Self-Supervised Learning of Pretext-Invariant Representations

Ishan Misra   Laurens van der Maaten
Facebook AI Research

Abstract

The goal of self-supervised learning from images is to construct image representations that are semantically meaningful via pretext tasks that do not require semantic annotations. Many pretext tasks lead to representations that are covariant *with image transformations. We argue that, instead, semantic representations ought to be* invariant *under such transformations. Specifically, we develop Pretext-Invariant Representation Learning (PIRL, pronounced as "pearl") that learns invariant representations based on pretext tasks. We use PIRL with a commonly used pretext task that involves solving jigsaw puzzles. We find that PIRL substantially improves the semantic quality of the learned image representations. Our approach sets a new state-of-the-art in self-supervised learning from images on several popular benchmarks for self-supervised learning. Despite being unsupervised,* ***PIRL outperforms supervised pre-training*** *in learning image representations for object detection. Altogether, our results demonstrate the potential of self-supervised representations with good invariance properties.*

1. Introduction

Modern image-recognition systems learn image representations from large collections of images and corresponding semantic annotations. These annotations can be provided in the form of class labels [66], hashtags [46], bounding boxes [16, 43], *etc.* Pre-defined semantic annotations scale poorly to the long tail of visual concepts [75], which hampers further improvements in image recognition.

Self-supervised learning tries to address these limitations by learning image representations from the pixels themselves without relying on pre-defined semantic annotations. Often, this is done via a *pretext* task that applies a transformation to the input image and requires the learner to predict properties of the transformation from the transformed image (see Figure 1). Examples of image transformations used include rotations [20], affine transformations [33, 57, 65, 85], and jigsaw transformations [54]. As the pretext task involves predicting a property of the image transformation, it encourages the construction of image representations that are *covariant* to the transformations. Although such covariance is beneficial for tasks such as predicting 3D correspondences [33, 57, 65], it is undesirable for most semantic recognition tasks. Representations ought to be *invariant* under image transformations to be useful for image recognition [14, 31] because the transformations do not alter visual semantics. In fact, invariance is one of the core tenets of designing 'good' features [8, 45, 48].

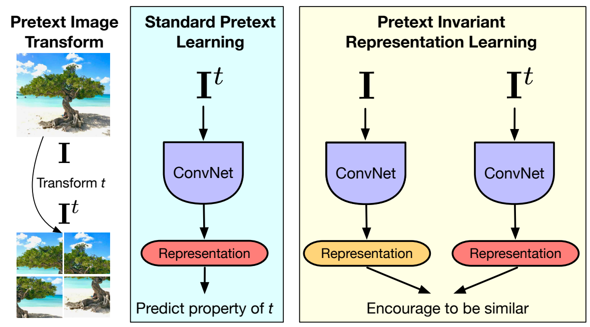

Figure 1: Pretext-Invariant Representation Learning (PIRL). Many pretext tasks for self-supervised learning [20, 54, 85] involve transforming an image $\mathbf{I}$, computing a representation of the transformed image, and predicting properties of transformation t from that representation. As a result, the representation must *covary* with the transformation t and may not contain much semantic information. By contrast, PIRL learns representations that are *invariant* to the transformation t and retain semantic information.

Motivated by this observation, we propose a method that learns invariant representations rather than covariant ones. Instead of predicting properties of the image transformation, Pretext-Invariant Representation Learning (PIRL) constructs image representations that are *similar* to the representation of transformed versions of the same image and *different* from the representations of other images. We adapt the "Jigsaw" pretext task [54] to work with PIRL and find that the resulting invariant representations perform better than their covariant counterparts across a range of vision tasks. PIRL substantially outperforms all prior art in self-supervised learning from ImageNet (Figure 2) and from uncurated image data (Table 4). Interestingly, PIRL even outperforms supervised pre-training in learning image representations suitable for object detection (Table 1 & supplemental material).

2. Related Work

Modeling invariances in features is a well studied concept in computer vision with decades of research [48] and plays a critical role in hand-designed features such as SIFT [45], HOG [8], and learned representations from ConvNets [37, 40, 69]. Practically useful representations are designed to be invariant to ‘nuisance’ factors like translations of pixels, change in scale, color, lighting, *e.g.*, by using data augmentation [37] during training. In our work, we propose to leverage the invariance to self-supervised ‘pretext tasks’.

We learn feature representations without considering a corresponding (image-conditional) label distribution. Prior work has studied reconstructing images from a small, intermediate representation, *e.g.*, using sparse coding [58], adversarial training [12, 13, 50], autoencoders [49, 63, 76], or probabilistic versions thereof [67].

More recently, interest has shifted to specifying pretext tasks [10] that require modeling a more limited set of properties of the data distribution. For video data, these pretext tasks learn representations by ordering video frames [1, 18, 34, 41, 51, 79, 83], tracking [62, 77], or using cross-modal signals like audio [2, 3, 19, 36, 60, 61].

Our work focuses on image-based pretext tasks. Prior pretext tasks include image colorization [9, 30, 38, 39, 86, 87], orientation prediction [20], affine transform prediction [85], predicting contextual image patches [10], reordering image patches [5, 21, 53, 54, 56], counting visual primitives [55], or their combinations [11]. These pretext tasks typically involve predicting some low-level property of an image transformation which makes the final representations covariant to image transformations. In contrast, our work learns image representations that are invariant to the image transformations rather than covariant.

PIRL is related to approaches that learn invariant image representations via contrastive learning [15, 29, 68, 77, 81], clustering [6, 7, 56, 78], or maximizing mutual information [4, 29, 31]. PIRL is most similar to methods that learn representations that are invariant under standard data augmentation [4, 14, 29, 31, 81, 82]. PIRL learns representations that are invariant to both the data augmentation and to the pretext image transformations. Similar to our work, recent methods also focus on invariance [47] or decoupling the pretext task [17] to learn representations. PIRL can be viewed as extending the set of data augmentations to include prior pretext tasks and provides a new way to combine pretext tasks with contrastive learning.

