Scaling and Benchmarking Self-Supervised Visual Representation Learning

Priya Goyal Dhruv Mahajan Abhinav Gupta* Ishan Misra*
Facebook AI Research

Abstract

Self-supervised learning aims to learn representations from the data itself without explicit manual supervision. Existing efforts ignore a crucial aspect of self-supervised learning - the ability to ***scale*** *to large amount of data because self-supervision requires no manual labels. In this work, we revisit this principle and scale two popular self-supervised approaches to* ***100 million images****. We show that by scaling on various axes (including data size and problem 'hardness'), one can largely match or* ***even exceed*** *the performance of supervised pre-training on a variety of tasks such as object detection, surface normal estimation (3D) and visual navigation using reinforcement learning. Scaling these methods also provides many interesting insights into the limitations of current self-supervised techniques and evaluations. We conclude that current self-supervised methods are not 'hard' enough to take full advantage of large scale data and do not seem to learn effective high level semantic representations. We also introduce an* ***extensive benchmark*** *across* ***9 different datasets and tasks****. We believe that such a benchmark along with* ***comparable*** *evaluation settings is necessary to make meaningful progress. Code is at:* `https://github.com/facebookresearch/fair_self_supervision_benchmark`.

1. Introduction

Computer vision has been revolutionized by high capacity Convolutional Neural Networks (ConvNets) [39] and large-scale labeled data (e.g., ImageNet [10]). Recently [42, 64], weakly-supervised training on hundreds of millions of images and thousands of labels has achieved state-of-the-art results on various benchmarks. Interestingly, even at that scale, performance increases only log-linearly with the amount of labeled data. Thus, sadly, what has worked for computer vision in the last five years has now become a bottleneck: the size, quality, and availability of supervised data.

One alternative to overcome this bottleneck is to use the self-supervised learning paradigm. In discriminative self-supervised learning, which is the main focus of this work, a model is trained on an auxiliary or 'pretext' task for which ground-truth is available for free. In most cases, the pretext task involves predicting some hidden portion of the data (for example, predicting color for gray-scale images [11, 37, 74]). Every year, with the introduction of new pretext tasks, the performance of self-supervised methods keeps coming closer to that of ImageNet supervised pre-training. The hope around self-supervised learning outperforming supervised learning has been so strong that a researcher has even bet gelato [1].

Yet, even after multiple years, this hope remains unfulfilled. Why is that? In attempting to come up with clever pretext tasks, we have forgotten a crucial tenet of self-supervised learning: **scalability**. Since no manual labels are required, one can easily scale training from a million to billions of images. However, it is still unclear what happens when we scale up self-supervised learning beyond the ImageNet scale to 100M images or more. Do we still see performance improvements? Do we learn something insightful about self-supervision? Do we surpass the ImageNet supervised performance?

In this paper, we explore scalability which is a core tenet of self-supervised learning. Concretely, we scale two popular self-supervised approaches (`Jigsaw` [48] and `Colorization` [74]) along three axes:

1. **Scaling pre-training data:** We first scale up both methods to 100× more data (YFCC-100M [65]). We observe that low capacity models like AlexNet [35] do not show much improvement with more data. This motivates our second axis of scaling.
2. **Scaling model capacity:** We scale up to a higher capacity model, specifically ResNet-50 [28], that shows much larger improvements as the data size increases. While recent approaches [14, 33, 72] used models like ResNet-50 or 101, we explore the relationship between model capacity and data size which we believe is crucial for future efforts in self-supervised learning.
3. **Scaling problem complexity:** Finally, we observe that to take full advantage of large scale data and higher capacity models, we need 'harder' pretext tasks. Specifically, we scale the 'hardness' (problem complexity) and observe that higher capacity models show a larger improvement on 'harder' tasks.

*Equal contribution

Task	Datasets	Description
Image classification		
§ 6.1	Places205	Scene classification. 205 classes.
(Linear Classifier)	VOC07	Object classification. 20 classes.
	COCO2014	Object classification. 80 classes.
Low-shot image classification		
§ 6.2	VOC07	$\leq$ 96 samples per class
(Linear Classifier)	Places205	$\leq$ 128 samples per class
Visual navigation		
§ 6.3 (Fixed ConvNet)	Gibson	Reinforcement Learning for navigation.
Object detection		
§ 6.4	VOC07	20 classes.
(Frozen conv body)	VOC07+12	20 classes.
Scene geometry (3D)		
§ 6.5 (Frozen conv body)	NYUv2	Surface Normal Estimation.

Table 1: 9 transfer datasets and tasks used for **Benchmarking** in §6.

Another interesting question that arises is: how does one quantify the visual representation's quality? We observe that due to the lack of a **standardized evaluation methodology** in self-supervised learning, it has become difficult to compare different approaches and measure the advancements in the area. To address this, we propose an **extensive benchmark suite** to evaluate representations using a consistent methodology. Our benchmark is based on the following principle: a good representation (1) transfers to *many* different tasks, and, (2) transfers with *limited* supervision and *limited* fine-tuning. We carefully choose 9 different tasks (Table 1) ranging from semantic classification/detection to 3D and actions (specifically, navigation).

Our results show that by scaling along the three axes, self-supervised learning can **outperform ImageNet supervised** pre-training using the *same* evaluation setup on non-semantic tasks of Surface Normal Estimation and Navigation. For semantic classification tasks, although scaling helps outperform previous results, the gap with supervised pre-training remains significant when evaluating fixed feature representations (without full fine-tuning). Surprisingly, self-supervised approaches are quite competitive on object detection tasks with or without full fine-tuning. For example, on the VOC07 **detection** task, **without any bells and whistles, our performance matches** the supervised ImageNet pre-trained model.

2. Related Work

Visual representation learning without supervision is an old and active area of research. It has two common modeling approaches: generative and discriminative. A generative approach tries to model the data distribution directly. This can be modeled as maximizing the probability of reconstructing the input [43, 51, 67] and optionally estimating latent variables [29, 58] or using adversarial training [15, 44]. Our work focuses on discriminative learning.

