Edinburgh Research Explorer

How Well Do Self-Supervised Models Transfer?

How Well Do Self-Supervised Models Transfer?

Linus Ericsson
University of Edinburgh

Henry Gouk
University of Edinburgh

Timothy M. Hospedales
University of Edinburgh,
Samsung AI Research, Cambridge

linus.ericsson@ed.ac.uk   henry.gouk@ed.ac.uk   t.hospedales@ed.ac.uk
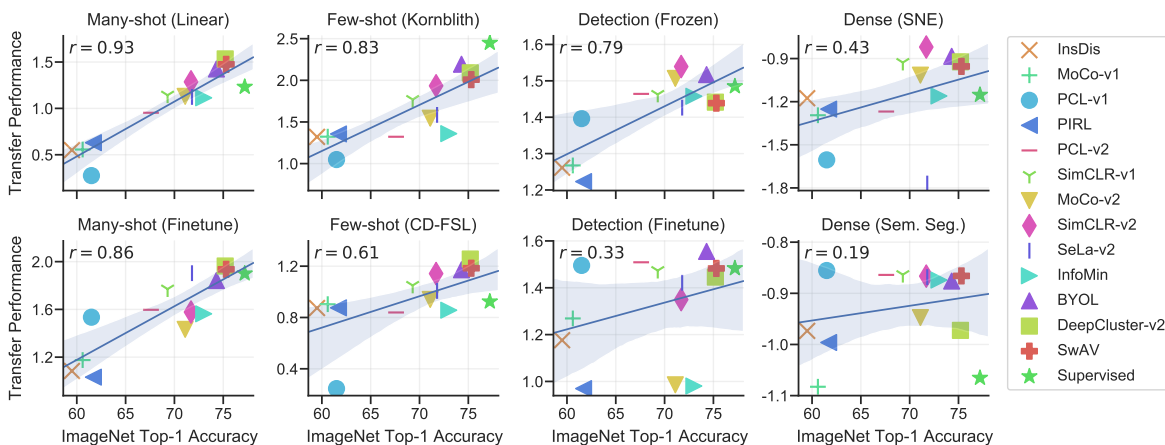
Figure 1. Transfer performance is highly correlated with ImageNet performance for many-shot recognition but increasingly less correlated for few-shot recognition, object detection and dense prediction. On the x-axes we plot ImageNet top-1 accuracy and on the y-axes the average transfer log-odds. The gradients of the regression lines describe the correlation, with confidence intervals in shaded areas. For perfect correlation, the ideal line is a positive slope diagonal. Correlation coefficients (Pearson's r) are shown in the top left of each plot.

Abstract

Self-supervised visual representation learning has seen huge progress recently, but no large scale evaluation has compared the many models now available. We evaluate the transfer performance of 13 top self-supervised models on 40 downstream tasks, including many-shot and few-shot recognition, object detection, and dense prediction. We compare their performance to a supervised baseline and show that on most tasks the best self-supervised models outperform supervision, confirming the recently observed trend in the literature. We find ImageNet Top-1 accuracy to be highly correlated with transfer to many-shot recognition, but increasingly less so for few-shot, object detection and dense prediction. No single self-supervised method dominates overall, suggesting that universal pre-training is still unsolved. Our analysis of features suggests that top self-supervised learners fail to preserve colour information as well as supervised alternatives, but tend to induce better classifier calibration, and less attentive overfitting than supervised learners.

1. Introduction

Computer vision in the last decade has been driven by increasingly sophisticated convolutional neural networks (CNNs) and the increasingly large datasets used to train them. Nevertheless, progress in this paradigm is ultimately bottlenecked by the data annotation process. This has motivated a growing wave of research in self-supervised representation learning, where CNN representations are trained on pretext tasks with freely available labels. Once trained, these CNN representations can be used to learn new tasks more data efficiently through feature re-use or finetuning.

Self-supervised learning (SSL) has been around for some time [47], but historically has lagged behind state of the art supervised representation learning. However, the recent pace of progress has increased dramatically and led to self-supervised deep representations that appear to approach and possibly even surpass that of fully-supervised representations [17, 5]. This has raised hopes that self-supervised methods could indeed replace the ubiquitous annotation-intensive paradigm of supervised deep learning in state of the art computer vision going forward.

Given the growing practical importance of self-supervised learning as it approaches state of the art in computer vision tasks, there is increasing interest in understanding and benchmarking its empirical performance. Major recent evaluation studies have looked at aspects such as the fit between CNN architectures and choice of pretext task [27] and the impact of the pre-training set size and CNN capacity on downstream task performance [16].

Despite this initial progress, there are a number of important open questions that remain to to be understood. Firstly, given the plethora of self-supervised representations on the market using diverse pre-text tasks and data-augmentations: which methods are the most empirically effective? This is currently hard to assess given the limited commonality in the evaluation conditions reported by each method. Secondly: While the most widely adopted benchmark metric is image classification performance, there are hopes that pre-trained representations will generalise to other downstream tasks such as detection and dense prediction [16]. However, the published self-supervision literature is particularly inconsistent with regard to benchmarking these alternative tasks, making it impossible to determine the most effective methods. In particular, while we hope that the methods with best performance on the most popular benchmark of ImageNet recognition will also perform well on alternative tasks, this conjecture has never been systematically tested empirically. Thirdly: While core academic vision research is happy to focus on ImageNet as a benchmark, the wider community of computer vision practitioners work with diverse data types from medical [54] to agricultural [40], to earth-observation [24] data and beyond. From this perspective a crucial question is to what extent self-supervised features pre-trained on ImageNet can generalise directly to these diverse downstream tasks? This is important to know practically, because it dictates whether users in different vision domains can use pre-trained features directly, or whether they would need to collect their own datasets and perform domain-specific self-supervised learning – a major data, compute and environmental [48] hurdle given that state of the art methods can take around 20 GPU days to train [8]. Academically, this is also important to know, as an indicator of whether pursuing higher ImageNet accuracy in self-supervised learning research leads to higher accuracy on diverse real-world vision tasks, or is our research overfitting to ImageNet recognition?

To answer these questions and more, we conduct a large empirical benchmarking study on the efficacy of different pre-trained representations for diverse downstream tasks. In particular, we evaluate 13 pre-trained self-supervised models on 40 transfer tasks covering many-shot and few-shot image classification, object detection, surface normal prediction and semantic segmentation, as summarised in Fig. 1. Our downstream tasks cover diverse datasets with a wide range of similarity to the source ImageNet data, which all our models were pre-trained on.

Among other questions, we aim to answer the following:

Q1. *How do state of the art self-supervised methods compare to supervised feature learning for diverse downstream datasets and tasks?* A: The best self-supervised methods can match and outperform supervised representation learning across most tasks considered. Only in few-shot recognition with small domain shift to ImageNet does supervised representation learning win.

Q2. *Do self-supervised representations that perform well on ImageNet classification systematically perform well on diverse downstream datasets and tasks?* A: For recognition on datasets similar to ImageNet, performance is highly correlated. However, for some of the least similar recognition datasets such as ISIC2018, there is little to no correlation with ImageNet performance. For different tasks such as detection and dense prediction, correlation exists but is lower than for recognition.

Q3. *Is there a best self-supervised representation overall?* A: No. For example, the recent methods SwAV and DeepCluster-v2 work well for recognition on ImageNet-like data, but under-perform on non-recognition tasks and on different data such as medical skin images. This suggests that the vision of a universal pre-trained model suited for all downstream tasks is yet to be realised.

Q4. *Do self-supervised and supervised features represent the same information?* A: Contemporary self-supervised features seem to discard colour information, presumably due to the data augmentation they use. They also tend to be more attentively diffuse in contrast to the high spatial focus of attention in supervised features, which may contribute to their improved uncertainty calibration.

2. Related Work

Self-supervised learning Self-supervised representation learning is now a large topic that it is impossible to cover completely here, and we point the reader to excellent recent surveys [26, 37] for thorough reviews. In this paper, we focus on still-image self-supervised learning, where a common paradigm is to pre-train on ImageNet [11] using a variety of pre-text tasks from jigsaw puzzles [42] to colorization [65, 33] to instance discrimination [58, 12, 6, 21] and clustering [34, 5]. Evaluation is then typically performed by using the learned representation to train a linear classifier on ImageNet [21], or finetune the representation with a small amount of data [7]. However, evaluation of the impact on different downstream datasets (where there is domain shift [67] with respect to ImageNet), and non-recognition tasks has been highly inconsistent – a gap in the literature that we aim to remedy in this paper.

To do this we wish to evaluate a large number of self-supervised methods, covering a wide range of training objectives. Many recent works adopt a form of instance discrimination [12, 58, 39], whereby each training image is treated as its own class. By applying strong data augmentation to these images, and comparing them using a contrastive [20, 51, 25] loss, a model can learn features which

Table 1. Top self-supervised models beat the supervised pre-training baseline on popular many-shot recognition datasets, both in linear evaluation and when finetuning. The top half of the table shows results from linear transfer of pre-trained models using logistic regression, and the bottom half shows the results when these models are finetuned. We also include the ImageNet linear evaluation performance (logistic regression or SGD) reported by the authors. Results style: **best**, <u>second best</u>.