Finally, PIRL is also related to approaches that use a contrastive loss [24] in predictive learning [25, 26, 28, 59, 70, 73]. These prior approaches predict missing parts of the data, *e.g.*, future frames in videos [25, 59], or operate on multiple views [73]. In contrast to those approaches, PIRL learns invariances rather than predicting missing data.

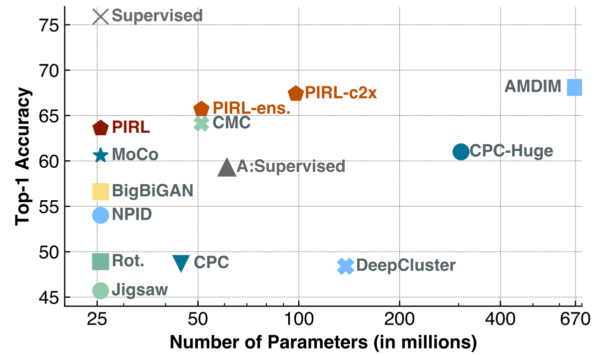

Figure 2: ImageNet classification with linear models. Single-crop top-1 accuracy on the ImageNet validation data as a function of the number of parameters in the model that produces the representation (“A” represents AlexNet). Pretext-Invariant Representation Learning (PIRL) sets a new state-of-the-art in this setting (red marker) and uses significantly smaller models (ResNet-50). See Section 4.2 for more details.

3. PIRL: Pretext-Invariant Representation Learning

Our work focuses on pretext tasks for self-supervised learning in which a known image transformation is applied to the input image. For example, the “Jigsaw” task divides the image into nine patches and perturbs the image by randomly permuting the patches [54]. Prior work used Jigsaw as a pretext task by predicting the permutation from the perturbed input image. This requires the learner to construct a representation that is *covariant* to the perturbation. The same is true for a range of other pretext tasks that have recently been studied [10, 20, 51, 85]. In this work, we adopt the existing Jigsaw pretext task in a way that encourages the image representations to be *invariant* to the image patch perturbation. While we focus on the Jigsaw pretext task in this paper, our approach is applicable to any pretext task that involves image transformations (see Section 5.3).

3.1. Overview of the Approach

Suppose we are given an image dataset, $\mathcal{D} = \{\mathbf{I}_1, \dots, \mathbf{I}_{|\mathcal{D}|}\}$ with $\mathbf{I}_n \in \mathbb{R}^{H \times W \times 3}$, and a set of image transformations, $\mathcal{T}$. The set $\mathcal{T}$ may contain transformations such as a re-shuffling of patches in the image [54], image rotations [20], *etc*. We aim to train a convolutional network, $\phi_\theta(\cdot)$, with parameters θ that constructs image representations $\mathbf{v_I} = \phi_\theta(\mathbf{I})$ that are invariant to image transformations $t \in \mathcal{T}$. We adopt an empirical risk minimization approach to learning the network parameters θ. Specifically, we train the network by minimizing the empirical risk:

$$\ell_{inv}(\theta; \mathcal{D}) = \mathbb{E}_{t \sim p(\mathcal{T})}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{I} \in \mathcal{D}} L\left(\mathbf{v_I}, \mathbf{v_{I^t}}\right)\right], \quad (1)$$

where $p(\mathcal{T})$ is some distribution over the transformations in $\mathcal{T}$, and $\mathbf{I}^t$ denotes image $\mathbf{I}$ after application of transformation t, that is, $\mathbf{I}^t = t(\mathbf{I})$. The function $L(\cdot,\cdot)$ is a loss function that measures the similarity between two image representations. Minimization of this loss encourages the network $\phi_\theta(\cdot)$ to produce the same representation for image $\mathbf{I}$ as for its transformed counterpart $\mathbf{I}^t$, *i.e.*, to make representation invariant under transformation t.

We contrast our loss function to losses [10, 20, 51, 54, 85] that learn image representations $\mathbf{v_I} = \phi_\theta(\mathbf{I})$ that are covariant to image transformations $t \in \mathcal{T}$ by minimizing:

$$\ell_{co}(\theta; \mathcal{D}) = \mathbb{E}_{t\sim p(\mathcal{T})}\left[\frac{1}{|\mathcal{D}|}\sum_{\mathbf{I}\in\mathcal{D}} L_{co}\left(\mathbf{v_I}, z(t)\right)\right], \quad (2)$$

where z is a function that measures some properties of transformation t. Such losses encourage network $\phi_\theta(\cdot)$ to learn image representations that contain information on transformation t, thereby encouraging it to maintain information that is not semantically relevant.

Loss function. We implement $\ell_{inv}(\cdot)$ using a contrastive loss function $L(\cdot,\cdot)$ [24]. Specifically, we define a matching score, $s(\cdot,\cdot)$, that measures the similarity of two image representations and use this matching score in a noise contrastive estimator [23]. In our noise contrastive estimator (NCE), each "positive" sample $(\mathbf{I}, \mathbf{I}^t)$ has N corresponding "negative" samples. The negative samples are obtained by computing features from other images, $\mathbf{I}' \neq \mathbf{I}$. The noise contrastive estimator models the probability of the binary event that $(\mathbf{I}, \mathbf{I}^t)$ originates from data distribution as:

$$h(\mathbf{v_I}, \mathbf{v_{I^t}}) = \frac{\exp\left(\frac{s(\mathbf{v_I}, \mathbf{v_{I^t}})}{\tau}\right)}{\exp\left(\frac{s(\mathbf{v_I}, \mathbf{v_{I^t}})}{\tau}\right) + |\mathcal{D}_N|/|\mathcal{D}|}. \quad (3)$$

Herein, $\mathcal{D}_N \subseteq \mathcal{D}$ is a set of N negative samples that are drawn uniformly at random from dataset $\mathcal{D}$, τ is a temperature parameter, and $s(\cdot,\cdot)$ is the cosine similarity between the representations.

