaluate on val2012 set. Table 3 shows that on the VOC2007 test set, our method yields 72.1% mIoU outperforming BYOL by a 7.7% and PixPro by 1% mIoU.

Cityscapes Segmentation. CityScapes [10] contains images from urban street scenes. Table 3 shows that for fine tuning setting our approach yield 77.8% AP which is 6.2% mIoU improvement over BYOL and 0.6% improvement over PixPro.

Method	Pretrain Epochs	VOC mIoU	CityScapes mIoU
Scratch	-	40.7	63.5
Supervised	90	67.7	74.6
Moco v2	200	67.5	74.5
BYOL	300	63.3	71.6
DenseCL	200	69.4	69.4
PixPro†	400	71.1	77.2
OURS	400	**72.1**	**77.8**

Table 3: **Evaluation on Semantic Segmentation using FCN ResNet-50 network on PASCAL VOC and CityScapes dataset.** (†): We use pretrained checkpoint released by the authors and fine-tune the full networks on the VOC dataset. All other scores are obtained from the respective papers.

4.4. Analysis

Frozen Backbone Analysis. We also report detection and segmentation results for frozen backbone following [16, 23, 41]. Training a linear classifier on a frozen backbone is a standard approach to evaluate self-supervised representation quality for image classification [1, 5, 17, 18]. We adopt the standard strategy in 'frozen backbone' setting where we freeze the pre-trained ResNet50 backbone

and only fine tune the remaining layers. Frozen backbone might be an ideal evaluation strategy because fine tuning the full network evaluates quality of representations along with initialization and optimization, whereas frozen backbone evaluates mostly the representation quality of the backbone [16, 41].

For frozen backbone (Table 4), we achieve 30.5% AP outperforming PixPro by 2.8% AP on COCO object detection, 55.1% AP for VOC detection outperforming PixPro by 1.6% and BYOL by 2.7%. We achieve 63.4% mIoU on VOC semantic segmentation, which is *more than the score achieved by BYOL in finetuning setting (63.3% mIoU)*. We also outperform PixPro by a significant 2.9% mIoU. On CityScapes semantic segmentation, We achieve 60.7% mIoU which improves upon BYOL by 5.1% and PixPro by 2.5%.

Efficient Pre-training. In Table 5, we report results of VOC object detection with FasterRCNN-FPN, COCO object detection from MaskRCNN-FPN, VOC and CityScapes segmentation from FCN for BYOL and OURS pre-trained with different epochs. Results reveal that *our model pre-trained with 200 epochs and with training image size 160 can achieve better results than BYOL pre-trained with 1000 epochs saving 5.3× computational resource*. Even our 100-epoch pre-trained model seems to be comparable with 1000-epoch pre-trained BYOL model. *This validates efficacy of our local loss during self-supervised pre-training*.

Importance of Local Contrast. In Table 6, we show relative performance of our local contrastive loss against non-contrastive BYOL-type loss. In, 'BYOL+Local MSE loss', we apply the same L2-normalized MSE local loss as the global loss in BYOL. Models are trained for 200 epochs on the ImageNet dataset. We report the average AP scores for VOC detection, COCO detection with Mask-RCNN, and CityScapes segmentation, which shows that our approach of calculating local consistency using contrastive loss works better than non-contrastive BYOL-type local loss.

Few-shot Image Classification. Since global and local losses appear to be complementary to each other, we ascertain if our method hurts the image classification performance for transfer learning. We use our pre-trained models as fixed feature extractors, and perform 5-way 5-shot few-shot learning on 7 datasets from diverse domains using a logistic regression classifier. Table 7 reports the 5-shot top-1 accuracy for the 7 diverse datasets. Table 7 reveals that OURS shows the best performance on average among the self-supervised models that use local consistency. OURS outperforms PixPro by 2.4% top-1 accuracy on average; the minor fluctuation can be attributed to random noise.

Transfer to Other Vision Tasks. Even though we mainly evaluted on detection and segmentation, we also show results for **keypoint estimation** a task that might benefit from models trained with local consistency. We use Mask-RCNN (keypoint version) with ResNet50 FPN network to evaluate keypoint estimation. We fine tune on COCO `train 2017` for 90k iterations. Table 8 shows that our method outperforms all other approaches in keypoint estimation task.