One form of discriminative learning combines clustering with hand-crafted features to learn visual representations such as image-patches [13, 62], object discovery [57, 63]. We focus on discriminative approaches that learn representations directly from the the visual input. A large portion of such approaches are grouped under the term 'self-supervised' learning [9] in which the key principle is to automatically generate 'labels' from the data. The label generation can either be domain agnostic [6, 8, 52, 72] or exploit structural properties of the domain, *e.g.*, spatial structure of images [12]. We explore the 'pretext' tasks [12] that exploit structural information of the visual data to learn representations. These approaches can broadly be divided into two types - methods that use multi-modal information, *e.g.* sound [53] and methods that use only the visual data (images, videos). Multi-modal information such as depth from a sensor [17], sound in a video [3, 4, 23, 53], sensors on an autonomous vehicle [2, 30, 79] *etc*. can be used to automatically learn visual representations without human supervision. One can also use the temporal structure in a video for self-supervised methods [21, 27, 41, 46, 47]. Videos can provide information about how objects move [54], the relation between viewpoints [69, 70] *etc*.

In this work, we choose to scale image-based self-supervised methods because of their ease of implementation. Many pretext tasks have been designed for images that exploit their spatial structure [12, 48–50], color information [11, 37, 38, 74], illumination [16], rotation [24] *etc*. These pretext tasks model different properties of images and have been shown to contain complementary information [14]. Given the abundance of such approaches to use, in our work, we focus on two popular approaches that are simple to implement, intuitive, and diverse: `Jigsaw` from [48] and `Colorization` from [74]. A concurrent work [33] also explores multiple self-supervised tasks but their focus is on the architectural details which is complementary to ours.

3. Preliminaries

We briefly describe the two image based self-supervised approaches [49, 74] that we study in this work and refer the reader to the original papers for detailed explanations. Both these methods do *not* use any supervised labels.

3.1. `Jigsaw` Self-supervision

This approach by Noroozi *et al*. [48] learns an image representation by solving jigsaw puzzles created from an input image. The method divides an input image $\boldsymbol{I}$ into $N = 9$ non-overlapping square patches. A 'puzzle' is then created by shuffling these patches randomly and a ConvNet is trained to predict the permutation used to create the puzzle. Concretely, each patch is fed to a N-way Siamese ConvNet with shared parameters to obtain patch representations. The patch representations are concatenated and used to predict the permutation used to create the puzzle. In practice, as the total number of permutations $N!$ can be large, a fixed subset $\mathcal{P}$ of the total $N!$ permutations is used. The prediction problem is reduced to classification into one of $|\mathcal{P}|$ classes.

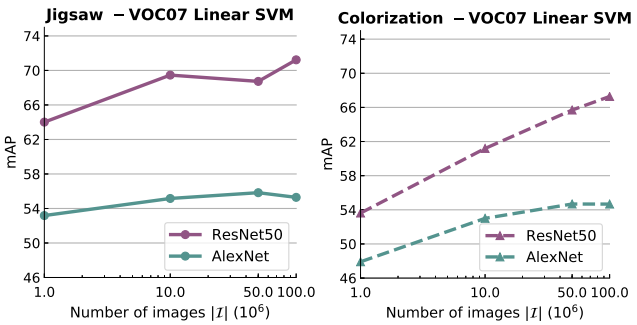

Figure 1: Scaling the Pre-training Data Size: The transfer learning performance of self-supervised methods on the VOC07 dataset for AlexNet and ResNet-50 as we vary the pre-training data size. We keep the problem complexity and data domain (different sized subsets of YFCC-100M) fixed. More details in § 4.1.

3.2. `Colorization` Self-supervision

Zhang *et al*. [74] learn an image representation by predicting color values of an input 'grayscale' image. The method uses the CIE *Lab* color space representation of an input image $\boldsymbol{I}$ and trains a model to predict the *ab* colors (denoted by $\boldsymbol{Y}$) from the input lightness L (denoted by $\boldsymbol{X}$). The output *ab* space is quantized into a set of discrete bins $\mathcal{Q} = 313$ which reduces the problem to a $|\mathcal{Q}|$-way classification problem. The target *ab* image $\boldsymbol{Y}$ is soft-encoded into $|\mathcal{Q}|$ bins by looking at the K-nearest neighbor bins (default value K=10). We denote this soft-encoded target explicitly by $\boldsymbol{Z}^K$. Thus, each $|\mathcal{Q}|$-way classification problem has K non-zero values. The ConvNet is trained to predict $\boldsymbol{Z}^K$ from the input lightness image $\boldsymbol{X}$.

4. Scaling Self-supervised Learning

In this section, we scale up current self-supervised approaches and present the insights gained from doing so. We *first* scale up the data size to $\mathbf{100}\times$ the size commonly used in existing self-supervised methods. However, observations from recent works [31, 42, 64] show that higher capacity models are required to take full advantage of large datasets. Therefore, we explore the *second axis* of scaling: model capacity. Additionally, self-supervised learning provides an interesting *third axis*: the complexity (hardness) of pretext tasks which can control the quality of the learned representations.

Finally, we observe the relationships between these three axes: whether the performance improvements on each of the axes are complementary or they encompass one other. To study this behavior, we introduce a simple investigation setup. Note that this setup is different from the extensive evaluation benchmark we propose in §6.

Investigation Setup: We use the task of image classification on PASCAL VOC2007 [19] (denoted as VOC07). We train linear SVMs [7] (with 3-fold cross validation to select the cost parameter) on fixed feature representations obtained from the ConvNet (setup from [53]). Specifically, we choose the best performing layer: `conv4` layer for AlexNet and the output of the last `res4` block (notation from [26]) for ResNet-50. We train on the `trainval` split and report mean Average Precision (mAP) on the `test` split.

Symbol	Description
YFCC-XM	Images from the YFCC-100M [65] dataset. We use subsets of size $X \in [1M, 10M, 50M, 100M]$.
ImageNet-22k	The full ImageNet dataset (22k classes, $14M$ images) [10].
ImageNet-1k	ILSVRC2012 dataset ($1k$ classes, $1.28M$ images) [56].