In practice, we apply different "heads" to the features before computing the score $s(\cdot,\cdot)$. Specifically, we apply head $f(\cdot)$ on features $(\mathbf{v_I})$ of $\mathbf{I}$ and head $g(\cdot)$ on features $(\mathbf{v_{I^t}})$ of $\mathbf{I}^t$; see Figure 3 and Section 3.3. NCE then amounts to minimizing the following loss:

$$\begin{aligned} L_{\text{NCE}}\left(\mathbf{I}, \mathbf{I}^t\right) = & -\log\left[h\left(f(\mathbf{v_I}), g(\mathbf{v_{I^t}})\right)\right] \quad (4) \\ & - \sum_{\mathbf{I}'\in\mathcal{D}_N} \log\left[1 - h\left(g(\mathbf{v}_\mathbf{I}^t), f(\mathbf{v_{I'}})\right)\right]. \end{aligned}$$

This loss encourages the representation of image $\mathbf{I}$ to be similar to that of its transformed counterpart $\mathbf{I}^t$, and the representation of $\mathbf{I}^t$ to be dissimilar to that of other images $\mathbf{I}'$.

3.2. Using a Memory Bank of Negative Samples

Prior work has found that it is important to use a large number of negatives in the NCE loss of Equation 4 [59, 81].

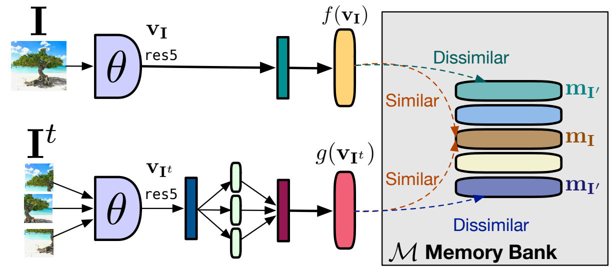

Figure 3: Overview of PIRL. Pretext-Invariant Representation Learning (PIRL) aims to construct image representations that are invariant to the image transformations $t \in \mathcal{T}$. PIRL encourages the representations of the image, $\mathbf{I}$, and its transformed counterpart, $\mathbf{I}^t$, to be similar. It achieves this by minimizing a contrastive loss (see Section 3.1). Following [81], PIRL uses a memory bank, $\mathcal{M}$, of negative samples to be used in the contrastive learning. The memory bank contains a moving average of representations, $\mathbf{m_I} \in \mathcal{M}$, for all images in the dataset (see Section 3.2).

In a mini-batch SGD optimizer, it is difficult to obtain a large number of negatives without increasing the batch to an infeasibly large size. To address this problem, we follow [81] and use a memory bank of "cached" features.

The memory bank, $\mathcal{M}$, contains a feature representation $\mathbf{m_I}$ for each image $\mathbf{I}$ in dataset $\mathcal{D}$. The representation $\mathbf{m_I}$ is an exponential moving average of feature representations $f(\mathbf{v_I})$ that were computed in prior epochs. This allows us to replace negative samples, $f(\mathbf{v}'_\mathbf{I})$, by their memory bank representations, $\mathbf{m_{I'}}$, in Equation 4 without having to increase the training batch size. We emphasize that the representations that are stored in the memory bank are all computed on the original images, $\mathbf{I}$, without the transformation t. This design decision gave better results.

Final loss function. A potential issue of the loss in Equation 4 is that it does not compare the representations of untransformed images $\mathbf{I}$ and $\mathbf{I}'$. We address this issue by using a convex combination of two NCE loss functions in $\ell_{inv}(\cdot)$:

$$\begin{aligned} L\left(\mathbf{I}, \mathbf{I}^t\right) = & \lambda L_{\text{NCE}}(\mathbf{m_I}, g(\mathbf{v_{I^t}})) \\ & + (1-\lambda) L_{\text{NCE}}(\mathbf{m_I}, f(\mathbf{v_I})). \quad (5) \end{aligned}$$

Herein, the first term is simply the loss of Equation 4 but uses memory representations $\mathbf{m_I}$ and $\mathbf{m_{I'}}$ instead of $f(\mathbf{v_I})$ and $f(\mathbf{v}'_\mathbf{I})$, respectively. The second term does two things: (1) it encourages the representation $f(\mathbf{v_I})$ to be similar to its memory representation $\mathbf{m_I}$, thereby dampening the parameter updates; and (2) it encourages the representations $f(\mathbf{v_I})$ and $f(\mathbf{v}'_\mathbf{I})$ to be dissimilar. Both the first and the second term use $\mathbf{m_{I'}}$ instead of $f(\mathbf{v}'_\mathbf{I})$ in Equation 4. Setting $\lambda = 0$ in Equation 5 leads to the loss used in [81]. We study the effect of λ on the learned representations in Section 5.

3.3. Implementation Details

Although PIRL can be used with any pretext task that involves image transformations, we focus on the Jigsaw pretext task [54] in this paper. To demonstrate that PIRL is more generally applicable, we also experiment with the Rotation pretext task [20] and with a combination of both tasks in Section 5.3. Below, we describe the implementation details of PIRL with the Jigsaw pretext task.

Convolutional network. We use a ResNet-50 (R-50) network architecture in our experiments [27]. The network is used to compute image representations for both $\mathbf{I}$ and $\mathbf{I}^t$. These representations are obtained by applying function $f(\cdot)$ or $g(\cdot)$ on features extracted from the the network.

Specifically, we compute the representation of $\mathbf{I}$, $f(\mathbf{v_I})$, by extracting `res5` features, average pooling, and a linear projection to obtain a 128-dimensional representation.

To compute the representation $g(\mathbf{v}_{\mathbf{I}^t})$ of a transformed image $\mathbf{I}^t$, we closely follow [21, 54]. We: (1) extract nine patches from image $\mathbf{I}$, (2) compute an image representation for each patch separately by extracting activations from the `res5` layer of the ResNet-50 and average pool the activations, (3) apply a linear projection to obtain a 128-dimensional patch representations, and (4) concatenate the patch representations in random order and apply a second linear projection on the result to obtain the final 128-dimensional image representation, $g(\mathbf{v}_{\mathbf{I}^t})$. Our motivation for this design of $g(\mathbf{v}_{\mathbf{I}^t})$ is the desire to remain as close as possible to the covariant pretext task of [20, 21, 54]. This allows apples-to-apples comparisons between the covariant approach and our invariant approach.