Detection on Mini COCO. As the full COCO dataset contains extensive annotated images for supervision, it might not always reveal the generalization ability of the network [19]. We also report results for object detection on smaller versions of COCO training set in Table 9. We report results when only 5% and 10% of the images (randomly sampled) are used for fine tuning the mask-RCNN with FPN network with 1× schedule. The evaluation is performed on the full `val2017` set. For the 5% setting, our method outperforms BYOL by 1.4% AP for the ImageNet pretrained models. For the 10% setting, our method achieves improvement over BYOL by 1.8% AP.

Generalization to other SSL methods. In the Appendix, we show results of our approach applied on other SSL approaches (e.g., DINO), where we also show consistent improvement over baseline methods.

4.5. Ablation Studies

Effect of Pretraining Epochs. Figure 3a reports object detection performance on PASCAL VOC with faster-RCNN-FPN and MS-COCO with Mask-RCNN-FPN for different numbers of pre-training epochs. The models are pre-trained on the ImageNet training set. Longer training generally results in better downstream object detection performance. For example, for the 100 epoch pre-trained model, the AP is 39.8%, whereas for the 600 epoch pre-trained model it improves to 42.8% for COCO evaluation. Upswing is also observed for PASCAL VOC object detection.

Ablation on Loss Weight α. Figure 3b reports the AP for object detection on PASCAL VOC for different values of the weight parameter α. The models are pre-trained on the ImageNet dataset for 200 epochs with training image size of 160 for faster training. α balances the weight between the global and local loss functions. For $\alpha = 0.05$, the mean AP is 58.4%. We get a slightly better performance with $\alpha = 0.1$ (58.9%) and $\alpha = 0.3$ (59.0%). The performance degrades a little when α is increased to $\alpha = 0.7$ (57.5%). Results reveal the best performance is achieved when we use both global and local loss functions, and a proper balance between them ensures better downstream performance.

4.6. Qualitative Analysis

Correspondence Visualization. In Figure 4, we show visual examples of correspondence from our model. For two transformed images I_1 and I_2, we extract feature representations F_1 and F_2. For each feature in F_1, the corresponding feature in F_2 is calculated based on maximum cosine sim-

Method	PASCAL VOC OD			COCO OD			VOC SS	Cityscapes SS
	AP^b	AP^b_{50}	AP^b_{75}	AP^b	AP^b_{50}	AP^b_{75}	mIoU	mIoU
Supervised	50.7	80.4	55.1	30.3	50.0	31.3	56.6	55.7
BYOL	52.4	81.1	57.5	30.2	49.1	31.5	55.7	55.6
DenseCL	50.9	79.9	55.0	25.5	43.6	25.8	63.0	58.5
PixPro	53.5	80.4	59.7	27.7	44.6	29.1	60.3	58.2
OURS	**55.1**	**82.6**	**61.7**	**30.5**	**49.8**	**31.7**	**63.4**	**60.7**

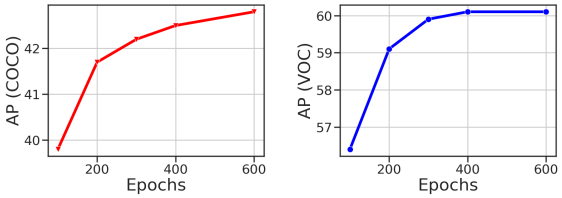
Table 4: **Frozen backbone evaluation.** We freeze the ResNet-50 backbone and finetune the other layers (RPN, FPN, classifier networks, regression layers, etc.). We use faster-RCNN with FPN for PASCAL VOC object detection (OD), RetinaNet-RCNN for COCO detection (OD), and FCN network for VOC and CityScapes segmentation (SS). For this experiment, we use publicly available checkpoint for the backbone networks and evaluate on the downstream tasks.

Method	Pretrain Epochs	Pretrain Im-size	Pretrain time	VOC AP^b	COCO AP^b	VOC mIoU	Cityscapes mIoU
BYOL	300	224	$\times$1.6	56.9	40.4	63.3	71.6
BYOL	1000	224	$\times$5.3	57.0	40.9	69.0	73.4
OURS	100	224	$\times$0.8	58.2	40.9	68.4	76.5
OURS	200	160	$\times$1	59.0	41.6	68.5	77.0
OURS	200	224	$\times$1.6	59.6	42.0	70.9	77.4

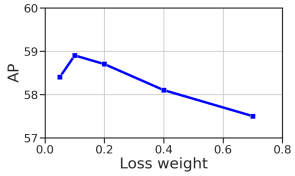
Table 5: **Efficient SSL training on ImageNet.** Performance of object detection and segmentation for BYOL and OURS for different pre-training epochs and training image size. We achieve better performance than BYOL (1000 epochs pretraining) with our model pre-trained with 200 epochs and with training image size 160 that is 5.3$\times$ faster to pre-train.