Table 2: A list of **self-supervised pre-training datasets** used in this work. We train AlexNet [35] and ResNet-50 [28] on these datasets.

4.1. Axis 1: Scaling the Pre-training Data Size

The first premise in self-supervised learning is that it requires 'no labels' and thus can make use of large datasets. But do the current self-supervised approaches benefit from increasing the pre-training data size? We study this for both the `Jigsaw` and `Colorization` methods. Specifically, we train on various subsets (see Table 2) of the YFCC-100M dataset - YFCC-$[1, 10, 50, 100]$ million images. These subsets were collected by randomly sampling respective number of images from the YFCC-100M dataset. We specifically create these YFCC subsets so we can keep the data domain fixed. Further, during the self-supervised pre-training, we keep other factors that may influence the transfer learning performance such as the model, the problem complexity ($|\mathcal{P}| = 2000$, $K = 10$) *etc*. fixed. This way we can isolate the effect of data size on performance. We provide training details in the supplementary material.

Observations: We report the transfer learning performance on the VOC07 classification task in Figure 1. We see that increasing the size of pre-training data improves the transfer learning performance for both the `Jigsaw` and `Colorization` methods on ResNet-50 and AlexNet. We also note that the `Jigsaw` approach performs better compared to `Colorization`. Finally, we make an interesting observation that the performance of the `Jigsaw` model saturates (log-linearly) as we increase the data scale from 1M to 100M.

4.2. Axis 2: Scaling the Model Capacity

We explore the relationship between model capacity and self-supervised representation learning. Specifically, we observe this relationship in the context of the pre-training dataset size. For this, we use AlexNet and the higher capacity ResNet-50 [28] model to train on the same pre-training subsets from § 4.1.

Observations: Figure 1 shows the transfer learning performance on the VOC07 classification task for `Jigsaw` and `Colorization` approaches. We make an important observation that the performance gap between AlexNet and ResNet-50 (as a function of the pre-training dataset size) keeps *increasing*. This suggests that higher capacity models

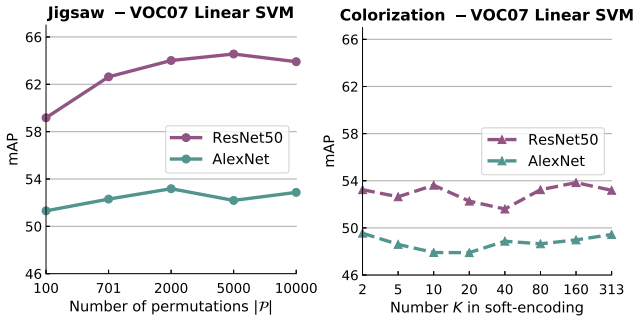

Figure 2: Scaling Problem Complexity: We evaluate transfer learning performance of `Jigsaw` and `Colorization` approaches on VOC07 dataset for both AlexNet and ResNet-50 as we vary the problem complexity. The pre-training data is fixed at YFCC-1M (§ 4.3) to isolate the effect of problem complexity.

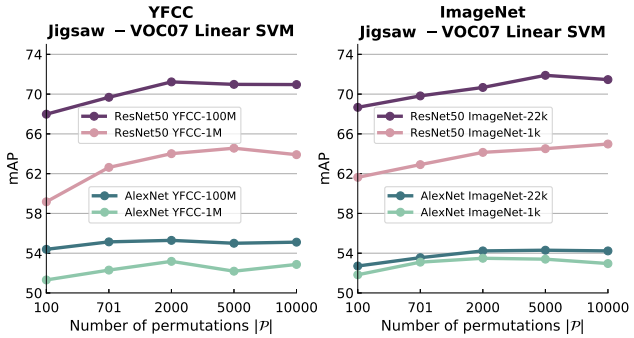

Figure 3: Scaling Data and Problem Complexity: We vary the pre-training data size and `Jigsaw` problem complexity for both AlexNet and ResNet-50 models. We pre-train on two datasets: ImageNet and YFCC and evaluate transfer learning performance on VOC07 dataset.

are needed to take full advantage of the larger pre-training datasets.

4.3. Axis 3: Scaling the Problem Complexity

We now scale the problem complexity ('hardness') of the self-supervised approaches. We note that it is important to understand how the complexity of the pretext tasks affects the transfer learning performance.

Jigsaw: The number of permutations $|\mathcal{P}|$ (§ 3.1) determines the number of puzzles seen for an image. We vary the number of permutations $|\mathcal{P}| \in [100,\ 701,\ 2k,\ 5k,\ 10k]$ to control the problem complexity. Note that this is a $\mathbf{10}\times$ increase in complexity compared to [48].

Colorization: We vary the number of nearest neighbors K for the soft-encoding (§ 3.2) which controls the hardness of the colorization problem.

To isolate the effect of problem complexity, we fix the pre-training data at YFCC-1M. We explore additional ways of increasing the problem complexity in the supplementary material.

Observations: We report the results on the VOC07 classification task in Figure 2. For the `Jigsaw` approach, we see an improvement in transfer learning performance as the size of the permutation set increases. ResNet-50 shows a **5 point** mAP improvement while AlexNet shows a smaller 1.9 point improvement. The `Colorization` approach appears to be less sensitive to changes in problem complexity. We see $\sim$2 point mAP variation across different values of K. We believe one possible explanation for this is in the structure encoded in the representation by the pretext task. For `Colorization`, it is important to represent the relationship between the semantic categories and their colors, but fine-grained color distinctions do not matter as much. On the other hand, `Jigsaw` encodes more spatial structure as the problem complexity increases which may matter more for downstream transfer task performance.

4.4. Putting it together

Finally, we explore the relationship between all the three axes of scaling. We study if these axes are orthogonal and if the performance improvements on each axis are complementary. We show this for `Jigsaw` approach only as it outperforms the `Colorization` approach consistently. Further, besides using YFCC subsets for pretext task training (from § 4.1), we also report self-supervised results for ImageNet datasets (without using any labels). Figure 3 shows the transfer learning performance on VOC07 task as function of data size, model capacity and problem complexity.