Hyperparameters. We implement the memory bank as described in [81] and use the same hyperparameters for the memory bank. Specifically, we set the temperature in Equation 3 to $\tau = 0.07$, and use a weight of 0.5 to compute the exponential moving averages in the memory bank. Unless stated otherwise, we use $\lambda = 0.5$ in Equation 5.

4. Experiments

Following common practice in self-supervised learning [21, 87], we evaluate the performance of PIRL in transfer-learning experiments. We perform experiments on a variety of datasets, focusing on object detection and image classification tasks. Our empirical evaluations cover: (1) a learning setting in which the parameters of the convolutional network are *finetuned* during transfer, thus evaluating the network "initialization" obtained using self-supervised learning and (2) a learning setting in which the parameters of the network are *fixed* during transfer learning, thus using the network as a feature extractor.

Baselines. An important baseline is the Jigsaw ResNet-50 model of [21] as it implements the covariant counterpart of our PIRL approach with the Jigsaw pretext task.

We also compare PIRL to a range of other self-supervised methods. An important comparison is to NPID [81]. NPID is a special case of PIRL: setting $\lambda = 0$ in Equation 5 leads to the loss function of NPID. We found it is possible to improve the original implementation of NPID by using more negative samples and training for more epochs (see Section 5). We refer to our improved version of NPID as NPID++. Comparisons between PIRL and NPID++ allow us to study the effect of the pretext-invariance that PIRL aims to achieve, *i.e.*, the effect of using $\lambda > 0$ in Equation 5.

Method	**Network**	**AP$^{\text{all}}$**	**AP^{50}**	**AP^{75}**	**ΔAP^{75}**
Supervised	R-50	52.6	**81.1**	57.4	=0.0
Jigsaw [21]	R-50	48.9	75.1	52.9	-4.5
Rotation [21]	R-50	46.3	72.5	49.3	-8.1
NPID++ [81]	R-50	52.3	79.1	56.9	-0.5
PIRL **(ours)**	R-50	**54.0**	<u>80.7</u>	**59.7**	**+2.3**
MoCo [26]	R-50	$55.2^\dagger$	$81.4^\dagger$	$61.2^\dagger$	

Table 1: Object detection on VOC07+12 trainval using Faster R-CNN. Detection AP on the VOC07 test set after finetuning Faster R-CNN models (BatchNorm fixed) with a ResNet-50 backbone pre-trained using self-supervised learning on ImageNet. Results for supervised ImageNet pre-training are presented for reference. Method with † finetunes BatchNorm. PIRL significantly outperforms supervised pre-training without extra pre-training data or changes in the network architecture. Additional results on VOC07 in supplemental.

Pre-training data. To facilitate comparisons with prior work, we use the 1.28M images from the ImageNet [66] `train` split (without labels) to pre-train our models.

Training details. We train our models using mini-batch SGD using the cosine learning rate decay [44] scheme with an initial learning rate of 1.2×10^{-1} and a final learning rate of 1.2×10^{-4}. We train the models for 800 epochs using a batch size of 1,024 images and using $N = 32,000$ negative samples in Equation 3. We do not use data-augmentation approaches such as Fast AutoAugment [42] because they are the result of supervised-learning approaches. We provide a full overview of all hyperparameter settings that were used in the supplemental material.

Transfer learning. Prior work suggests that the hyperparameters used in transfer learning can play an important role in the evaluation of pre-trained representations [21, 35, 87]. To facilitate fair comparisons with prior work, we closely follow the transfer-learning setup described in [21, 87].

4.1. Object Detection

Following prior work [21, 81], we perform object-detection experiments on the the Pascal VOC dataset [16] using the VOC07+12 trainval split. We use the Faster R-CNN [64] C4 object-detection model implemented in Detectron2 [80] with a ResNet-50 (R-50) backbone. We pre-train the ResNet-50 using PIRL to initialize the detection model before finetuning it on the VOC training data. We use the same training schedule as [21] for all models finetuned on VOC and follow [21, 80] to keep the BatchNorm parameters fixed during finetuning. We evaluate object-detection performance in terms of AP$^{\text{all}}$, AP^{50}, and AP^{75} [43].

The results of our detection experiments are presented in

Table 1. The results demonstrate the strong performance of PIRL: it outperforms all alternative self-supervised learnings in terms of all three AP measures. Compared to pre-training on the Jigsaw pretext task, PIRL achieves AP improvements of **5 points**. These results underscore the importance of learning invariant (rather than covariant) image representations. PIRL also outperforms NPID++, which demonstrates the benefits of learning pretext invariance.

Interestingly, PIRL even outperforms the supervised ImageNet-pretrained model in terms of the more conservative AP^{all} and AP^{75} metrics. Similar to concurrent work [26], we find that a self-supervised learner can **outperform supervised** pre-training for object detection. We emphasize that PIRL achieves this result using the *same* backbone model, the *same* number of finetuning epochs, and the exact *same* pre-training data (but without the labels). This result is a substantial improvement over prior self-supervised approaches that obtain worse performance than fully supervised baselines despite using orders of magnitude more curated training data [21] or much larger backbone models [28]. In the supplemental material, we show that PIRL also outperforms supervised pretraining when finetuning is done on the much smaller VOC07 trainval set. This suggests that PIRL learns image representations that are amenable to sample-efficient supervised learning.

4.2. Image Classification with Linear Models

Next, we assess the quality of image representations by training linear classifiers on fixed image representations. We follow the evaluation setup from [21] and measure the performance of such classifiers on four image-classification datasets: ImageNet [66], VOC07 [16], Places205 [88], and iNaturalist2018 [74]. These datasets involve diverse tasks such as object classification, scene recognition and fine-grained recognition. Following [21], we evaluate representations extracted from all intermediate layers of the pre-trained network, and report the image-classification results for the best-performing layer in Table 2.