		ImageNet	Aircraft	Caltech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food	Pets	SUN397	VOC2007	Avg.
Linear	InsDis	59.50	36.87	71.12	28.98	80.28	59.97	68.46	83.44	63.39	68.78	49.47	74.37	62.29
	MoCo-v1	60.60	35.55	75.33	27.99	80.16	57.71	68.83	82.10	62.10	69.84	51.02	75.93	62.41
	PCL-v1	61.50	21.61	76.90	12.93	81.84	55.74	62.87	64.73	48.02	75.34	45.70	78.31	56.73
	PIRL	61.70	37.08	74.48	28.72	82.53	61.26	68.99	83.60	64.65	71.36	53.89	76.61	63.92
	PCL-v2	67.60	37.03	86.42	30.51	91.91	73.54	70.59	85.34	64.88	82.79	56.25	81.14	69.13
	SimCLR-v1	69.30	44.90	90.05	43.73	91.18	72.73	74.20	90.87	67.47	83.33	59.21	80.77	72.59
	MoCo-v2	71.10	41.79	87.92	39.31	92.28	74.90	73.88	90.07	68.95	83.30	60.32	82.69	72.31
	SimCLR-v2	71.70	46.38	89.63	50.37	92.53	76.78	76.38	92.90	73.08	84.72	61.47	81.57	75.07
	SeLa-v2	71.80	37.29	87.20	36.86	92.73	74.81	74.15	90.22	71.08	83.22	62.71	82.73	72.09
	InfoMin	73.00	38.58	87.84	41.04	91.49	73.43	74.73	87.18	69.53	86.24	61.00	83.24	72.21
	BYOL	74.30	53.87	**91.46**	<u>56.40</u>	93.26	77.86	76.91	94.50	73.01	89.10	59.99	81.14	77.05
	DeepCluster-v2	75.20	**54.49**	<u>91.33</u>	**58.60**	**94.02**	**79.61**	**78.62**	**94.72**	**77.94**	<u>89.36</u>	<u>65.48</u>	**83.94**	**78.92**
	SwAV	<u>75.30</u>	<u>54.04</u>	90.84	54.06	<u>93.99</u>	<u>79.58</u>	<u>77.02</u>	<u>94.62</u>	<u>76.62</u>	87.60	**65.58**	<u>83.68</u>	<u>77.97</u>
	Supervised	**77.20**	43.59	90.18	44.92	91.42	73.90	72.23	89.93	69.49	**91.45**	60.49	83.60	73.75
Finetune	InsDis		73.38	72.04	61.56	93.32	68.26	63.99	89.51	76.78	76.22	51.84	71.90	72.62
	MoCo-v1		75.61	74.95	65.02	93.89	71.52	65.37	89.45	77.28	76.96	53.35	74.91	74.39
	PCL-v1		74.97	87.62	73.24	96.35	79.62	70.00	90.83	78.30	86.98	58.40	82.08	79.85
	PIRL		72.68	70.83	61.02	92.23	66.48	64.26	89.81	74.96	76.26	50.38	69.90	71.71
	PCL-v2		79.37	88.04	71.68	96.50	80.26	71.76	92.95	80.34	85.39	58.82	82.20	80.66
	SimCLR-v1		81.06	90.35	83.78	**97.07**	<u>84.53</u>	71.54	93.75	82.40	84.10	63.31	82.58	83.13
	MoCo-v2		79.87	84.38	75.20	96.45	71.33	69.47	94.35	76.78	79.80	55.77	71.71	77.74
	SimCLR-v2		78.71	82.94	79.84	96.22	79.05	70.16	94.32	82.22	83.20	61.12	78.19	80.54
	SeLa-v2		81.99	88.99	85.62	96.80	84.37	74.36	**95.80**	86.24	88.55	65.84	<u>84.85</u>	84.86
	InfoMin		80.24	83.92	78.76	96.94	71.15	71.12	95.24	78.93	85.28	57.66	76.63	79.62
	BYOL		79.45	89.40	84.60	97.01	83.95	73.62	94.48	85.54	<u>89.62</u>	63.96	82.70	84.03
	DeepCluster-v2		82.52	<u>90.75</u>	**87.27**	<u>97.06</u>	**85.15**	<u>74.84</u>	95.31	**87.51**	89.43	**66.42**	**84.90**	**85.56**
	SwAV		<u>83.08</u>	89.85	<u>86.76</u>	96.78	84.37	**75.16**	95.46	<u>87.22</u>	89.05	<u>66.24</u>	84.66	<u>85.33</u>
	Supervised		**83.50**	**91.01**	82.61	96.39	82.91	73.30	<u>95.50</u>	84.60	**92.42**	63.56	84.76	84.60

are resilient to various changes in view. The main difficulty in instance discrimination lies in approximating the loss over all instances, as it becomes intractable for large datasets. This leads to metric learning methods which require large numbers of pairwise comparisons. The scaling problem that still remains has been tackled by using memory banks of features [58], momentum encoders [21] or very large batches [6]. On the other side, clustering-based approaches [4, 1] compare groups of images with similar features, sidestepping the intractability of instance discrimination. The problem here instead is computing the cluster assignments over the entire training set. These approaches therefore tend to focus on ways of performing this assignment online [64, 5]. Among recent methods, BYOL stands out as one which does not directly use either a contrastive or clustering approach, but as noted by [53], an implicit contrastive loss term is created by their use of batch normalisation. In this paper, we evaluate methods using all of the above approaches, investigating the effect of training objective on transfer performance and representation quality.

Prior evaluations and benchmarks The importance of empirical evaluation of general purpose representation learning is highlighted by the growing number of major evaluation papers in this area [28, 16, 63, 27]. In terms of transfer performance from supervised pre-training, [28] proposes a suite of downstream recognition task evaluations and evaluates transfer performance of several supervised models of varying architecture and pre-training details. They find very strong correlations between ImageNet performance and transfer performance on down-

stream tasks. In contrast, we compare pre-trained models of exactly the same (ResNet-50) architecture, and instead evaluate the impact of the different training objectives and augmentation strategies used by self-supervised learners; as well as considering a more diverse suite of downstream benchmarks including few-shot recognition, object detection and dense prediction. Our results are more nuanced, with high correlation visible in recognition tasks similar to ImageNet and lower correlation elsewhere. [16] propose a richer range of downstream benchmarks to evaluate self-supervised pre-training, but focus on the impact of different pre-training datasets and CNN architectures. In contrast, we provide the first comprehensive comparison of different self-supervised algorithms, holding architecture and dataset constant. [27] compares a few architectures and SSL algorithms on a small number of downstream tasks, and draw observations such as pre-text task performance being uncorrelated with representation performance on ImageNet recognition. In contrast, we evaluate whether performance on the commonly evaluated ImageNet recognition is indicative of in-the-wild performance on diverse downstream datasets and non-recognition tasks. The evaluation in [63] finds that self-supervised methods can not beat supervised models. We find that a more recent family of self-supervised learners consistently achieve the highest performances, on recognition, detection, surface normal estimation and semantic segmentation, with the one exception of few-shot recognition on ImageNet-like data.

Table 2. Few-shot transfer (5-way 20-shot) of pre-trained models using prototypical networks on popular recognition datasets. Results style: **best**, <u>second best</u>.

	Aircraft	Caltech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food	Pets	SUN397
InsDis	48.67 ± 0.93	89.20 ± 0.50	55.18 ± 0.69	70.16 ± 0.56	75.17 ± 0.68	82.02 ± 0.50	93.76 ± 0.36	70.67 ± 0.64	82.96 ± 0.57	90.81 ± 0.43
MoCo-v1	48.76 ± 0.93	91.45 ± 0.43	53.04 ± 0.70	66.74 ± 0.55	72.68 ± 0.70	83.08 ± 0.50	93.60 ± 0.35	71.21 ± 0.65	83.68 ± 0.58	90.89 ± 0.45
PCL-v1	43.31 ± 0.86	87.51 ± 0.49	47.44 ± 0.75	68.16 ± 0.53	69.90 ± 0.75	74.41 ± 0.62	82.75 ± 0.64	65.38 ± 0.69	89.90 ± 0.52	86.40 ± 0.48
PIRL	49.69 ± 0.92	90.41 ± 0.46	55.82 ± 0.68	71.23 ± 0.55	75.99 ± 0.70	81.98 ± 0.51	93.72 ± 0.35	70.09 ± 0.66	83.61 ± 0.55	91.20 ± 0.45
PCL-v2	37.68 ± 0.76	88.99 ± 0.45	49.46 ± 0.73	78.22 ± 0.47	80.63 ± 0.59	81.22 ± 0.54	91.81 ± 0.39	69.75 ± 0.66	89.17 ± 0.52	89.37 ± 0.44
SimCLR-v1	53.55 ± 0.91	95.87 ± 0.28	63.95 ± 0.78	78.10 ± 0.52	82.97 ± 0.59	84.24 ± 0.46	95.69 ± 0.29	74.10 ± 0.61	91.90 ± 0.43	93.83 ± 0.33
MoCo-v2	39.64 ± 0.77	91.87 ± 0.40	57.67 ± 0.76	76.65 ± 0.48	81.30 ± 0.63	84.57 ± 0.50	94.31 ± 0.33	74.39 ± 0.64	91.78 ± 0.43	92.34 ± 0.39
SimCLR-v2	53.93 ± 0.94	96.97 ± 0.22	64.25 ± 0.76	79.50 ± 0.53	86.33 ± 0.55	86.42 ± 0.43	96.55 ± 0.24	78.88 ± 0.57	92.24 ± 0.42	95.07 ± 0.30
SeLa-v2	40.75 ± 0.86	92.67 ± 0.51	57.12 ± 0.77	77.67 ± 0.51	82.42 ± 0.64	85.85 ± 0.45	93.86 ± 0.34	77.26 ± 0.62	88.19 ± 0.51	94.50 ± 0.33
InfoMin	38.64 ± 0.75	89.12 ± 0.46	57.58 ± 0.79	72.90 ± 0.52	77.25 ± 0.64	80.90 ± 0.53	91.60 ± 0.40	73.99 ± 0.63	91.06 ± 0.45	90.39 ± 0.45
BYOL	<u>62.65 ± 0.92</u>	<u>98.38 ± 0.15</u>	71.01 ± 0.75	78.73 ± 0.50	85.92 ± 0.56	**87.56 ± 0.45**	**97.88 ± 0.19**	80.07 ± 0.56	<u>95.71 ± 0.31</u>	95.36 ± 0.29
DeepCluster-v2	54.68 ± 0.93	97.06 ± 0.22	69.50 ± 0.77	<u>81.08 ± 0.49</u>	<u>86.52 ± 0.54</u>	87.56 ± 0.42	<u>97.51 ± 0.20</u>	<u>81.69 ± 0.55</u>	93.80 ± 0.39	<u>96.26 ± 0.26</u>
SwAV	53.09 ± 0.89	96.82 ± 0.23	67.83 ± 0.76	79.22 ± 0.50	85.24 ± 0.57	87.33 ± 0.43	97.10 ± 0.23	79.07 ± 0.59	93.84 ± 0.39	96.12 ± 0.27
Supervised	**68.90 ± 0.87**	**98.51 ± 0.16**	**82.72 ± 0.65**	**84.29 ± 0.44**	**88.89 ± 0.49**	86.58 ± 0.49	96.95 ± 0.25	**82.93 ± 0.55**	**98.25 ± 0.19**	**96.28 ± 0.27**