We note that transfer learning performance increases on all three axes, *i.e.*, increasing problem complexity still gives performance boost on ResNet-50 even at 100M data size. Thus, we conclude that the three axes of scaling are complementary. We also make a crucial observation that the performance gains for increasing problem complexity are almost negligible for AlexNet but significantly higher for ResNet-50. This indicates that we need higher capacity models to exploit hardness of self-supervised approaches.

5. Pre-training and Transfer Domain Relation

Thus far, we have kept the pre-training dataset and the transfer dataset/task fixed at YFCC and VOC07 respectively. We now add the following pre-training and transfer dataset/task to better understand the relationship between pre-training and transfer performance.

Pre-training dataset: We use both the ImageNet [10] and YFCC datasets from Table 2. Although the ImageNet datasets [10, 56] have supervised labels, we use them (without labels) to study the effect of the pre-training domain.

Transfer dataset and task: We further evaluate on the Places205 scene classification task [77]. In contrast to the object centric VOC07 dataset, Places205 is a scene centric dataset. Following the investigation setup from §4, we keep the feature representations of the ConvNets fixed. As the Places205 dataset has >2M images, we follow [75] and

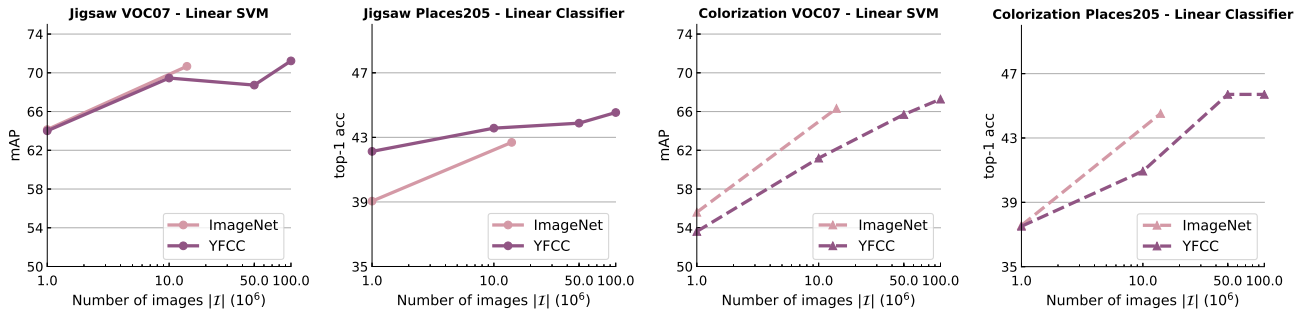

Figure 4: Relationship between pre-training and transfer domain: We vary pre-training data domain - (ImageNet-[1k, 22k], subsets of YFCC-100M) and observe transfer performance on the VOC07 and Places205 classification tasks. The similarity between the pre-training and transfer task domain shows a strong influence on transfer performance.

train linear classifiers using SGD. We use a batchsize of 256, learning rate of 0.01 decayed by a factor of 10 after every $40k$ iterations, and train for $140k$ iterations. Full details are provided in the supplementary material.

Observations: In Figure 4, we show the results of using different pre-training datasets and transfer datasets/tasks. Comparing Figures 4 (a) and (b), we make the following observations for the `Jigsaw` method:

- On the VOC07 classification task, pre-training on ImageNet-22k (14M images) transfers as well as pre-training on YFCC-100M (100M images).
- However, on the Places205 classification task, pre-training on YFCC-1M (1M images) transfers as well as pre-training on ImageNet-22k (14M images).

We note a similar trend for the `Colorization` problem wherein pre-training ImageNet, rather than YFCC, provides a greater benefit when transferring to VOC07 classification (also noted in [8, 12, 31]). A possible explanation for this benefit is that the domain (image distribution) of ImageNet is closer to VOC07 (both are object-centric) whereas YFCC is closer to Places205 (both are scene-centric). This motivates us to evaluate self-supervised methods on a variety of different domain/tasks and we propose an extensive evaluation suite next.

6. Benchmarking Suite for Self-supervision

We evaluate self-supervised learning on a diverse set of 9 tasks (see Table 1) ranging from semantic classification/detection, scene geometry to visual navigation. We select this benchmark based on the principle that a good representation should *generalize* to many different tasks with *limited* supervision and *limited* fine-tuning. We view self-supervised learning as a way to learn feature representations rather than an 'initialization method' [34] and thus perform limited fine-tuning of the features. We first describe each of these tasks and present our benchmarks.

Consistent Evaluation Setup: We believe that having a consistent evaluation setup, wherein hyperparameters are set fairly for all methods, is important for easier and meaningful comparisons across self-supervised methods. This is crucial to isolate the improvements due to better representations or better transfer optimization[1].

Method	`layer1`	`layer2`	`layer3`	`layer4`	`layer5`
ResNet-50 ImageNet-1k Supervised	14.8	32.6	42.1	50.8	52.5
ResNet-50 Places205 Supervised	16.7	32.3	43.2	54.7	62.3
ResNet-50 Random	12.9	16.6	15.5	11.6	9.0
ResNet-50 (NPID) [72]◁	18.1	22.3	29.7	42.1	45.5
ResNet-50 `Jigsaw` ImageNet-1k	15.1	28.8	36.8	41.2	34.4
ResNet-50 `Jigsaw` ImageNet-22k	11.0	30.2	36.4	41.5	36.4
ResNet-50 `Jigsaw` YFCC-100M	11.3	28.6	38.1	44.8	37.4
ResNet-50 `Coloriz.` ImageNet-1k	14.7	27.4	32.7	37.5	34.8
ResNet-50 `Coloriz.` ImageNet-22k	15.0	**30.5**	37.8	44.0	**41.5**
ResNet-50 `Coloriz.` YFCC-100M	**15.2**	30.4	**38.6**	**45.4**	**41.5**

Table 3: ResNet-50 top-1 center-crop accuracy for linear classification on Places205 dataset (§ 6.1). Numbers with ◁ use a different fine-tuning procedure. All other models follow the setup from Zhang *et al*. [75].