Method	VOC AP^b	COCO mAP	VOC mIoU	Cityscapes mIoU
BYOL	57.0	40.9	69.0	73.4
BYOL +Local MSE loss	58.7	42.0	70.5	76.7
OURS	**59.6**	**42.5**	**72.1**	**77.8**

Table 6: **Effectiveness of our LC-loss over MSE-loss.**



(a) **Effect of longer pre-training.** The plot shows average AP for object detection on COCO validation sets (left) and PASCAL VOC (right).



(b) **Effect of relative weights of LC-loss**. The plot shows the average AP for object detection on VOC.

Figure 3: **Ablation Studies** on pretraining epochs and relative weights of local contrastive loss.

ilarity between the feature representations. We show the matching at the original image resolution. Figure 4 shows

Figure 4: **Correspondence Visualization.** (Top) Ground-truth correspondence between two transformed versions of the same image. (Bottom) Correspondence prediction from OURS.

that our method predicts accurate matches most of the time (considering the resolution error due to the 32×32 grid size in the pixel space for each feature point).

More Analysis on Correspondence. To show that our method is learning better correspondence across datasets, we perform a simple experiment. Given an image, we flip the image along the horizontal direction, and apply color transformations (Gaussian blur, color jitter, and random grayscale operations). Since the only spatial transformation is horizontal flipping, the corresponding pixels are simply at the mirror locations of the original pixels, i.e., for a pixel location (x,y), the correct correspondence location in the transformed image is (w-x, y), where w is the width. Note that the correspondence is measured in the feature locations, not in actual pixel locations. We can also measure the correspondence accuracy based on whether the matching is correct or not. We use the ImageNet pre-trained backbone from BYOL and OURS, and evaluate the correspondence

Method	EuroSAT[22]	CropDisease[30]	ChestX[36]	ISIC[8]	Sketch[35]	DTD[7]	Omniglot[26]	Avg
Supervised	85.8	92.5	25.2	43.4	86.3	81.9	93.0	72.6
SoCo	78.3	84.1	25.1	41.2	81.5	73.9	92.2	68.0
DenseCL	77.7	81.0	23.8	36.8	76.5	78.3	77.4	64.5
PixPro	80.5	86.4	26.5	41.2	81.5	73.9	92.2	68.9
OURS	84.5	90.1	25.2	41.9	85.6	80.2	91.5	71.3

Table 7: **Few-shot learning results on downstream datasets.** The pre-trained models are used as fixed feature extractors We report top-1 accuracy for 5-way 5-shot averaged over 600 episodes. We use the publicly available pre-trained backbone as feature extractor for the few-shot evaluation.

Method	Pretrain Epoch	AP	AP_{50}	AP_{75}
Supervised	90	65.7	87.2	71.5
BYOL	300	66.3	87.4	72.4
VADeR [32]	200	66.1	87.3	72.1
SCRL [34]	1000	66.5	87.8	72.3
DenseCL†	200	66.2	87.3	71.9
PixPro †	400	66.6	**87.8**	72.8
OURS	400	**67.2**	87.4	**73.7**

Table 8: **COCO keypoint estimation.** Supervised and BYOL results are from [34]. (†) denotes We use the publicly available ImageNet-pretrained checkpoints released by the authors and finetune on the COCO dataset.

	5%			10%		
Method	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Supervised	19.2	31.0	20.5	25.0	39.9	26.6
BYOL	21.9	36.2	23.2	27.1	43.4	29.3
PixPro	20.3	31.4	22.1	25.4	39.5	27.4
OURS	**23.3**	**37.4**	**25.0**	**27.9**	**44.0**	**30.1**

Table 9: **Object detection on mini-COCO with 1× schedule.** All scores are obtained from finetuning the publicly available pre-trained backbone on the downstream dataset.

on the COCO dataset. Figure 5 shows some visual examples of the correspondence map on images from the COCO dataset. We also measure the accuracy of correct correspondences on the COCO val dataset. For the BYOL pre-trained model, the accuracy is only ∼33%, PixPro achieves ∼99% accuracy, and OURS achieves ∼96% accuracy. The results suggest that our approach is more robust against color transformations. We infer that PixPro achieves better accuracy as PixPro is trained only with local consistency loss, whereas we use both global and local correspondence during pre-training. The models have not been trained on COCO. Hence, the results also show that the correspondence maps generalize to other datasets.