3. Preliminaries

Representation learning methods   We consider the following thirteen self-supervised learning methods. **Contrastive**: InsDis (also known as NPID) [58], MoCo-v1 [21] and its upgrade MoCo-v2 [8], PIRL [39], SimCLR-v1 [6] and SimCLR-v2 [7], InfoMin [52] and BYOL [17]. **Clustering**: PCL-v1 and PCL-v2 [34], SeLa-v2 [1, 5], DeepCluster-v2 [4, 5] and SwAV [5].

For these methods, we download pre-trained weights of ResNet50(1×) [23] models and use the backbone as a feature extractor when transferring to downstream tasks. Additionally, we evaluate a supervised baseline for comparison, a standard pre-trained ResNet50 available from the PyTorch [44] library. All models have 23.5M parameters in their backbones and were pre-trained on the ImageNet [11] training set, consisting of 1.28M images, and only the supervised baseline used labels. More details of the pre-trained models can be found in Section A.1 of the appendix.

As we cannot control the pre-training setup, there are differences in how long the models were trained for, what data augmentation they applied, what loss they trained with and what additional architectural elements they used. These differences are detailed in Table 10 in the appendix. However, all models use the same ResNet50(1×) [23] backbone,

Table 3. Few-shot transfer (5-way 20-shot) of pre-trained models using prototypical networks on CD-FSL. Results style: **best**, <u>second best</u>.

	CropDiseases	EuroSAT	ISIC	ChestX
InsDis	91.95 ± 0.44	86.52 ± 0.51	52.19 ± 0.53	29.13 ± 0.44
MoCo-v1	92.04 ± 0.43	86.55 ± 0.51	**53.79 ± 0.54**	30.00 ± 0.43
PCL-v1	80.74 ± 0.57	75.19 ± 0.67	38.01 ± 0.44	25.54 ± 0.43
PIRL	91.19 ± 0.49	87.06 ± 0.49	53.24 ± 0.56	29.48 ± 0.45
PCL-v2	92.58 ± 0.44	87.94 ± 0.40	44.40 ± 0.52	28.28 ± 0.42
SimCLR-v1	94.03 ± 0.37	89.38 ± 0.40	53.00 ± 0.54	30.82 ± 0.43
MoCo-v2	92.12 ± 0.46	88.92 ± 0.41	52.39 ± 0.49	29.43 ± 0.45
SimCLR-v2	94.92 ± 0.34	91.05 ± 0.36	53.15 ± 0.53	30.90 ± 0.44
SeLa-v2	94.75 ± 0.37	88.34 ± 0.57	48.43 ± 0.54	30.43 ± 0.46
InfoMin	92.34 ± 0.44	86.76 ± 0.47	48.21 ± 0.54	29.48 ± 0.44
BYOL	96.07 ± 0.33	89.62 ± 0.39	<u>53.76 ± 0.55</u>	30.71 ± 0.47
DeepCluster-v2	**96.63 ± 0.29**	**92.02 ± 0.37**	49.91 ± 0.53	**31.51 ± 0.45**
SwAV	<u>96.15 ± 0.31</u>	<u>91.99 ± 0.36</u>	47.08 ± 0.50	<u>30.91 ± 0.45</u>
Supervised	93.09 ± 0.43	88.36 ± 0.43	48.79 ± 0.53	29.26 ± 0.44

meaning we can evaluate them in the same way. For a given target dataset we pass the training data through the backbone to obtain feature vectors. On top of the backbone we attach a task-specific head to produce label predictions for the target task. When fitting to the target training set we either optimise only the head or finetune the entire network.

4. Experiments

We now thoroughly evaluate our large suite of recent SSL methods on transfer to a variety of downstream domains and tasks. Our evaluation consists of four sets of transfer experiments: (1) many-shot recognition, where a substantial amount of labelled training data is available in the target domain for fitting a classifier, (2) few-shot recognition where only a few labelled training images are available for each class in the target domain, and two cases of cross-task transfer, (3) object detection and (4) dense prediction, using the two exemplar tasks: surface normal estimation and semantic segmentation. The first two experiments contain some benchmarks with significant amounts of domain-shift compared to the ImageNet source data, while the last two experiments contain task-shift, that may make different demands on the features. For example, detection may require stronger spatial sensitivity of features compared to recognition; and dense prediction may require something closer to spatial equivariance, in contrast to recognition which may benefit from spatial invariance.

4.1. Many-shot recognition

Experimental setup   For many-shot recognition, we adopt the benchmark suite proposed in the transfer learning study [28], which includes the target datasets FGVC Aircraft [38], Caltech-101 [15], Stanford Cars [29], CIFAR-10 [30], CIFAR-100 [30], DTD [9], Oxford 102 Flowers [41], Food-101 [3], Oxford-IIIT Pets [43], SUN397 [59] and Pascal VOC2007 [14]. These datasets cover a wide range of classification tasks, including texture, scene and fine/coarse-grained object classification. While they are all in the 'many-shot' regime, they include significant variety in amount of training data (2,000-75,000 images), and cardinality of classification (10-397 classes). We exclude the

Table 4. Detection transfer from pre-trained models using Faster R-CNN FPN on PASCAL VOC. We train models both with frozen backbones and with all layers finetuned. We report the metrics AP, AP50 and AP75. Results style: **best**, <u>second best</u>.

	VOC (Frozen)			VOC (Finetune)		
	AP	AP50	AP75	AP	AP50	AP75
InsDis	50.13	77.92	53.34	48.82	76.43	52.40
MoCo-v1	50.39	78.03	54.08	50.51	78.06	54.55
PCL-v1	51.05	80.16	54.36	<u>53.93</u>	81.69	59.33
PIRL	49.54	77.26	52.79	45.08	72.50	47.80
PCL-v2	52.45	81.22	57.13	53.92	<u>81.89</u>	<u>59.35</u>
SimCLR-v1	51.94	81.19	56.49	52.19	81.36	56.92
MoCo-v2	<u>54.22</u>	81.86	<u>59.97</u>	44.74	72.82	47.01
SimCLR-v2	**54.95**	**82.34**	**61.18**	51.42	79.40	55.89
SeLa-v2	49.66	80.63	53.15	50.41	80.55	54.35
InfoMin	53.45	81.12	58.96	44.92	72.72	47.41
BYOL	53.32	<u>82.01</u>	58.37	**54.91**	**82.57**	**60.82**
DeepCluster-v2	50.05	80.87	53.21	51.03	80.93	55.51
SwAV	50.68	80.82	54.11	52.07	81.50	56.03
Supervised	51.99	81.53	56.21	53.26	81.51	59.07

Birdsnap [2] dataset as a significant number of the original images are no longer available at the given URLs. When using these datasets throughout the paper, we will refer to them collectively as the *Kornblith* datasets.

We report results for both linear evaluation and finetuning. For linear, we fit multinomial logistic regression on the extracted features. When finetuning, we train the models for 5,000 steps using SGD with Nesterov momentum. Full details about our fitting, the dataset splits, metrics and preprocessing can be found in Appendix A.2.

Results   The results can be found in Table 1[1].

Linear: We draw the following observations: (i) On all but one downstream task, the best self-supervised methods outperform supervised pre-training on ImageNet (bottom row). This is notably the case on Aircraft and Cars benchmarks, where the best self-supervised models outperform supervised pre-training by over 10% absolute performance. Although supervised pre-training is best for within-dataset transfer to ImageNet (leftmost column), this shows that the self-supervised methods are learning a more general purpose feature for diverse downstream tasks. (ii) The recent methods, DeepCluster-v2 [5], BYOL [17] and SwAV [5] stand out as being regularly highly ranked in each case.

Finetuning: The bottom half of Table 1 shows a similar picture. The supervised model is more competitive here, achieving top results on three datasets including Aircraft where its frozen weights under-performed. However, DeepCluster-v2, SwAV and SimCLR-v2 still outperform it overall, confirming that, on the whole, the best self-supervised learners have surpassed supervision for many-shot recognition transfer. We present further discussion about these results in Section 4.6.