ImageNet results. The results on ImageNet highlight the benefits of learning invariant features: PIRL improves recognition accuracies by over 15% compared to its covariant counterpart, Jigsaw. PIRL achieves the **highest single-crop top-1** accuracy of all self-supervised learners that use a single ResNet-50 model.

The benefits of pretext invariance are further highlighted by comparing PIRL with NPID. Our re-implementation of NPID (called NPID++) substantially outperforms the results reported in [81]. Specifically, NPID++ achieves a single-crop top-1 accuracy of 59%, which is higher or on par with existing work that uses a single ResNet-50. Yet, PIRL substantially outperforms NPID++. We note that PIRL also outperforms concurrent work [26] in this setting.

Akin to prior approaches, the performance of PIRL improves with network size. For example, CMC [73] uses a combination of two ResNet-50 models and trains the linear classifier for longer to obtain 64.1% accuracy. We performed an experiment in which we did the same for PIRL, and obtained a top-1 accuracy of 65.7%; see "PIRL-ens." in Figure 2. To compare PIRL with larger models, we also performed experiments in which we followed [35, 84] by doubling the number of channels in ResNet-50; see "PIRL-c2x" in Figure 2. PIRL-c2x achieves a top-1 accuracy of 67.4%, which is close to the accuracy obtained by AMDIM [4] with a model that has $6\times$ more parameters.

Altogether, the results in Figure 2 demonstrate that PIRL outperforms all prior self-supervised learners on ImageNet in terms of the trade-off between model accuracy and size. Indeed, PIRL even outperforms most self-supervised learners that use much larger models [28, 59].

Results on other datasets. The results on the other image-classification datasets in Table 2 are in line with the results on ImageNet: PIRL substantially outperforms its covariant counterpart (Jigsaw). The performance of PIRL is within 2% of fully supervised representations on Places205, and improves the previous best results of [21] on VOC07 by more than 16 AP points. On the challenging iNaturalist dataset, which has over $8,000$ classes, we obtain a gain

Method	Parameters	Transfer Dataset			
		ImageNet	VOC07	Places205	iNat.
ResNet-50 using evaluation setup of [21]					
Supervised	25.6M	75.9	87.5	51.5	45.4
Colorization [21]	25.6M	39.6	55.6	37.5	–
Rotation [20]	25.6M	48.9	63.9	41.4	23.0
NPID++ [81]	25.6M	59.0	76.6	46.4	32.4
MoCo [26]	25.6M	60.6	–	–	–
Jigsaw [21]	25.6M	45.7	64.5	41.2	21.3
PIRL (**ours**)	25.6M	**63.6**	**81.1**	**49.8**	**34.1**
Different architecture or evaluation setup					
NPID [81]	25.6M	54.0	–	45.5	–
BigBiGAN [13]	25.6M	56.6	–	–	–
AET [85]	61M	40.6	–	37.1	–
DeepCluster [6]	61M	39.8	–	37.5	–
Rot. [35]	61M	54.0	–	45.5	–
LA [89]	25.6M	$60.2^{\dagger}$	–	$50.2^{\dagger}$	–
CMC [73]	51M	64.1	–	–	–
CPC [59]	44.5M	48.7	–	–	–
CPC-v2 [28]	305M	61.0	–	–	–
BigBiGAN-Big [13]	86M	61.3	–	–	–
AMDIM [4]	670M	68.1	–	55.1	–

Table 2: Image classification with linear models. Image-classification performance on four datasets using the setup of [21]. We train linear classifiers on image representations obtained by self-supervised learners that were pre-trained on ImageNet (without labels). We report the performance for the best-performing layer for each method. We measure mean average precision (mAP) on the VOC07 dataset and top-1 accuracy on all other datasets. Numbers for PIRL, NPID++, Rotation were obtained by us; the other numbers were adopted from their respective papers. Numbers with † were measured using 10-crop evaluation. The best-performing self-supervised learner on each dataset is **boldfaced**.

Method	Data fraction → Backbone	1%	10%
		Top-5 Accuracy	
Random initialization [81]	R-50	22.0	59.0
NPID [81]	R-50	39.2	77.4
Jigsaw [21]	R-50	45.3	79.3
NPID++ [81]	R-50	52.6	81.5
VAT + Ent Min. [22, 52]	R-50v2	47.0	83.4
S^4L Exemplar [84]	R-50v2	47.0	83.7
S^4L Rotation [84]	R-50v2	53.4	**83.8**
PIRL (**ours**)	R-50	**57.2**	**83.8**
Colorization [39]	R-152	29.8	62.0
CPC-v2 [28]	R-170 and R-11	64.0	84.9

Table 3: Semi-supervised learning on ImageNet. Single-crop top-5 accuracy on the ImageNet validation set of self-supervised models that are finetuned on 1% and 10% of the ImageNet training data, following [81]. All numbers except for Jigsaw, NPID++ and PIRL are adopted from the corresponding papers. Best performance is **boldfaced**.

of 11% in top-1 accuracy over the prior best result [20]. We observe that the NPID++ baseline performs well on these three datasets but is consistently outperformed by PIRL. Indeed, **PIRL sets a new state-of-the-art** for self-supervised representations in this learning setting on the VOC07, Places205, and iNaturalist datasets.

4.3. Semi-Supervised Image Classification

We perform semi-supervised image classification experiments on ImageNet following the experimental setup of [28, 81, 84]. Specifically, we randomly select 1% and 10% of the ImageNet training data (with labels). We finetune our models on these training-data subsets following the procedure of [81]. Table 3 reports the top-5 accuracy of the resulting models on the ImageNet validation set.

The results further highlight the quality of the image representations learned by PIRL: finetuning the models on just 1% (∼13,000) labeled images leads to a top-5 accuracy of 57%. PIRL performs at least as well as S^4L [84] and better than VAT [22], which are both methods specifically designed for semi-supervised learning. In line with earlier results, PIRL also outperforms Jigsaw and NPID++.