5. Discussion and Conclusion

Even though our model consistently improves performance on detection and segmentation, it has some limitations. First, we calculate local correspondence loss at low spatial resolution (32× down-sampled from the original image resolution for ResNet50). Computing LC-loss at higher resolutions may be beneficial but is much more computationally expensive. Thus, the trade-off between performance and accuracy needs to be further studied. Second, the local correspondences are not sampled at *good* feature points (e.g., corners); rather they are sampled on a uniform 2D grid. Thus, LC-loss might be too strict when dealing with large, texture-less image regions. Our loss also does not account for presence of self-similar image regions and the effect of not modeling them needs to be evaluated.

(a) BYOL (b) PixPro (c) OURS

Figure 5: **Correspondence map for an image and its transformed versions with only flipping and color transformation.** (a) Results from BYOL without local loss. There are many erroneous corresponding pixels denoting that global loss alone does not learn good features for local correspondence. (b) and (c): Results from PixPro and OURS, both of which perfectly detects most of the correspondences.

To summarise, we propose a simple framework for self-supervised learning that leverages known pixel correspondences between different transformations of an image. We showed that the model pre-trained with our approach provides better representations for detection and segmentation. Imposing our loss enables a single network to retain both spatial and global information, both of which we have shown are necessary to obtain good features. Our training does not require any external supervision, since all the local and global constraints are generated from the input image itself. We showed that our method outperforms existing self-supervised methods that impose local consistency without requiring complex architectural components such as encoder-decoder layers, propagation modules, and regions proposal networks.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, et al. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint:2006.09882*, 2020.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, et al. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

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

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

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

[7] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.

[8] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019.

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

[10] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*, 2015.

[13] 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.

[14] Hugo Germain, Vincent Lepetit, and Guillaume Bourmaud. Visual correspondence hallucination: Towards geometric reasoning. *arXiv preprint arXiv:2106.09711*, 2021.

[15] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised Representation Learning by Predicting Image Rotations. In *ICLR*. arXiv, 3 2018.

[16] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *Proceedings of the ieee/cvf International Conference on computer vision*, pages 6391–6400, 2019.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[18] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[19] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4918–4927, 2019.

[20] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask RCNN. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

[23] Olivier J Henaff, Skanda Koppula, Jean-Baptiste Alayrac, Aaron van den Oord, Oriol Vinyals, and João Carreira. Efficient visual pretraining with contrastive detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10086–10096, 2021.

[24] Olivier J Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[25] Ashraful Islam, Chun-Fu Richard Chen, Rameswar Panda, Leonid Karlinsky, Richard Radke, and Rogerio Feris. A broad study on the transferability of visual representations with contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8845–8855, 2021.

[26] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[29] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[30] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419, 2016.

[31] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context Encoders: Feature Learning by Inpainting. In *CVPR*, volume 2016-December, pages 2536–2544. IEEE Computer Society, 4 2016.

[32] Pedro O Pinheiro, Amjad Almahairi, Ryan Y Benmalek, Florian Golemo, and Aaron Courville. Unsupervised learning of dense visual representations. *arXiv preprint arXiv:2011.05499*, 2020.

[33] Shaoqing Ren, Kaiming He, Ross 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(6):1137–1149, 2016.

[34] Byungseok Roh, Wuhyun Shin, Ildoo Kim, and Sungwoong Kim. Spatially consistent representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1144–1153, 2021.

[35] Haohan Wang, Songwei Ge, Eric P. Xing, and Zachary C. Lipton. Learning robust global representations by penalizing local predictive power. *arXiv preprint arXiv:1905.13549*, 2019.

[36] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2106, 2017.

[37] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[38] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021.

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

[40] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 418–434, 2018.

[41] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10539–10548, 2021.

[42] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[43] 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*, pages 16684–16693, 2021.

[44] 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*, pages 3987–3996, 2021.

[45] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *ECCV*, volume 9907 LNCS, pages 649–666. Springer Verlag, 3 2016.