---

[1]Note that the linear evaluation in [28] uses weights from different checkpoints during pre-training, while we only use the final released weights. This explains why our numbers differ on some datasets.

4.2. Few-shot recognition

Experimental setup   To evaluate the performance of self-supervised features on downstream tasks in the few-shot regime, we use the same Kornblith datasets as for the many-shot regime, save for the multi-label VOC2007. Additionally, we evaluate on the Broader Study of Cross-Domain Few-Shot Learning (CD-FSL) benchmark introduced by [19]. It consists of four datasets that exhibit increasing dissimilarity to natural images, CropDiseases [40], EuroSAT [24], ISIC2018 [54, 10] and ChestX [56].

Our evaluation uses a nearest-centroid classifier (also known as Prototypical Networks [50]) on the features extracted from the ResNet50 backbones. Across the 14 datasets, we consider 5-way 20-shot transfer (with 5-way 5-shot and 5-way 50-shot reported in the appendix). The test set (query set) always has 15 images per class and we perform 600 randomly sampled few-shot episodes and report the average accuracy along with a 95% confidence interval.

Results   Table 2 shows the results on the Kornblith datasets. We see that: (i) The supervised model dominates in this setting, on all datasets but DTD and Flowers. (ii) It does so by a large margin on Aircraft and Cars (5+%), in stark contrast to our linear many-shot results above. (iii) The best self-supervised models are BYOL and DeepCluster-v2, followed by SwAV and SimCLR-v2.

The CD-FSL results are shown in Table 3, from which we make the following observations: (i) Across all datasets and evaluation setups several self-supervised models outperform the supervised baseline. (ii) On CropDiseases, the dataset most similar to ImageNet, the standout models are similar to those in the many-shot experiment: DeepCluster-v2, SwAV and BYOL. On EuroSAT, SimCLR-v2 overtakes BYOL in third place after the same top two. (iii) PCL-v1 consistently transfers the worst in the few-shot setting. (iv) On ISIC, the least 'object-like' of all the datasets, the ranking of the methods is very different. We present further discussion about these results in Section 4.6.

Summarising these results, we see that self-supervision still lags behind for low domain shift few-shot transfer while it consistently beats supervision for larger domain shifts.

4.3. Detection

Experimental setup   We evaluate the pre-trained networks on Pascal VOC using Faster R-CNN [46] with a Feature Pyramid Network [35] backbone. We use the detectron2 [57] framework and base our evaluation on the suggested hyperparameters therein. Training is done on both the trainval07 and the trainval12 datasets and evaluation is done on the test2007 set. We report AP50, the default VOC metric as well as the COCO-style metrics AP and AP75. We evaluate both freezing the backbone (all but the last residual block) and finetuning all layers end-to-end. Full training details can be found in Section A.4 in the appendix.

Results   The results are presented in Table 4, from which we observe that: (i) The best self-supervised models again

Table 5. Surface normal estimation on NYUv2 (left), with mean and median angular error (lower is better) and percentage of pixels within $11.25°$, $22.5°$, and $30°$ degrees of ground truth surface normal (higher is better). Semantic segmentation on ADE20K (right), with the metrics mean intersection over union and pixel accuracy. Results style: **best**, <u>second best</u>.

| | Surface Normal Estimation | | | | | Semantic Segmentation | |
	Mean	Median	$11.25°$	$22.5°$	$30°$	Mean IoU	Accuracy
InsDis	32.99	27.35	23.58	43.02	53.51	0.2742	68.03
MoCo-v1	33.69	28.63	21.51	41.07	51.87	0.2530	62.48
PCL-v1	37.90	33.58	16.73	34.96	45.43	**0.2983**	**75.00**
PIRL	33.16	27.66	22.24	42.41	53.12	0.2697	66.09
PCL-v2	33.98	28.67	21.95	41.21	51.76	0.2965	74.81
SimCLR-v1	30.47	23.26	28.34	48.88	<u>59.01</u>	<u>0.2966</u>	74.83
MoCo-v2	30.49	24.19	26.59	47.43	58.03	0.2794	67.69
SimCLR-v2	**28.77**	**21.30**	**30.58**	**51.87**	**62.05**	0.2960	<u>74.90</u>
SeLa-v2	39.57	36.10	14.56	32.49	42.51	0.2956	74.71
InfoMin	32.45	26.58	23.86	44.00	54.66	0.2944	74.78
BYOL	30.56	<u>23.12</u>	<u>29.23</u>	<u>49.10</u>	<u>59.01</u>	0.2940	74.74
DeepClust.	<u>30.19</u>	23.54	28.44	48.42	58.76	0.2744	67.08
SwAV	31.64	24.86	27.80	46.70	56.67	0.2961	74.87
Supervised	33.52	27.91	24.00	42.33	52.80	0.2563	61.83

outperform supervised pre-training as a transfer learning source. (ii) However, the best performing models are now quite different from those in the previous sections (more on this in Section 4.6) with SimCLR-v2 excelling for a frozen backbone, and BYOL excelling for a finetuned backbone.

Our results are in contrast to the headline claim in [22], which is that ImageNet pre-training is not necessarily useful in transfer to detection tasks. However, this observation in [22] was based on the COCO benchmark, and did not hold for their experiments on Pascal VOC. This is most likely due to the lesser number of images and categories in VOC.

4.4. Surface normal estimation

Experimental setup We evaluate the pre-trained features for surface normal estimation on NYUv2 [49] (ground-truth from [32]) as the first exemplar task for dense prediction problems. We train PSPNet models [68] with ResNet50 backbones, as in previous experiments. The performance is measured by the mean and median angular error, as well as the percentage of estimated surface normals within $11.25°$, $22.5°$, and $30°$ of the ground truth.

Results From the results in Table 5[2], we can see that the best self-supervised models again outperform supervised pre-training for transfer from ImageNet, with SimCLR-v2 winning across the board followed by BYOL. In this case the margins are often substantial with SimCLR-v2 outperforming supervised pre-training by around 4-10% depending on the metric.

4.5. Semantic segmentation

Experimental setup The second dense prediction task we consider is semantic segmentation on ADE20K [70].

---

[2]Note that our numbers are *not* directly comparable to [16] as they based model (checkpoint) selection on test performance. Given the absence of a validation split for NYUv2, we considered it better practice to train all methods for a fixed number of iterations. As the focus of our benchmark is on comparison across models, this should not be an issue.

We use the CSAIL Semantic Segmentation framework implementation of UPerNet [60], which is based on the Feature Pyramid Network [35] and the Pyramid Pooling Module [68]. We report both the mean intersection over union (IoU) and accuracy.

Results We present the results of these experiments in the two rightmost columns of Table 5. The main insights to be gleaned from these performance measurements are: (i) the supervised baseline is among the worst performing methods; (ii) PCL-v1 achieves the top results, while it consistently performed poorly in recognition; and (iii) there is only a very slight correlation between the performance of SSL methods on ImageNet recognition and their performance on semantic segmentation.

4.6. Does better ImageNet performance lead to better performance on downstream tasks?

As we mentioned in the introduction, a major question we set out to answer is whether ImageNet performance is in general representative of downstream performance on diverse tasks and datasets? This determines whether practitioners can safely select the latest benchmark leading SSL methods for downstream tasks; and influences whether state-of-the-art self-supervised representations are likely to be useful off-the-shelf for practical problems in diverse domains [45, 19], or whether practitioners would need to collect domain-specific data for large scale training. It is also indicative of whether pursuing ImageNet recognition performance is the right benchmark for the self-supervision research community, or whether we need a richer set of benchmarks to properly assess the value of self-supervision research progress to the broader vision community.

Analysis Based on our experiments in Sections 4.1-4.5, we compute the Pearson and Spearman (rank) correlation coefficients between ImageNet and downstream task performance across all dataset pairs. Detailed performance plots for every dataset are shown in Figs 5-6 in the appendix. From the summary of correlations in Figs 1-2 we can see that: (i) The ImageNet-to-downstream task correlation is generally high for many-shot recognition tasks. (ii) In the case of few-shot recognition, the correlations are fairly strong for low domain shift transfer. For the larger domain shifts in CD-FSL the correlation is weaker, but present for three of the four datasets. It is entirely absent for the ISIC skin lesion benchmark, which is arguably the least ImageNet-like out of the four due to unstructured texture. (Chest Xray dataset is different due to being greyscale, but similar in the presence of structure in the images). (iii) For detection, AP50 is the strongest correlated metric, and frozen fitting correlates stronger than finetuning. (iv) For surface-normal estimation, weak but clear correlation is present across all metrics. (v) For semantic segmentation the correlation is weak and even non-existent for ranks.