4.4. Pre-Training on Uncurated Image Data

Most representation learning methods are sensitive to the data distribution used during pre-training [21, 32, 46, 71]. To study how much changes in the data distribution impact PIRL, we pre-train models on uncurated images from the unlabeled YFCC dataset [72]. Following [7, 21], we randomly select a subset of 1 million images (YFCC-1M) from the 100 million images in YFCC. We pre-train PIRL ResNet-50 networks on YFCC-1M using the same procedure that was used for ImageNet pre-training. We evaluate using the setup in Section 4.2 and train linear classifiers on fixed image representations.

Table 4 reports the top-1 accuracy of the resulting classifiers. In line with prior results, PIRL outperforms competing self-supervised learners. In fact, PIRL even outperforms Jigsaw and DeeperCluster models that were trained on 100× more data from the same distribution. Comparing pre-training on ImageNet (Table 2) with pre-training YFCC-1M (Table 4) leads to a mixed set of observations. On ImageNet classification, pre-training (without labels) on ImageNet works substantially better than pre-training on YFCC-1M. In line with prior work [21, 32], however, pre-training on YFCC-1M leads to better representations for image classification on the Places205 dataset.

Method	Dataset	Transfer Dataset			
		ImageNet	VOC07	Places205	iNat.
Jigsaw [21]	YFCC1M	–	64.0	42.1	–
DeepCluster [6, 7]	YFCC1M	34.1	63.9	35.4	–
PIRL (**ours**)	YFCC1M	**57.8**	**78.8**	**51.0**	**29.7**
Jigsaw [21]	YFCC100M	48.3	71.0	44.8	–
DeeperCluster [7]	YFCC100M	45.6	73.0	42.1	–

Table 4: Pre-training on uncurated YFCC images. Top-1 accuracy or mAP (for VOC07) of linear image classifiers for four image-classification tasks, using various image representations. All numbers (except those for PIRL) are adopted from the corresponding papers. Deep(er)Cluster uses VGG-16 rather than ResNet-50. The best performance on each dataset is **boldfaced**. Top: Representations obtained by training ResNet-50 models on a randomly selected subset of one million images. Bottom: Representations learned from about 100 million YFCC images.

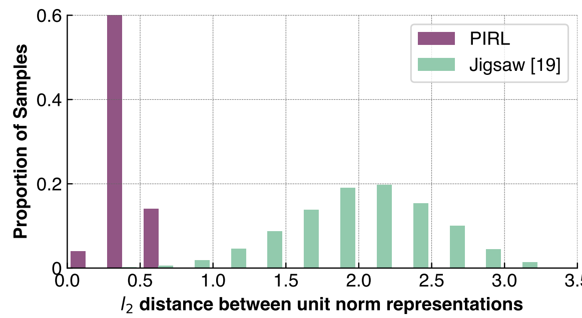

Figure 4: Invariance of PIRL representations. Distribution of l_2 distances between unit-norm image representations, $f(\mathbf{v_I})/\|f(\mathbf{v_I})\|^2$, and unit-norm representations of the transformed image, $g(\mathbf{v_{I^t}})/\|g(\mathbf{v_{I^t}})\|^2$. Distance distributions are shown for PIRL and Jigsaw representations.

5. Analysis

We performed a set of experiments aimed at better understanding the properties of PIRL. To make it feasible to train the larger number of models needed for these experiments, we train the models we study in this section for fewer epochs (400) and with fewer negatives ($N = 4,096$) than in Section 4. As a result, we obtain lower absolute performances. Apart from that, we did not change the experimental setup or any of the other hyperparameters. Throughout the section, we use the evaluation setup from Section 4.2 that trains linear classifiers on fixed image representations to measure the quality of image representations.

5.1. Analyzing PIRL Representations

Does PIRL learn invariant representations? PIRL was designed to learn representations that are invariant to image transformation $t \in \mathcal{T}$. We analyzed whether the learned representations actually have the desired invariance properties. Specifically, we normalize the representations to have unit norm and compute l_2 distances between the (normalized) representation of image, $f(\mathbf{v_I})$, and the (normalized) representation its transformed version, $g(\mathbf{v_{I^t}})$. We repeat this for all transforms $t \in \mathcal{T}$ and for a large set of images. We plot histograms of the distances thus obtained in Figure 4. The figure shows that, for PIRL, an image representation and the representation of a transformed version of that image are generally similar. This suggests that PIRL has learned representations that are invariant to the transformations. By contrast, the distances between Jigsaw representations have a much larger mean and variance, which suggests that Jigsaw representations covary with the image transformations that were applied.

Which layer produces the best representations? All prior experiments used PIRL representations that were extracted from the `res5` layer and Jigsaw representations that were extracted from the `res4` layer (which work better for Jigsaw). Figure 5 studies the quality of representations in earlier layers of the convolutional networks. The figure reveals that the quality of Jigsaw representations improves from the `conv1` to the `res4` layer but that their quality sharply decreases in the `res5` layer. We surmise this happens because the `res5` representations in the last layer of the network covary with the image transformation t and are not encouraged to contain semantic information. By contrast, PIRL representations are invariant to image transformations, which allows them to focus on modeling semantic information. As a result, the best image representations are extracted from the `res5` layer of PIRL-trained networks.

Multi-task Jigsaw and NPID++. To further understand PIRL, we implemented a multi-task baseline, similar to [17], which does not learn invariance to the Jigsaw task. This baseline uses two separate loss functions - NPID [81] which learns invariance to data augmentation, and Jigsaw classification which learns to predict the Jigsaw permutation applied to the input. This baseline performs similar or worse to NPID++ (within 0.2% transfer performance on ImageNet) showing that learning invariance to Jigsaw is important for better representations.

5.2. Analyzing the PIRL Loss Function

What is the effect of λ in the PIRL loss function? The PIRL loss function in Equation 5 contains a hyperparameter λ that trades off between two NCE losses. All prior experiments were performed with $\lambda = 0.5$. NPID(++) [81] is a special case of PIRL in which $\lambda = 0$, effectively removing the pretext-invariance term from the loss. At $\lambda = 1$, the network does not compare untransformed images at training time and updates to the memory bank $\mathbf{m_I}$ are not dampened.