Common Setup (Pre-training, Feature Extraction and Transfer): The common transfer process for the benchmark tasks is as follows:

- First, we perform self-supervised **pre-training** using a self-supervised pretext method (`Jigsaw` or `Colorization`) on a pre-training dataset from Table 2.
- We **extract features** from various layers of the network. For AlexNet, we do this after every `conv` layer; for ResNet-50, we extract features from the last layer of every residual stage denoted, *e.g.*, `res1`, `res2` (notation from [26]) *etc*. For simplicity, we use the term `layer`.
- We then evaluate quality of these features (from different self-supervised approaches) by transfer learning, *i.e.*, benchmarking them on various **transfer** datasets and tasks that have supervision.

We summarize these benchmark tasks in Table 1 and discuss them in the subsections below. For each subsection, we provide *full details* of the training setup: model architecture, hyperparameters *etc*. in the supplementary material.

[1]We discovered inconsistencies across previous methods (different image crops for evaluation, weights re-scaling, pre-processing, longer fine-tuning schedules *etc*.) which affects the final performance.

Method	Places205				
	layer1	layer2	layer3	layer4	layer5
AlexNet ImageNet-1k Supervised	22.4	34.7	37.5	39.2	38.0
AlexNet Places205 Supervised	23.2	35.6	39.8	43.5	44.8
AlexNet Random	15.7	20.8	18.5	18.2	16.6
AlexNet (Jigsaw) [48]	19.7	26.7	31.9	32.7	30.9
AlexNet (Colorization) [74]	16.0	25.7	29.6	30.3	29.7
AlexNet (SplitBrain) [75]	21.3	30.7	34.0	34.1	32.5
AlexNet (Counting) [49]	23.3	33.9	36.3	34.7	29.6
AlexNet (Rotation) [24]◁	21.5	31.0	35.1	34.6	33.7
AlexNet (DeepCluster) [8]	17.1	28.8	35.2	36.0	32.2
AlexNet Jigsaw ImageNet-1k	23.7	33.2	36.6	36.3	31.9
AlexNet Jigsaw ImageNet-22k	**24.2**	**34.7**	37.7	37.5	31.7
AlexNet Jigsaw YFCC-100M	24.1	**34.7**	**38.1**	**38.2**	31.6
AlexNet Coloriz. ImageNet-1k	18.1	28.5	30.2	31.3	30.3
AlexNet Coloriz. ImageNet-22k	18.9	30.3	33.4	34.9	34.2
AlexNet Coloriz. YFCC-100M	18.4	30.0	33.4	34.8	**34.6**

Table 4: AlexNet top-1 center-crop accuracy for linear classification on Places205 dataset (§ 6.1). Numbers for [48, 74] are from [75]. Numbers with ◁ use a different fine-tuning schedule.

Method	layer1	layer2	layer3	layer4	layer5
ResNet-50 ImageNet-1k Supervised	24.5	47.8	60.5	80.4	88.0
ResNet-50 Places205 Supervised	28.2	46.9	59.1	77.3	80.8
ResNet-50 Random	9.6	8.3	8.1	8.0	7.7
ResNet-50 Jigsaw ImageNet-1k	**27.1**	45.7	56.6	64.5	57.2
ResNet-50 Jigsaw ImageNet-22k	20.2	**47.7**	57.7	**71.9**	**64.8**
ResNet-50 Jigsaw YFCC-100M	20.4	47.1	**58.4**	71.0	62.5
ResNet-50 Coloriz. ImageNet-1k	24.3	40.7	48.1	55.6	52.3
ResNet-50 Coloriz. ImageNet-22k	25.8	43.1	53.6	66.1	62.7
ResNet-50 Coloriz. YFCC-100M	26.1	42.3	53.8	67.2	61.4

Table 5: ResNet-50 Linear SVMs mAP on VOC07 classification (§ 6.1).

6.1. Task 1: Image Classification

We extract image features from various layers of a self-supervised network and train linear classifiers on these fixed representations. We evaluate performance on the classification task for three datasets: Places205, VOC07 and COCO2014. We report results for ResNet-50 in the main paper; AlexNet results are in the supplementary material.

Places205: We strictly follow the training and evaluation setup from Zhang *et al*. [75] so that we can draw comparisons to existing works (and re-evaluate the model from [8]). We use a batchsize of 256, learning rate of 0.01 decayed by a factor of 10 after every $40k$ iterations, and train for $140k$ iterations using SGD on the `train` split. We report the top-1 center-crop accuracy on the `val` split for ResNet-50 in Table 3 and AlexNet in Table 4.

VOC07 and COCO2014: For smaller datasets that fit in memory, we follow [53] and train linear SVMs [7] on the frozen feature representations using LIBLINEAR package [20]. We train on `trainval` split of VOC07 dataset, and evaluate on `test` split of VOC07. Table 5 shows results on VOC07 for ResNet-50. AlexNet and COCO2014 [40] results are provided in the supplementary material.

Observations: We see a significant accuracy gap between self-supervised and supervised methods despite our scaling efforts. This is expected as unlike self-supervised methods, both the supervised pre-training and benchmark transfer tasks solve a semantic image classification problem.

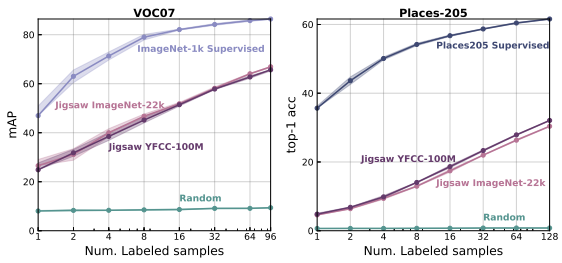

Figure 5: Low-shot Image Classification on the VOC07 and Places205 datasets using linear SVMs trained on the features from the best performing layer for ResNet-50. We vary the number of labeled examples (per class) used to train the classifier and report the performance on the `test` set. We show the mean and standard deviation across five runs (§ 6.2).

6.2. Task 2: Low-shot Image Classification

It is often argued that a good representation should not require many examples to learn about a concept. Thus, following [71], we explore the quality of feature representation when per-category examples are few (unlike § 6.1).