Overall we can distill the following take-home messages for practitioners. (1) *For recognition tasks on structured images, one is safe to choose the current benchmark-leading*
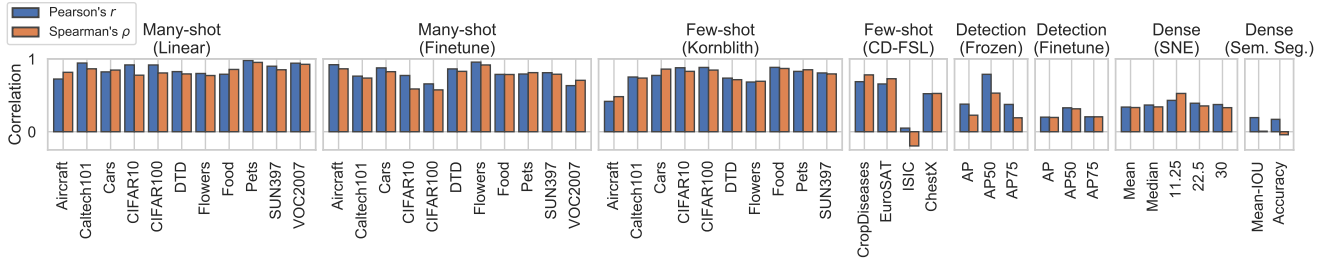
Figure 2. The correlations between ImageNet and downstream transfer performance, showing high correlation for many-shot recognition, but increasingly less so for few-shot, object detection and dense prediction. The blue bars show Pearson's r correlations between logit-transformed ImageNet top-1 accuracy and the transfer performance (which is logit-transformed for metrics bounded between 0 and 1, and negated for minimisation metrics). The orange bars show the rank correlation (Spearman's ρ).

self-supervised representations for direct transfer purposes in either the many-shot or few-shot regime, and this feature may well out-perform supervised transfer from ImageNet with the exception of few-shot on ImageNet-like data. (2) *For spatially sensitive prediction tasks such as detection and dense prediction, the current SimCLR-v2 and BYOL are good bets and may outperform supervised transfer, but taking the future ImageNet benchmark leader may not necessarily lead to best performance.* (3) *For recognition tasks on unstructured images and textures, there is no clear recipe to choose a self-supervised representation and task-specific comparison is required.*

4.7. Does pre-training strategy influence downstream model calibration?

As computer vision is deployed in many high-importance real-world applications that are safety critical [31], or have potential impact on social fairness [13], the *calibration* [18] of predictive models is as important as overall accuracy, if not a hard-requirement for system deployment. Mistaken predictions should be flagged as such by low-confidence probabilities, so they can be dealt with by another process. Given the growing social importance of this issue, we also evaluate whether pre-training strategy has an influence on downstream model calibration.

We compute the expected calibration error (ECE) [18] with 15 bins of the models from our two many-shot benchmarks, linear and finetuning. We exclude VOC2007 as it is a multi-label problem. As a simple post-hoc calibration method, we also perform temperature scaling [18] on the predictions. Figure 3 shows the average ECE for each model over its ImageNet performance both with and without further calibration via temperature scaling.

Analysis The overall trend shows better self-supervised methods (as measured on ImageNet accuracy) achieving better calibration. In the unscaled linear case, several SSL models get significantly lower ECE compared to supervision, which also partially holds true after temperature scaling. For unscaled finetuning, the supervised model is the best, though after scaling it is surpassed by DeepCluster-v2 and SwAV. Overall there is a strong inverse correlation of

ECE to ImageNet performance – though less so after temperature scaling – showing better self-supervised models are better calibrated in downstream transfer.

4.8. What information is retained in features?

How to measure what information is retained in CNN features is an open research question in itself [62]. However, to complement our prior performance-driven comparisons, we conduct a preliminary analysis on this topic using the methodology suggested in [69]. Specifically, we compare the ability to reconstruct RGB images from the features extracted by our pre-trained models, when using the deep image prior [55]. This feature inversion algorithm trains an encoder-decoder architecture to produce an image which achieves similar features to the original image when passed through the pre-trained model. We perform image reconstruction from features across all 14 pre-trained models and all 15 unique recognition datasets.

Analysis To quantify the results we compare: (i) the perceptual difference between original images and reconstructions as measured by [66], and (ii) pixel-wise mean
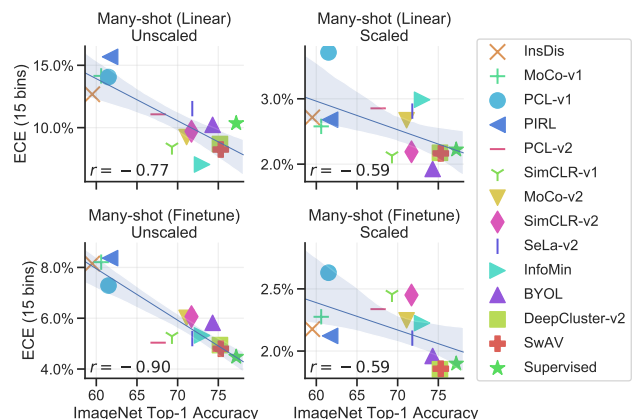

Figure 3. In the linear evaluation setting, many recent self-supervised methods are better calibrated than the supervised baseline. However, after finetuning the supervised model has the best calibration. Overall there is a clear trend that newer SSL models have better calibration (ECE metric, lower is better).
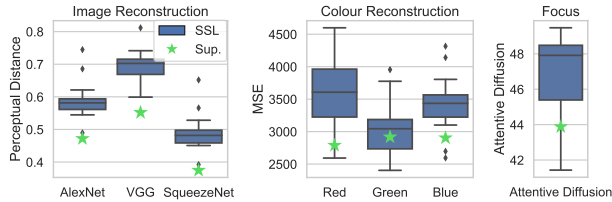
Figure 4. Left: When using features from the supervised baseline (star), the reconstructions are perceptually more similar to the original images compared to the self-supervised models (boxplot). Middle: The supervised model better reconstructs colour information, especially red and blue channels. SSL models likely underperform here because of heavy data augmentation during training. See Fig 8 in appendix for reconstructed images. Right: The supervised model has smaller attentive focus compared to SSL models. See Fig 9 in appendix for attention maps.

squared error between original images and reconstructions. We summarise the results in Figure 4, with complete qualitative examples given in Figure 8 of the appendix. From the qualitative results we can see that all methods can provide a somewhat recognisable reconstruction, with the noticeable difference that supervised pre-training tends to provide much cleaner colour in the reconstruction. We conjecture that the poor colour fidelity is due to the heavy colour distortions used in the data augmentation of state of the art self-supervised methods leading them to learn colour-invariant features. If so this means that downstream users should be cautious about applying such features to tasks where colour is a critical feature for decision-making. There is a general trend towards stronger methods (in the ImageNet accuracy sense) providing better reconstructions (correlation of -0.69 for perceptual distance computed by the VGG network and for the colour errors, red -0.56, green -0.11, blue -0.22).

4.9. Does pre-training strategy influence where downstream networks attend?

We adapt traditional occlusion-based saliency methods [62] to a task-agnostic setting. By occluding part of the image we compute the distance between the features of the clean and occluded images. As we pass the occlusion mask over the image we compute the average feature distance for each pixel. The larger the value for a given pixel, the more the feature changes if that pixel is occluded in the input, indicating the network is highly sensitive to this region. More details can be found in Section A.9 of the appendix.

Analysis We summarise the results quantitatively in Figure 4, with complete qualitative examples given in Figure 9 of the appendix. From the qualitative results, some notable observations are that on the aircraft image, the supervised baseline attends to mainly the sky, while the self-supervised ones focus on the actual aircraft. This explains why the supervised model performed so poorly at this fine-grained classification task earlier, as it fails to focus on the details of the aircraft. Overall, there is a trend that the supervised

model attends to smaller regions than the self-supervised models. This is summarised quantitatively in Figure 4, which reports attentive diffusion/focus in terms of the percentage of the attention map with values above its mean. The correlation with ImageNet performance here is very low at 0.09, but the correlation with average transfer performance (many-shot linear) is significantly higher at 0.38, suggesting that a larger attentive region helps in transfer to recognition tasks. Overall we consider these results to be reflective of widely reported [61] attentive overfitting of supervised learning models, which self-supervised learners seem less vulnerable to, and which may contribute to their superior performance in most recognition tasks and superior calibration for un-tuned backbones.

5. Discussion

We have conducted the first thorough and up-to-date empirical evaluation of state of the art SSL performance when applied to diverse downstream tasks, a comparison that has been missing in the literature until now. Our evaluation showed that: (1) The best self-supervised methods today can usually outperform supervised pre-training as a source of knowledge transfer, an exciting milestone for the field that has long been speculated on, but now clearly confirmed. (2) Performance of self-supervised representations on ImageNet is reassuringly broadly representative of downstream performance on natural image recognition tasks, confirming the relevance of this metric for research. (3) However, ImageNet performance is not reliably representative of downstream performance on unstructured image recognition, or other spatially sensitive tasks such as detection, surface normal prediction and semantic segmentation. Thus the vision of a 'universal' pre-trained feature with best performance on diverse downstream tasks is yet to be realised. Furthermore, SSL researchers should adopt a wider range of benchmarks to better impact the broader computer vision community.

There are several limitations of our current study. Most notably, we were not able to compare the value of self-supervised representations transferred from ImageNet to domain-specific self-supervised representations trained on each target dataset. This would answer the important question of whether domain-specific SSL is worthwhile, and if ImageNet can provide truly generic features. This is an important but complex question to answer given the different training protocols of existing methods and diversity of downstream datasets, so we leave this to future work.