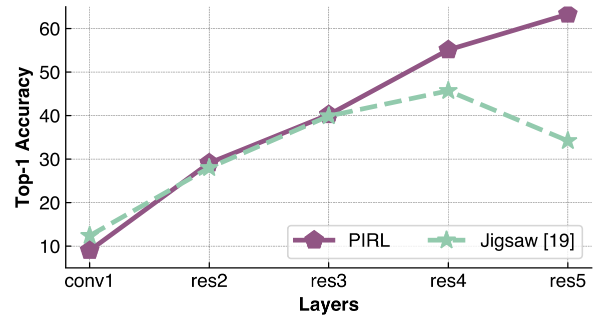

Figure 5: Quality of PIRL representations per layer. Top-1 accuracy of linear models trained to predict ImageNet classes based on representations extracted from various layers in ResNet-50 trained using PIRL and Jigsaw.

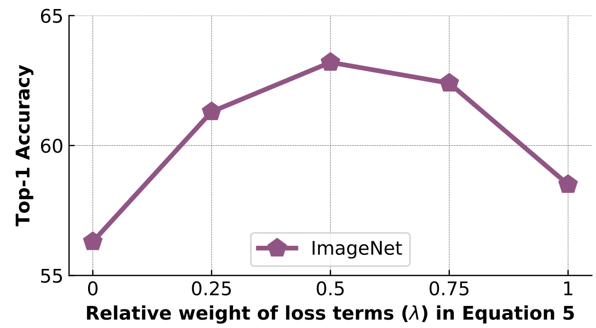

Figure 6: Effect of varying the trade-off parameter λ. Top-1 accuracy of linear classifiers trained to predict ImageNet classes from PIRL representations as a function of hyperparameter λ in Equation 5.

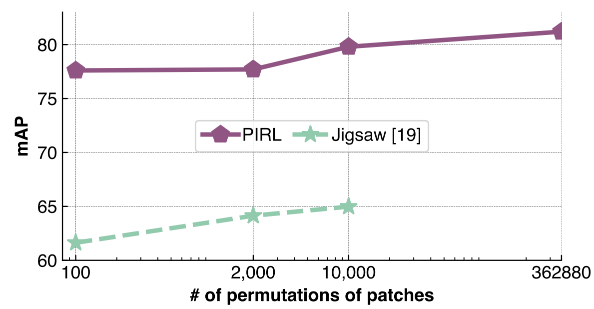

Figure 7: Effect of varying the number of patch permutations in $\mathcal{T}$. Performance of linear image classification models trained on the VOC07 dataset in terms of mAP. Models are initialized by PIRL and Jigsaw, varying the number of image transformations, $\mathcal{T}$, from 1 to $9! \approx 3.6 \times 10^5$.

We study the effect of λ on the quality of PIRL representations. As before, in Figure 6, we measure representation quality by the top-1 accuracy of linear classifiers operating on fixed ImageNet representations. The performance of PIRL is sensitive to the setting of λ, and the best performance is obtained by setting $\lambda = 0.5$.

Effect of the number of image transforms. Both in PIRL

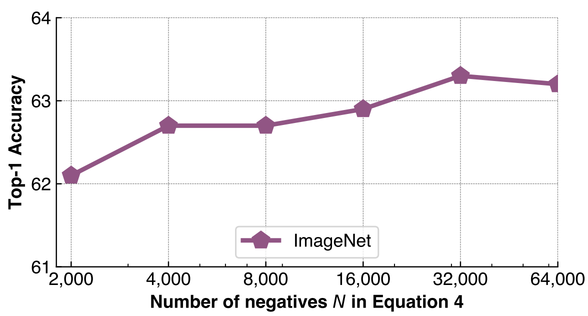

Figure 8: Effect of varying the number of negatives. Top-1 accuracy of linear classifiers trained to perform ImageNet classification using PIRL representations as a function of the number of negative samples, N.

Method	Params	Transfer Dataset			
		ImageNet	VOC07	Places205	iNat.
Rotation [20]	25.6M	48.9	63.9	41.4	23.0
PIRL (Rotation; **ours**)	25.6M	60.2	77.1	47.6	31.2
Δ of PIRL	-	**+11.3**	**+13.2**	**+6.2**	**+8.2**
Combining pretext tasks using PIRL					
PIRL (Jigsaw; **ours**)	25.6M	62.2	79.8	48.5	31.2
PIRL (Rotation + Jigsaw; **ours**)	25.6M	**63.1**	**80.3**	**49.7**	**33.6**

Table 5: Using PIRL with (combinations of) different pretext tasks. Top-1 accuracy / mAP of linear image classifiers trained on PIRL image representations. Top: Performance of PIRL used in combination with the Rotation pretext task [20]. Bottom: Performance of PIRL using a combination of multiple pretext tasks.

and Jigsaw, it is possible to vary the complexity of the task by varying the number of permutations of the nine image patches that are included in the set of image transformations, $\mathcal{T}$. Prior work on Jigsaw suggests that increasing the number of possible patch permutations leads to better performance [21, 54]. However, the largest value $|\mathcal{T}|$ can take is restricted because the number of learnable parameters in the output layer grows linearly with the number of patch permutations in models trained to solve the Jigsaw task. This problem does not apply to PIRL because it never outputs the patch permutations, and thus has a fixed number of model parameters. As a result, PIRL can use all $9! \approx 3.6 \times 10^5$ permutations in $\mathcal{T}$.

We study the quality of PIRL and Jigsaw as a function of the number of patch permutations included in $\mathcal{T}$. To facilitate comparison with [21], we measure quality in terms of image classification performance of linear models using the VOC07 dataset, following the setup in Section 4.2. The results are presented in Figure 7 and show that PIRL outperforms Jigsaw for all cardinalities of $\mathcal{T}$. PIRL particularly benefits from being able to use very large numbers of image transformations (*i.e.*, large $|\mathcal{T}|$) during training.

Effect of the number of negative samples. We study the effect of the number of negative samples, N, on the quality of the learned image representations. We measure the accuracy of linear ImageNet classifiers on fixed representations produced by PIRL as a function of the value of N used in pre-training. The results are presented in Figure 8. They suggest that increasing the number of negatives tends to have a positive influence on the quality of the image representations constructed by PIRL.