Setup: We vary the number k of positive examples (per class) and use the setup from § 6.1 to train linear SVMs on Places205 and VOC07 datasets. We perform this evaluation for ResNet-50 only. For each combination of k/dataset/method, we report the mean and standard deviation of 5 independent samples of the training data evaluated on a fixed test set (`test` split for VOC07 and `val` split for Places205). We show results for the `Jigsaw` method in Figure 5; `Colorization` results are in the supplementary material as we draw the same observations.

Observations: We report results for the best performing layer `res4` (notation from [26]) for ResNet-50 on VOC07 and Places205 in Figure 5. In the supplementary material, we show that for the lower layers, similar to Table 3, the self-supervised features are competitive to their supervised counterpart in low-shot setting on both the datasets. However, for both VOC07 and Places205, we observe a significant gap between supervised and self-supervised settings on their 'best' performing layer. This gap is much larger at lower sample size, *e.g.*, at $k=1$ it is 30 points for Places205, whereas at higher values (full-shot in Table 3) it is 20 points.

6.3. Task 3: Visual Navigation

In this task, an agent receives a stream of images as input and learns to navigate to a pre-defined location to get a reward. The agent is spawned at random locations and must build a contextual map in order to be successful at the task.

Setup: We use the setup from [59] who train an agent using reinforcement learning (PPO [60]) in the Gibson environment [73]. The agent uses *fixed* feature representations from a ConvNet for this task and only updates the policy network.

Method	VOC07	VOC07+12
ResNet-50 ImageNet-1k Supervised*	66.7 ± 0.2	71.4 ± 0.1
ResNet-50 ImageNet-1k Supervised	**68.5** ± 0.3	**75.8** ± 0.2
ResNet-50 Places205 Supervised	65.3 ± 0.3	73.1 ± 0.3
ResNet-50 `Jigsaw` ImageNet-1k	56.6 ± 0.5	64.7 ± 0.2
ResNet-50 `Jigsaw` ImageNet-22k	**67.1** ± 0.3	**73.0** ± 0.2
ResNet-50 `Jigsaw` YFCC-100M	62.3 ± 0.2	69.7 ± 0.1

Table 6: Detection mAP for frozen `conv` body on VOC07 and VOC07+12 using Fast R-CNN with ResNet-50-C4 (mean and std computed over 5 trials). We freeze the `conv` body for all models. Numbers with * use `Detectron` [26] default training schedule. All other models use slightly longer training schedule (see § 6.4).

We evaluate the representation of layers `res3`, `res4`, `res5` (notation from [26]) of a ResNet-50 by separately training agents for these settings. We use the training hyperparameters from [59], who use a rollout of size 512 and optimize using Adam [32].

Observations: Figure 6 shows the average training reward (and variance) across 5 runs. Using the `res3` layer features, we observe that our `Jigsaw` ImageNet model gives a much **higher training reward** and is more **sample efficient** (higher reward with fewer steps) than its supervised counterpart. The deeper `res4` and `res5` features perform similarly for the supervised and self-supervised networks. We also observe that self-supervised pre-training on the ImageNet domain outperforms pre-training on the YFCC domain.

6.4. Task 4: Object Detection

Setup: We use the `Detectron` [26] framework to train the Fast R-CNN [25] object detection model using Selective Search [66] object proposals on the VOC07 and VOC07+12 [18] datasets. We provide results for Faster R-CNN [55] in the supplementary material. We note that we use the *same training schedule* for both the supervised and self-supervised methods since it impacts final object detection performance significantly. We report mean and standard deviation result of 5 independent runs for ResNet-50 only as `Detectron` does not support AlexNet.

We freeze the full `conv` body of Fast R-CNN and only train the RoI heads (last ResNet-50 stage `res5` onwards). We follow the same setup as in `Detectron` and only change the training schedule to be slightly longer. Specifically, we train on 2 GPUS for $22k/8k$ schedule on VOC07 and for $66k/14k$ schedule on VOC07+12 (compared to original $15k/5k$ schedule on VOC07 and $40k/15k$ schedule on VOC07+12). This change improves object detection performance for both supervised and self-supervised methods.

Observations: We report results in Table 6 and note that the self-supervised initialization is competitive with the ImageNet pre-trained initialization on VOC07 dataset even when fewer parameters are fine-tuned on the detection task. We also highlight that the performance gap between supervised and self-supervised initialization is very low.

6.5. Task 5: Surface Normal Estimation

Setup: We use the surface normal estimation task [22], with the evaluation, and dataset splits as formulated in [5, 45, 68]. We use the NYUv2 [61] dataset which consists of indoor scenes and use the surface normals calculated by [36]. We use the state-of-the-art PSPNet [76] architecture (implementation [78]). This provides a much stronger baseline (our scratch model outperforms the best numbers reported in [70]). We fine-tune `res5` onwards and train all the models with the same hyperparameters for 150 epochs. The scratch model (initialized randomly) is trained for 400 epochs. We use the training hyperparameters from [78], *i.e.*, batchsize of 16, learning rate of 0.02 decayed polynomially with a power of 0.9 and optimize using SGD.

Observations: We report the best `test` set performance for `Jigsaw` in Table 7 and results for `Colorization` are provided in the supplementary material. We use the metrics from [22] which measure the angular distance (error) of the prediction as well as the percentage of pixels within t° of the ground truth. We note that our `Jigsaw` YFCC-100M self-supervised model **outperforms both** the supervised models (ImageNet-1k and Places205 supervised) across all the metrics by a significant margin, *e.g.*, a **5 point** gain compared to the Places205 supervised model on the number of pixels within $t^\circ = 11.5$ metric. We, thus, conclude that self-supervised methods provide better features compared to supervised methods for 3D geometric tasks.