References

[1] Yuki Markus Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 3, 4

[2] Thomas Berg, Jiongxin Liu, Seung Woo Lee, Michelle L. Alexander, David W. Jacobs, and Peter N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014. 5

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 - Mining discriminative components with random forests. In *ECCV*, 2014. 4

[4] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep Clustering for Unsupervised Learning of Visual Features. In *ECCV*, 2018. 3, 4

[5] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*, 2020. 1, 2, 3, 4, 5

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 2020. 2, 3, 4, 11

[7] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big Self-Supervised Models are Strong Semi-Supervised Learners. In *NeurIPS*, 2020. 2, 4

[8] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved Baselines with Momentum Contrastive Learning. *arXiv*, 2020. 2, 4

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing Textures in the Wild. In *CVPR*, 2014. 4

[10] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M. Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, Harald Kittler, and Allan Halpern. Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC). *arXiv*, 2019. 5

[11] J Deng, W Dong, R Socher, L.-J. Li, K Li, and L Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2, 4

[12] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks. In *NeurIPS*, 2014. 2

[13] Mengnan Du, Fan Yang, Na Zou, and Xia Hu. Fairness in Deep Learning: A Computational Perspective. *IEEE Intelligent Systems*, 2020. 7

[14] Mark Everingham, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 2010. 4, 11

[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshops*, 2004. 4

[16] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and Benchmarking Self-Supervised Visual Representation Learning. In *ICCV*, 2019. 2, 3, 6, 11

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. In *NeurIPS*, 2020. 1, 4, 5

[18] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On Calibration of Modern Neural Networks. In *ICML*, 2017. 7

[19] Yunhui Guo, Noel C. Codella, Leonid Karlinsky, James V. Codella, John R. Smith, Kate Saenko, Tajana Rosing, and Rogerio Feris. A Broader Study of Cross-Domain Few-Shot Learning. In *ECCV*, 2020. 5, 6, 11

[20] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. *Journal of Machine Learning Research*, 2010. 2

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*, 2019. 2, 3, 4

[22] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking ImageNet Pre-training. In *ICCV*, 2018. 6

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4

[24] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 2, 5

[25] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, and Aaron van den Oord. Data-Efficient Image Recognition with Contrastive Predictive Coding. In *ICML*, 2020. 2

[26] Longlong Jing and Yingli Tian. Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[27] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting Self-Supervised Visual Representation Learning. In *CVPR*, 2019. 2, 3

[28] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do Better ImageNet Models Transfer Better? In *CVPR*, 2019. 3, 4, 5

[29] J. Krause, Jun Deng, M. Stark, and Li Fei-Fei. Collecting a Large-scale Dataset of Fine-grained Cars. In *Second Workshop on Fine-Grained Visual Categorization*, 2013. 4

[30] Alex Krizhevsky and Geoffrey Hinton. Learning Multiple Layers of Features from Tiny Images. *arXiv*, 2009. 4

[31] Lindsey Kuper, Guy Katz, Justin Gottschlich, Kyle Julian, Clark Barrett, and Mykel Kochenderfer. Toward Scalable Verification for Safety-Critical Deep Networks. In *SysML*, 2018. 7

[32] L Ladickỳ, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, 2014. 6

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

[34] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. Prototypical Contrastive Learning of Unsupervised Representations. In *ICLR*, 2021. 2, 4

[35] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature Pyramid Networks for Object Detection. In *CVPR*, 2017. 5, 6, 11

[36] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1989. 11

[37] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. Self-supervised Learning: Generative or Contrastive. *arXiv*, 2020. 2

[38] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-Grained Visual Classification of Aircraft. *arXiv*, 2013. 4

[39] Ishan Misra and Laurens van der Maaten. Self-Supervised Learning of Pretext-Invariant Representations. In *CVPR*, 2020. 2, 4

[40] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 2016. 2, 5

[41] Maria Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, 2008. 4

[42] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[43] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 4

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, 2019. 4

[45] Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding Transfer Learning for Medical Imaging. In *NeurIPS*, 2019. 6

[46] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*, 2015. 5, 11

[47] Jürgen Schmidhuber. Making the World Differentiable: On Using Self-Supervised Fully Recurrent Neural Networks for Dynamic Reinforcement Learning and Planning in Non-Stationary Environments. 1990. 1

[48] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *Communications of the ACM*, 2020. 2

[49] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *ECCV*, 2012. 6

[50] Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical Networks for Few-shot Learning. In *NeurIPS*, 2017. 5

[51] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive Multiview Coding. In *ECCV*, 2020. 2

[52] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What Makes for Good Views for Contrastive Learning? In *NeurIPS*, 2020. 4

[53] Yuandong Tian, Lantao Yu, Xinlei Chen, and Surya Ganguli. Understanding Self-supervised Learning with Dual Deep Networks. *arXiv*, 2020. 3

[54] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. Data descriptor: The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 2018. 2, 5

[55] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep Image Prior. In *CVPR*, 2018. 7, 12

[56] Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. ChestX-ray8: Hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, 2017. 5

[57] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5, 11

[58] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *CVPR*, 2018. 2, 3, 4

[59] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 4

[60] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 6

[61] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017. 8

[62] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 7, 8

[63] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, Lucas Beyer, Olivier Bachem, Michael Tschannen, Marcin Michalski, Olivier Bousquet, Sylvain Gelly, and Neil Houlsby. A Large-scale Study of Representation Learning with the Visual Task Adaptation Benchmark. *arXiv*, 2019. 3

[64] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew Soon Ong, and Chen Change Loy. Online Deep Clustering for Unsupervised Representation Learning. In *CVPR*, 2020. 3

[65] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful Image Colorization. In *ECCV*, 2016. 2

[66] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *CVPR*, 2018. 7, 13

[67] Youshan Zhang and Brian D. Davison. Impact of ImageNet Model Selection on Domain Adaptation. In *WACVW*, 2020. 2

[68] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 6

[69] Nanxuan Zhao, Zhirong Wu, Rynson W. H. Lau, and Stephen Lin. What Makes Instance Discrimination Good for Transfer Learning? In *ICLR*, 2021. 7, 12

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2019. 6, 11

A. Appendix

A.1. Pre-trained models

Note that for InsDis we use the model weights provided by the PyContrast GitHub repository which report higher ImageNet top-1 accuracy than originally reported (59.5 vs 54.0). As weights are not available for PIRL we likewise, take the ones provided by PyContrast which reports a slightly lower ImageNet accuracy of 61.7 (compared to 63.6). All other models are obtained from the original authors. We use the PyTorch framework in our code and therefore convert some of the models from their TensorFlow checkpoints. For most models we normalise the inputs by the mean and standard deviation on the ILSVRC12 train set, apart from SimCLR-v1/v2 which do not expect normalised inputs.

A.2. Many-shot evaluation details

The top-1 accuracy metric is reported on Food-101, CIFAR-10, CIFAR-100, SUN397, Stanford Cars, and DTD, mean per-class accuracy on FGVC Aircraft, Oxford-IIIT Pets, Caltech-101, and Oxford 102 Flowers and the 11-point mAP metric from [14] on Pascal VOC 2007. On Caltech-101 we randomly select 30 images per class to form the training set and we test on the rest. We use the first train/test split defined in DTD and SUN397. On FGVC Aircraft, Pascal VOC2007, DTD, and Oxford 102 Flowers we use the validation sets defined by the authors, and on the other datasets we randomly select 20% of the training set to form the validation set. The optimal hyperparameters were selected on the validation set, after which we retrained the model on all training and validation images. Finally, the accuracy is computed on the test set.

Linear We fit a multinomial logistic regression model on the extracted features of dimensionality 2048 from the frozen backbones. No augmentation was used and the images were resized to 224 pixels along the shorter side using bicubic resampling, followed by a center crop of 224×224. We select the $\ell 2$ regularisation constant on the validation set over 45 logarithmically spaced values between 10^{-6} and 10^5. The model is optimised using L-BFGS [36] on the softmax cross-entropy objective. As Pascal VOC2007 is a multi-label task, we fit one binary classifier for each class.

Finetuning We finetune the models following the protocol of [6] with minor modifications. We train for 5000 steps with a batch size of 64. The optimiser is SGD with Nesterov momentum and a momentum parameter of 0.9. The learning rate follows a cosine annealing schedule without restarts, and the initial learning rate is chosen from a grid of 4 logarithmically spaced values between 0.0001 and 0.1. The weight decay is similarly chosen from a grid of 4 logarithmically spaced values between 10^{-6} and 10^{-3}, along with no weight decay. These weight decay values are divided by the learning rate. We select the data augmentation from: random crop with resize and flip, or simply a center crop.

A.3. Few-shot evaluation details

For each few-shot learning episode we sample images from the combined sets of train, validation and test images. We fit a nearest centroid classifier on the extracted features of dimensionality 2048 from the frozen backbones. No augmentation was used and the images were resized to 224 pixels along the shorter side using bicubic resampling, followed by a center crop of 224×224. The fitted model is evaluated using 15 query images in each episode and the reported accuracies and errors are computed from 600 total episodes. In addition to the 20-shot results presented in the paper, we also report 5-shot and 50-shot results in Tables 6, 7 and 8. Note that in the original CD-FSL benchmark [19], models are only allowed to pre-train on mini-ImageNet and not the full version, so our results are not comparable to those of the original authors.

A.4. Detection evaluation details

We train the detectors on the VOC 2007 and 2012 train-val sets, and test on VOC 2007 test. When evaluating frozen backbones, we freeze all but the final residual block of the ResNets. In the full finetuning setup, we let the entire network be trainable. We extract features from the backbone using a Feature Pyramid Network [35] architecture and attach a Faster R-CNN [46] detector head to produce predictions. During training, the images are resized so the shorter side is one of [480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800] and during testing to 800 pixels. The models are trained for 144k iterations with a 100 iteration warm-up to an initial learning rate of 0.0025 which is decayed by a factor of 10 at iterations 96k and 128k. The batch size is 2 and we used a single GPU per model. Any other details of training uses the default values of the detectron2 [57] framework.

A.5. Surface normal estimation evaluation details

We use the implementation of [16], which is based on [70]. Each model is trained for 150 epochs, with the full backbone frozen. We use stochastic gradient descent with a momentum of 0.9, batch size of 4 and set the learning rate according to $(1 - \frac{t}{T})^{0.9}$, where t is the current epoch and T is the total number of epochs.