5.3. Generalizing PIRL to Other Pretext Tasks

Although we studied PIRL in the context of Jigsaw in this paper, PIRL can be used with any set of image transformations, $\mathcal{T}$. We performed an experiment evaluating the performance of PIRL using the Rotation pretext task [20]. We define $\mathcal{T}$ to contain image rotations by $\{0°, 90°, 180°, 270°\}$, and measure representation quality in terms of image-classification accuracy of linear models.

The results of these experiments are presented in Table 5 (top). In line with earlier results, models trained using PIRL (Rotation) outperform those trained using the Rotation pretext task of [20]. The performance gains obtained from learning a rotation-invariant representation are substantial, *e.g.* $+11\%$ top-1 accuracy on ImageNet. We also note that PIRL (Rotation) outperforms NPID++ (see Table 2).

In a second set of experiments, we combined the pretext image transforms from both the Jigsaw and Rotation tasks in the set of image transformations, $\mathcal{T}$. Specifically, we obtain $\mathbf{I}^t$ by first applying a rotation and then performing a Jigsaw transformation. The results of these experiments are shown in Table 5 (bottom). The results demonstrate that combining image transforms from multiple pretext tasks can further improve image representations.

6. Discussion and Conclusion

We studied Pretext-Invariant Representation Learning (PIRL) for learning representations that are invariant to image transformations applied in self-supervised pretext tasks. The rationale behind PIRL is that invariance to image transformations maintains semantic information in the representation. We obtain state-of-the-art results on multiple benchmarks for self-supervised learning in image classification and object detection. PIRL even outperforms supervised ImageNet pre-training on object detection.

In this paper, we used PIRL with the Jigsaw and Rotation image transformations. In future work, we aim to extend to richer sets of transformations. We also plan to investigate combinations of PIRL with clustering-based approaches [6, 7]. Like PIRL, those approaches use inter-image statistics but they do so in a different way. A combination of the two approaches may lead to even better image representations.

Acknowledgments: We are grateful to Rob Fergus, Yann LeCun, and Andrea Vedaldi for encouragement and feedback; Aaron Adcock, Naman Goyal, Priya Goyal, and Myle Ott for help with the code development; Rohit Girdhar, and Ross Girshick for feedback on the manuscript; Yuxin Wu, and Kaiming He for help with Detectron2.

References

[1] Unaiza Ahsan, Rishi Madhok, and Irfan Essa. Video jigsaw: Unsupervised learning of spatiotemporal context for video action recognition. In *WACV*, 2019.

[2] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.

[3] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[4] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *arXiv preprint arXiv:1906.00910*, 2019.

[5] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *CVPR*, 2019.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[7] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[9] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *ICCV*, 2015.

[10] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[11] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017.

[12] J. Donahue, P. Krahenbühl, and T. Darrell. Adversarial feature learning. In *ICLR*, 2016.

[13] Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. *arXiv preprint arXiv:1907.02544*, 2019.

[14] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9), 2016.

[15] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019.

[16] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan. 2015.

[17] Zeyu Feng, Chang Xu, and Dacheng Tao. Self-supervised representation learning by rotation feature decoupling. In *CVPR*, 2019.

[18] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017.

[19] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018.

[20] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[21] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning. In *ICCV*, 2019.

[22] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536, 2005.

[23] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *ICAIS*, pages 297–304, 2010.

[24] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[25] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshop*, pages 0–0, 2019.

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *arXiv 1911.05722*, 2019.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[28] Olivier J Hénaff, 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.

[29] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[30] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. *ACM Transactions on Graphics*, 35(4):110, 2016.

[31] X. Ji, J.F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019.

[32] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016.

[33] Angjoo Kanazawa, David W Jacobs, and Manmohan Chandraker. Warpnet: Weakly supervised matching for single-view reconstruction. In *CVPR*, 2016.

[34] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, 2019.

[35] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019.

[36] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[38] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016.

[39] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual

understanding. In *CVPR*, 2017.
[40] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1, 1989.
[41] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *CVPR*, 2017.
[42] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *NeurIPS*, 2019.
[43] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014.
[44] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.
[45] David Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999.
[46] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018.
[47] Pong C. Yuen Mang Ye, Xu Zhang and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, 2019.
[48] David Marr. Vision: A computational investigation into the human representation and processing of visual information, henry holt and co. *Inc., New York, NY*, 2(4.2), 1982.
[49] J. Masci, U. Meier, D. Cires, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *ICANN*, pages 52–59, 2011.
[50] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks. In *ICML*, 2017.
[51] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.
[52] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *TPAMI*, 41(8), 2018.
[53] T Nathan Mundhenk, Daniel Ho, and Barry Y Chen. Improvements to context based self-supervised learning. In *CVPR*, 2018.
[54] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.
[55] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.
[56] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018.
[57] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *CVPR*, 2018.
[58] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996.
[59] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
[60] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.
[61] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.
[62] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017.
[63] Marc'Aurelio Ranzato, Fu-Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007.
[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.
[65] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, 2017.
[66] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115, 2015.
[67] R. Salakhutdinov and G. Hinton. Deep Boltzmann machines. In *AI-STATS*, pages 448–455, 2009.
[68] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, 2018.
[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
[70] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.
[71] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017.
[72] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *arXiv preprint arXiv:1503.01817*, 2015.
[73] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.
[74] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, pages 8769–8778, 2018.
[75] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017.
[76] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.
[77] Xiaolong Wang and Abhinav Gupta. Unsupervised learning

of visual representations using videos. In *ICCV*, 2015.

[78] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, pages 1329–1338, 2017.

[79] Donglai Wei, Joseph Lim, Andrew Zisserman, and William T. Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

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

[81] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[82] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

[83] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019.

[84] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. *arXiv preprint arXiv:1905.03670*, 2019.

[85] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, 2019.

[86] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[87] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[88] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014.

[89] Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. Local aggregation for unsupervised learning of visual embeddings. In *ICCV*, pages 6002–6012, 2019.