Initialization	Angle Distance (Lower is better)		Within t° (Higher is better)		
	Mean	Median	11.25	22.5	30
ResNet-50 ImageNet-1k supervised	26.4	17.1	36.1	59.2	68.5
ResNet-50 Places205 supervised	23.3	14.2	41.8	65.2	73.6
ResNet-50 Scratch	26.3	16.1	37.9	60.6	69.0
ResNet-50 `Jigsaw` ImageNet-1k	24.2	14.5	41.2	64.2	72.5
ResNet-50 `Jigsaw` ImageNet-22k	22.6	13.4	43.7	66.8	74.7
ResNet-50 `Jigsaw` YFCC-100M	**22.4**	**13.1**	**44.6**	**67.4**	**75.1**

Table 7: Surface Normal Estimation on the NYUv2 dataset. We train ResNet-50 from `res5` onwards and freeze the `conv` body below (§ 6.5).

7. Legacy Tasks and Datasets

For completeness, we also report results on the evaluation tasks used by previous works. As we explain next, we do not include these tasks in our benchmark suite (§6).

Full fine-tuning for transfer learning: This setup fine-tunes all parameters of a self-supervised network and views it as an initialization method. We argue that this view evaluates not only the quality of the representations but also the initialization and optimization method. For completeness, we report results for AlexNet and ResNet-50 on VOC07 classification in the supplementary material.

VOC07 Object Detection with Full Fine-tuning: This task fine-tunes *all* the weights of a network for the object detection task. We use the same settings as in § 6.4 and

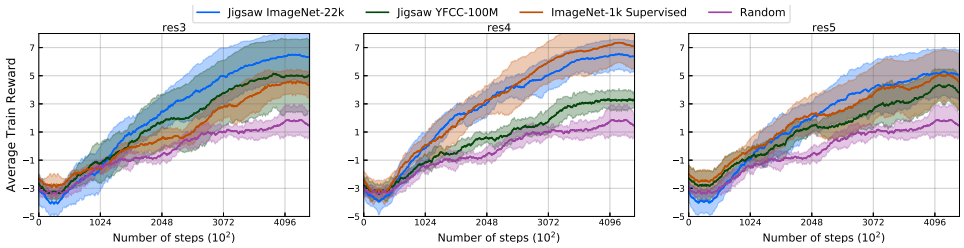

Figure 6: Visual Navigation. We train an agent on the navigation task in the Gibson environment. The agent is trained using reinforcement learning and uses fixed ConvNet features. We show results for different layers features of ResNet-50 trained on both supervised and self-supervised settings (§ 6.3).

Method	VOC07	VOC07+12
ResNet-50 ImageNet-1k Supervised*	69.1 ± 0.4	76.2 ± 0.4
ResNet-50 ImageNet-1k Supervised	**70.5** ± 0.4	**76.2** ± 0.1
ResNet-50 Places205 Supervised	67.2 ± 0.2	74.5 ± 0.4
ResNet-50 `Jigsaw` ImageNet-1k	61.4 ± 0.2	68.3 ± 0.4
ResNet-50 `Jigsaw` ImageNet-22k	**69.2** ± 0.3	**75.4** ± 0.2
ResNet-50 `Jigsaw` YFCC-100M	66.6 ± 0.1	73.3 ± 0.4

Table 8: Detection mAP for full fine-tuning on VOC07 and VOC07+12 using Fast R-CNN with ResNet-50-C4 (mean and std computed over 5 trials) (§7). Numbers with * use `Detectron` [26] default training schedule. All other models use a slightly longer training schedule.

Method	ImageNet-1k				
	`layer1`	`layer2`	`layer3`	`layer4`	`layer5`
AlexNet ImageNet-1k Supervised	19.4	37.1	42.5	48.0	49.6
AlexNet Places205 Supervised	18.9	35.5	38.9	40.9	37.3
AlexNet Random	11.9	17.2	15.2	14.8	13.5
AlexNet (Jigsaw) [48]	16.2	23.3	30.2	31.7	29.6
AlexNet (Colorization) [74]	13.1	24.8	31.0	32.6	31.8
AlexNet (SplitBrain) [75]	17.7	29.3	35.4	35.2	32.8
AlexNet (Counting) [49]	**23.3**	**33.9**	36.3	34.7	29.6
AlexNet (Rotation) [24]◁	18.8	31.7	38.7	38.2	36.5
AlexNet (DeepCluster) [8]	13.4	28.5	37.4	**39.2**	35.7
AlexNet `Jigsaw` ImageNet-1k	20.2	32.9	36.5	36.1	29.2
AlexNet `Jigsaw` ImageNet-22k	20.2	**33.9**	**38.7**	37.9	27.5
AlexNet `Jigsaw` YFCC-100M	20.2	33.4	38.1	37.4	25.8
AlexNet `Coloriz.` ImageNet-1k	14.1	27.5	30.6	32.1	31.1
AlexNet `Coloriz.` ImageNet-22k	15.0	30.5	35.5	37.9	**37.4**
AlexNet `Coloriz.` YFCC-100M	14.4	28.8	33.2	35.3	34.0

Table 9: AlexNet top-1 center-crop accuracy for linear classification on ImageNet-1k. Numbers for [48, 74] are from [75]. Numbers with ◁ use a different fine-tuning schedule.

report results for supervised and `Jigsaw` self-supervised methods in Table 8. Without any bells and whistles, our self-supervised model initialization **matches the performance** of the supervised initialization on both VOC07 and VOC07+12. We note that self-supervised pre-training on ImageNet performs better than YFCC (similar to §5).

ImageNet Classification using Linear Classifiers: While the task itself is meaningful, we do not include it in our benchmark suite for two reasons:

1. For supervised representations, the widely used baseline is trained on ImageNet-1k dataset. Hence, evaluating also on the same dataset (ImageNet-1k) does *not* test generalization of the supervised baseline.
2. Most existing self-supervised approaches [12, 75] use ImageNet-1k for pre-training and evaluate the representations on the same dataset. As observed in §5, pre-training and evaluating in the *same* domain biases evaluation. Further, the bias is accentuated as we pre-train the self-supervised features and learn the linear classifiers (for transfer) on *identical* images.

To compare with existing methods, we report results on ImageNet-1k classification for AlexNet in Table 9 (setup from § 6.1). We report results on ResNet-50 in the supplementary material.