A.6. Semantic segmentation evaluation details

Models are trained (without freezing any layers) using stochastic gradient descent with an initial learning rate of 0.02, which is decayed by a factor of 0.9 every 500 iterations, and a constant momentum rate of 0.9. All models are trained with a batch size of two for 150k iterations in total.

A.7. Computing correlations

At many points in this work we analyse the statistical relationships between different results. This includes the correlation coefficients in Figs. 1, 2, 3, 5 and 6, those reported in the text and more summarised in Table 9. In order to

Table 6. 5-way 5-shot transfer on the Kornblith datasets. We report the average accuracy and 95% confidence interval over 600 test episodes. Results style: **best**, <u>second best</u>.

	Aircraft	Caltech101	Cars	CIFAR10	CIFAR100	DTD	Flowers	Food	Pets	SUN397
InsDis	42.59 ± 0.90	83.31 ± 0.65	46.42 ± 0.72	62.64 ± 0.64	68.06 ± 0.76	73.74 ± 0.67	89.55 ± 0.53	61.50 ± 0.75	73.21 ± 0.68	84.77 ± 0.60
MoCo-v1	42.74 ± 0.94	86.98 ± 0.57	44.63 ± 0.69	60.07 ± 0.64	66.10 ± 0.79	74.98 ± 0.70	89.13 ± 0.53	62.45 ± 0.78	74.68 ± 0.69	85.14 ± 0.57
PCL-v1	39.49 ± 0.87	84.35 ± 0.60	40.59 ± 0.76	62.75 ± 0.63	64.09 ± 0.79	64.48 ± 0.78	77.25 ± 0.75	57.45 ± 0.83	85.51 ± 0.64	80.89 ± 0.62
PIRL	42.91 ± 0.93	85.04 ± 0.62	46.87 ± 0.74	64.39 ± 0.63	69.32 ± 0.76	72.80 ± 0.69	89.52 ± 0.51	61.32 ± 0.77	74.05 ± 0.69	85.03 ± 0.59
PCL-v2	34.36 ± 0.75	86.33 ± 0.54	42.57 ± 0.70	70.96 ± 0.59	74.10 ± 0.68	72.84 ± 0.74	87.52 ± 0.52	61.00 ± 0.78	85.16 ± 0.66	84.80 ± 0.57
SimCLR-v1	48.11 ± 0.98	94.10 ± 0.36	53.46 ± 0.80	70.65 ± 0.66	77.10 ± 0.70	76.71 ± 0.65	93.10 ± 0.38	65.13 ± 0.77	86.52 ± 0.58	89.71 ± 0.47
MoCo-v2	35.97 ± 0.80	90.14 ± 0.48	49.55 ± 0.80	69.47 ± 0.62	75.62 ± 0.70	78.08 ± 0.67	91.12 ± 0.46	66.34 ± 0.80	87.91 ± 0.59	89.18 ± 0.48
SimCLR-v2	47.12 ± 0.96	94.92 ± 0.34	52.64 ± 0.77	71.90 ± 0.61	79.71 ± 0.66	79.06 ± 0.63	93.83 ± 0.37	69.85 ± 0.74	86.29 ± 0.58	90.99 ± 0.45
SeLa-v2	36.35 ± 0.77	89.85 ± 0.53	47.99 ± 0.78	71.27 ± 0.59	76.29 ± 0.72	77.81 ± 0.62	90.11 ± 0.51	67.69 ± 0.77	81.36 ± 0.67	90.80 ± 0.46
InfoMin	35.06 ± 0.75	87.03 ± 0.53	49.67 ± 0.79	67.28 ± 0.62	71.72 ± 0.72	73.43 ± 0.75	87.53 ± 0.57	65.95 ± 0.77	86.98 ± 0.57	86.54 ± 0.55
BYOL	<u>53.88 ± 0.99</u>	96.84 ± 0.28	<u>58.77 ± 0.81</u>	70.59 ± 0.62	79.19 ± 0.68	**81.33 ± 0.59**	**96.06 ± 0.30**	71.39 ± 0.72	<u>92.20 ± 0.46</u>	91.63 ± 0.43
DeepCluster-v2	47.73 ± 0.97	94.75 ± 0.35	58.17 ± 0.82	<u>74.47 ± 0.61</u>	<u>80.52 ± 0.65</u>	78.79 ± 0.59	95.44 ± 0.32	<u>72.71 ± 0.72</u>	89.13 ± 0.56	92.95 ± 0.41
SwAV	46.22 ± 0.91	94.43 ± 0.37	56.08 ± 0.82	72.73 ± 0.62	79.32 ± 0.67	79.80 ± 0.57	94.55 ± 0.37	69.65 ± 0.73	88.76 ± 0.56	<u>93.00 ± 0.42</u>
Supervised	**58.35 ± 0.96**	**97.61 ± 0.24**	**73.68 ± 0.84**	**77.50 ± 0.55**	**83.74 ± 0.61**	<u>80.83 ± 0.59</u>	94.19 ± 0.41	**76.23 ± 0.71**	**97.45 ± 0.28**	**93.78 ± 0.38**

Table 7. 5-way 50-shot transfer on the Kornblith datasets, apart from Caltech101, Cars and Flowers which do not have enough images per class for this setup. We report the average accuracy and 95% confidence interval over 600 test episodes. Results style: **best**, <u>second best</u>.

	Aircraft	CIFAR10	CIFAR100	DTD	Food	Pets	SUN397
InsDis	51.06 ± 0.88	71.77 ± 0.52	77.57 ± 0.63	83.97 ± 0.47	73.43 ± 0.63	84.78 ± 0.56	92.10 ± 0.39
MoCo-v1	51.20 ± 0.89	68.22 ± 0.54	75.22 ± 0.70	84.76 ± 0.49	74.19 ± 0.60	85.65 ± 0.55	92.31 ± 0.38
PCL-v1	44.78 ± 0.82	69.35 ± 0.53	72.07 ± 0.70	77.18 ± 0.58	67.46 ± 0.67	90.76 ± 0.46	87.59 ± 0.47
PIRL	52.17 ± 0.88	72.23 ± 0.52	78.43 ± 0.64	83.94 ± 0.51	73.05 ± 0.62	85.58 ± 0.53	92.44 ± 0.39
PCL-v2	38.48 ± 0.78	79.51 ± 0.45	82.86 ± 0.53	83.79 ± 0.48	72.30 ± 0.65	89.96 ± 0.48	90.19 ± 0.42
SimCLR-v1	55.29 ± 0.93	79.72 ± 0.49	84.43 ± 0.55	86.24 ± 0.43	77.24 ± 0.59	92.83 ± 0.40	94.34 ± 0.33
MoCo-v2	41.22 ± 0.79	78.01 ± 0.45	83.01 ± 0.57	86.42 ± 0.46	77.17 ± 0.60	92.25 ± 0.42	92.98 ± 0.36
SimCLR-v2	56.33 ± 0.91	81.36 ± 0.48	87.79 ± 0.49	87.99 ± 0.42	81.65 ± 0.53	93.51 ± 0.38	95.51 ± 0.28
SeLa-v2	43.04 ± 0.83	79.16 ± 0.50	84.11 ± 0.59	87.77 ± 0.43	80.10 ± 0.56	89.84 ± 0.44	95.11 ± 0.29
InfoMin	39.91 ± 0.76	74.23 ± 0.53	79.16 ± 0.57	83.09 ± 0.49	76.12 ± 0.59	91.61 ± 0.42	91.05 ± 0.42
BYOL	<u>65.69 ± 0.88</u>	80.49 ± 0.47	87.57 ± 0.50	89.12 ± 0.42	83.04 ± 0.51	<u>96.18 ± 0.30</u>	95.89 ± 0.26
DeepCluster-v2	57.84 ± 0.93	<u>82.56 ± 0.47</u>	<u>88.11 ± 0.46</u>	**89.34 ± 0.40**	<u>84.38 ± 0.49</u>	94.62 ± 0.36	96.57 ± 0.24
SwAV	55.88 ± 0.89	80.30 ± 0.49	86.93 ± 0.51	<u>89.13 ± 0.41</u>	81.94 ± 0.54	94.58 ± 0.36	**96.64 ± 0.24**
Supervised	**71.97 ± 0.83**	**85.80 ± 0.40**	**90.24 ± 0.42**	88.23 ± 0.44	**85.26 ± 0.48**	**98.54 ± 0.16**	<u>96.61 ± 0.24</u>

capture the fact that an absolute increase of 1% in accuracy has varying significance depending on if, e.g., the accuracy goes from 50% to 51% or if it goes from 98% to 99%, we apply a logit-transformation to any metric that is bounded in the range 0 to 1.

All correlations computed against ImageNet performance use the logit-transformed ImageNet top-1 accuracy. Additionally, we logit-transform all recognition accuracies, AP metrics from detection, $11.25°$, $22.5°$ and $30°$ in surface normal estimation, and both mean-IOU and accuracy in semantic segmentation. The only metrics not transformed in this way are the Mean and Median errors in surface normal estimation. We negate these two error metrics before computing correlations in Fig. 2 so reading the figure is easier.

For correlations in Fig. 1, we average the logit-transformed accuracies across datasets in all many-shot and few-shot settings to produce a single correlation coefficient for each setting. For both detection settings we report the correlation of the logit-transformed AP50 metric and for the two dense settings we report correlations of the logit-transformed $11.25°$ and mean-IOU metrics.

For calibration (Fig. 3), perceptual similarity and attentive diffusion (Table 9), we similarly use logit-transformed values when computing correlations. For the red, green and blue colour channel errors in our image reconstruction, we report correlations of their raw values.