8. Conclusion

In this work, we studied the effect of scaling two self-supervised approaches along three axes: data size, model capacity and problem complexity. Our results indicate that transfer performance increases log-linearly with the data size. The quality of the representations also improves with higher capacity models and problem complexity. More interestingly, we observe that the performance improvements on the the three axes are complementary (§4). We obtain **state-of-the-art** results on linear classification using the ImageNet-1k and Places205 datasets. We also propose a benchmark suite of 9 diverse tasks to evaluate the quality of our learned representations. Our self-supervised learned representation: (a) **outperforms supervised** baseline on task of surface normal estimation; (b) performs competitively (or better) compared to supervised-ImageNet baseline on navigation task; (c) **matches the supervised object detection** baseline even with little fine-tuning; (d) performs worse than supervised counterpart on task of image classification and low-shot classification. We believe future work should focus on designing tasks that are complex enough to exploit large scale data and increased model capacity. Our experiments suggest that scaling self-supervision is crucial but there is still a long way to go before definitively surpassing supervised pre-training.

Acknowledgements: We would like to thank Richard Zhang, Mehdi Noroozi, and Andrew Owens for helping understand the experimental setup in their respective works. Rob Fergus and Léon Bottou for helpful discussions and valuable feedback. Alexander Sax, Bradley Emi, and Saurabh Gupta for helping with the Gibson experiments; Aayush Bansal and Xiaolong Wang for their help in the surface normal experiments. Ross Girshick and Piotr Dollár for helpful comments on the manuscript.

References

[1] The Gelato Bet. https://people.eecs.berkeley.edu/~efros/gelato_bet.html. Accessed: 2019-03-20. 1

[2] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 2

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

[4] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 2

[5] Aayush Bansal, Bryan Russell, and Abhinav Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *CVPR*, pages 5965–5974, 2016. 7

[6] Piotr Bojanowski and Armand Joulin. Unsupervised learning by predicting noise. In *ICML*, 2017. 2

[7] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. 3, 6

[8] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2, 5, 6, 8

[9] Virginia R de Sa. Learning classification with unlabeled data. In *NIPS*, 1994. 2

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. 2009. 1, 3, 4

[11] Aditya Deshpande, Jason Rock, and David Forsyth. Learning large-scale automatic image colorization. In *ICCV*, 2015. 1, 2

[12] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 2, 5, 8

[13] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012. 2

[14] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 1, 2

[15] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016. 2

[16] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9):1734–1747, 2016. 2

[17] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 2

[18] 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. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015. 7

[19] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2), 2010. 3

[20] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *JMLR*, 9:1871–1874, 2008. 6

[21] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2

[22] David F Fouhey, Abhinav Gupta, and Martial Hebert. Data-driven 3d primitives for single image understanding. In *ICCV*, 2013. 7

[23] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018. 2

[24] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018. 2, 6, 8

[25] Ross Girshick. Fast r-cnn. In *ICCV*, 2015. 7

[26] Ross Girshick, Ilija Radosavovic, Georgia Gkioxari, Piotr Dollár, and Kaiming He. Detectron, 2018. 3, 5, 6, 7, 8

[27] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 3

[29] Fu Jie Huang, Y-Lan Boureau, Yann LeCun, et al. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007. 2

[30] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015. 2

[31] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *ECCV*, 2016. 3, 5

[32] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[33] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. *arXiv preprint arXiv:1901.09005*, 2019. 1, 2

[34] Philipp Krähenbühl, Carl Doersch, Jeff Donahue, and Trevor Darrell. Data-dependent initializations of convolutional neural networks. *arXiv preprint arXiv:1511.06856*, 2015. 5

[35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3

[36] L Ladickỳ, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, 2014. 7

[37] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016. 1, 2

[38] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017. 2

[39] 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. 1

[40] 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. 6

[41] Pauline Luc, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun. Predicting deeper into the future of semantic segmentation. In *ICCV*, 2017. 2

[42] 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. 1, 3

[43] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International Conference on Artificial Neural Networks*, pages 52–59. Springer, 2011. 2

[44] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *ICML*, 2017. 2

[45] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016. 7

[46] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 2

[47] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, 2009. 2

[48] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1, 2, 4, 6, 8

[49] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 2, 6, 8

[50] Mehdi Noroozi, Ananth Vinjimoor, Paolo Favaro, and Hamed Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018. 2

[51] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607, 1996. 2

[52] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[53] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016. 2, 3, 6

[54] Deepak Pathak, Ross Girshick, Piotr Dollár, Trevor Darrell, and Bharath Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 2

[55] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 7

[56] 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. 3, 4

[57] Bryan C Russell, William T Freeman, Alexei A Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[58] Ruslan Salakhutdinov and Geoffrey Hinton. Deep boltzmann machines. In *Artificial intelligence and statistics*, pages 448–455, 2009. 2

[59] Alexander Sax, Bradley Emi, Amir R Zamir, Leonidas Guibas, Silvio Savarese, and Jitendra Malik. Mid-level visual representations improve generalization and sample efficiency for learning active tasks. *arXiv preprint arXiv:1812.11971*, 2018. 6, 7

[60] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 6

[61] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*. Springer, 2012. 7

[62] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *ECCV*. 2012. 2

[63] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 2

[64] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 1, 3

[65] 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. 1, 3

[66] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 7

[67] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*. ACM, 2008. 2

[68] Xiaolong Wang, David Fouhey, and Abhinav Gupta. Designing deep networks for surface normal estimation. In *CVPR*, 2015. 7

[69] Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 2

[70] Xiaolong Wang, Kaiming He, and Abhinav Gupta. Transitive invariance for self-supervised visual representation learning. In *ICCV*, pages 1329–1338, 2017. 2, 7

[71] Yu-Xiong Wang and Martial Hebert. Learning to learn: Model regression networks for easy small sample learning. In *ECCV*, 2016. 6

[72] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 1, 2, 5

[73] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *CVPR*, 2018. 6

[74] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 1, 2, 3, 6, 8

[75] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 4, 5, 6, 8

[76] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 7

[77] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 4

[78] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 2018. 7

[79] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017. 2