A.8. Image reconstruction by feature inversion

To see what information is retained by the models, we evaluate how well an image can be reconstructed from an extracted feature. We follow the deep image prior [55] protocol of feature inversion. Given an image I, we first extract its feature vector $f(I)$ by passing it through the pre-trained model backbone f. Next, we initialise a reconstruction network g_θ, parameterised by θ, which maps from a fixed noise input z to an image $g_\theta(z)$. The reconstruction network is then trained to output an image which, when passed through our pre-trained backbone, produces a feature close to that of I. The optimisation problem is:

$$\arg\min_\theta f(g_\theta(z)) - f(I). \qquad (1)$$

We extract the features from our pre-trained backbone from the 4th residual block, giving a vector size of $2048 \times 7 \times 7$. The reconstruction network is trained for 3000 iterations using the Adam optimiser with a learning rate of 0.001. The architecture of the reconstruction network is the same as in the original deep image prior paper [55] and the study in [69].

A.9. Computing the saliency maps

We use an occlusion mask of 10×10 pixels and pass it over images resized to 242×242 which we then crop to 224×224 to ensure all pixels are occluded the same number of times. The attention values are computed as the root relative squared error (RRSE) of the original features and

Table 8. Few-shot transfer of pre-trained models using prototypical networks. Here, we present few-shot transfer results for 5-way 5-shot and 5-way 50-shot settings on CD-FSL. We report the average accuracy and 95% confidence interval over 600 test episodes. Results style: **best**, <u>second best</u>.

	CropDiseases		EuroSAT		ISIC		ChestX	
	5-shot	50-shot	5-shot	50-shot	5-shot	50-shot	5-shot	50-shot
InsDis	88.01 ± 0.58	92.70 ± 0.43	81.29 ± 0.63	88.25 ± 0.47	43.90 ± 0.55	55.76 ± 0.50	25.67 ± 0.42	31.77 ± 0.44
MoCo-v1	87.87 ± 0.58	92.87 ± 0.42	81.32 ± 0.61	87.72 ± 0.46	**44.42 ± 0.55**	56.81 ± 0.52	25.92 ± 0.45	32.74 ± 0.43
PCL-v1	72.89 ± 0.69	82.83 ± 0.55	66.56 ± 0.76	76.41 ± 0.63	33.21 ± 0.48	39.77 ± 0.45	23.33 ± 0.40	27.40 ± 0.42
PIRL	86.22 ± 0.63	92.18 ± 0.44	82.14 ± 0.63	88.55 ± 0.44	43.89 ± 0.54	<u>56.89 ± 0.52</u>	25.60 ± 0.41	31.44 ± 0.47
PCL-v2	87.57 ± 0.60	93.57 ± 0.40	81.10 ± 0.54	89.23 ± 0.37	37.47 ± 0.52	46.82 ± 0.46	24.87 ± 0.42	30.56 ± 0.43
SimCLR-v1	90.29 ± 0.52	94.49 ± 0.37	82.78 ± 0.56	90.55 ± 0.36	<u>43.99 ± 0.55</u>	56.16 ± 0.53	26.36 ± 0.44	33.16 ± 0.47
MoCo-v2	87.62 ± 0.60	93.61 ± 0.40	84.15 ± 0.52	89.83 ± 0.37	42.60 ± 0.55	55.68 ± 0.53	25.26 ± 0.44	32.20 ± 0.43
SimCLR-v2	90.80 ± 0.52	95.80 ± 0.29	86.45 ± 0.49	92.07 ± 0.30	43.66 ± 0.58	56.83 ± 0.54	26.34 ± 0.44	33.23 ± 0.47
SeLa-v2	90.96 ± 0.54	95.40 ± 0.33	84.56 ± 0.57	88.51 ± 0.59	39.97 ± 0.55	51.31 ± 0.52	25.60 ± 0.44	32.81 ± 0.44
InfoMin	87.77 ± 0.61	92.93 ± 0.40	81.68 ± 0.59	87.61 ± 0.43	39.03 ± 0.55	51.58 ± 0.51	25.78 ± 0.44	31.58 ± 0.44
BYOL	92.71 ± 0.47	96.69 ± 0.27	83.64 ± 0.54	90.46 ± 0.35	43.09 ± 0.56	**58.03 ± 0.52**	26.39 ± 0.43	**34.17 ± 0.45**
DeepCluster-v2	**93.63 ± 0.44**	**97.04 ± 0.27**	**88.39 ± 0.49**	<u>93.07 ± 0.31</u>	40.73 ± 0.59	53.65 ± 0.54	<u>26.51 ± 0.45</u>	**34.17 ± 0.48**
SwAV	<u>93.49 ± 0.46</u>	<u>96.72 ± 0.28</u>	<u>87.29 ± 0.54</u>	**93.36 ± 0.31**	39.66 ± 0.54	51.10 ± 0.50	**26.54 ± 0.48**	<u>33.86 ± 0.46</u>
Supervised	89.37 ± 0.55	94.32 ± 0.36	83.81 ± 0.55	89.62 ± 0.37	39.38 ± 0.58	52.54 ± 0.56	25.22 ± 0.41	32.34 ± 0.45

Table 9. Numerical values for the results presented in Figs 3-4. Columns 1-4: Expected calibration error (ECE) using 15 bins for unscaled models and models further calibrated using temperature scaling. Columns 5-7: Average perceptual distance computed on reconstructed images, using three different measures of the Learned Perceptual Image Patch Similarity (LPIPS) metric [66]. Columns 8-10: Mean squared errors between the colour channels of reconstructed and original images. Column 11: Attentive diffusion measured as the percentage of attention values above the mean attention over an image. Higher value means wider attention. Results style: **lowest**, <u>second lowest</u>.

	Many-shot (Linear)		Many-shot (Finetune)		Perceptual Distance			Colour Error			Attention
	Unscaled	Scaled	Unscaled	Scaled	AlexNet	VGG	SqueezeNet	Red	Green	Blue	Diffusion
InsDis	12.68	2.72	8.15	2.18	0.58	0.71	0.48	3971	2734	3394	48.48
MoCo-v1	14.15	2.58	8.21	2.28	0.62	0.74	0.53	4073	3044	3512	47.92
PCL-v1	14.06	3.71	7.29	2.63	0.74	0.81	0.65	4598	3954	4141	**41.43**
PIRL	15.68	2.68	8.37	2.12	0.59	0.72	0.50	3607	3070	3435	48.12
PCL-v2	11.07	2.85	5.04	2.34	0.56	0.66	0.47	3008	2807	3101	43.91
SimCLR-v1	8.45	<u>2.13</u>	5.29	2.46	0.56	0.70	0.47	3224	2667	3223	46.07
MoCo-v2	9.25	2.67	6.01	2.25	0.54	0.67	0.45	3179	<u>2514</u>	<u>2695</u>	45.39
SimCLR-v2	9.71	2.19	6.06	2.45	0.55	0.68	0.46	3655	2855	3404	47.91
SeLa-v2	11.52	2.81	5.20	2.10	0.69	0.72	0.57	3962	3775	4315	47.68
InfoMin	**7.05**	2.99	5.32	2.23	<u>0.49</u>	<u>0.60</u>	<u>0.39</u>	**2592**	**2403**	**2594**	<u>43.73</u>
BYOL	10.23	**1.93**	5.82	1.96	0.59	0.71	0.48	3765	3268	3471	48.81
DeepCluster-v2	8.69	2.17	4.94	**1.85**	0.58	0.67	0.48	3527	3170	3804	48.69
SwAV	<u>8.25</u>	2.16	<u>4.80</u>	<u>1.86</u>	0.57	0.67	0.46	3560	3186	3565	49.47
Supervised	10.35	2.22	**4.48**	1.90	**0.47**	**0.55**	**0.37**	<u>2788</u>	2917	2903	43.88
Correlation to ImageNet	-0.77	-0.59	-0.90	-0.59	-0.51	-0.69	-0.57	-0.56	-0.11	-0.22	0.09

the occluded features, averaged over all times a pixel is occluded (10^2). The RRSE ensures that the distances are invariant to the scale of the original features.

Table 10. Training details as reported by original authors for all models used in this paper. Asterisks (*) note models we obtain from PyContrast instead of original authors.

|	Epochs	Batch size	Target net	Mom. enc.	Mem. bank	Proj. head	Jigsaw	Grayscale	Colour jitter	Solarize	Blur	Random crop	Horiz. flip	Normalize
InsDis*	200	256			✓			✓	✓			✓	✓	✓
MoCo-v1	200	256		✓				✓	✓			✓	✓	✓
PCL-v1	200	256		✓				✓	✓			✓	✓	✓
PIRL*	200	1024			✓		✓		✓			✓	✓	✓
PCL-v2	200	256		✓		✓		✓	✓		✓	✓	✓	✓
SimCLR-v1	1000	4096				✓		✓	✓		✓	✓	✓	
MoCo-v2	800	256		✓		✓		✓	✓		✓	✓	✓	✓
SimCLR-v2	800	4096		✓		✓		✓	✓		✓	✓	✓	
SeLa-v2	400	4096			✓	✓		✓	✓		✓	multi	✓	✓
InfoMin	800	256		✓		✓	✓	✓	✓		✓	✓	✓	✓
BYOL	1000	4096	✓			✓		✓	✓	✓	✓	✓	✓	✓
DeepCluster-v2	800	4096			✓	✓		✓	✓		✓	multi	✓	✓
SwAV	800	4096				✓		✓	✓		✓	multi	✓	✓
Supervised	120	256							PCA			✓	✓	✓
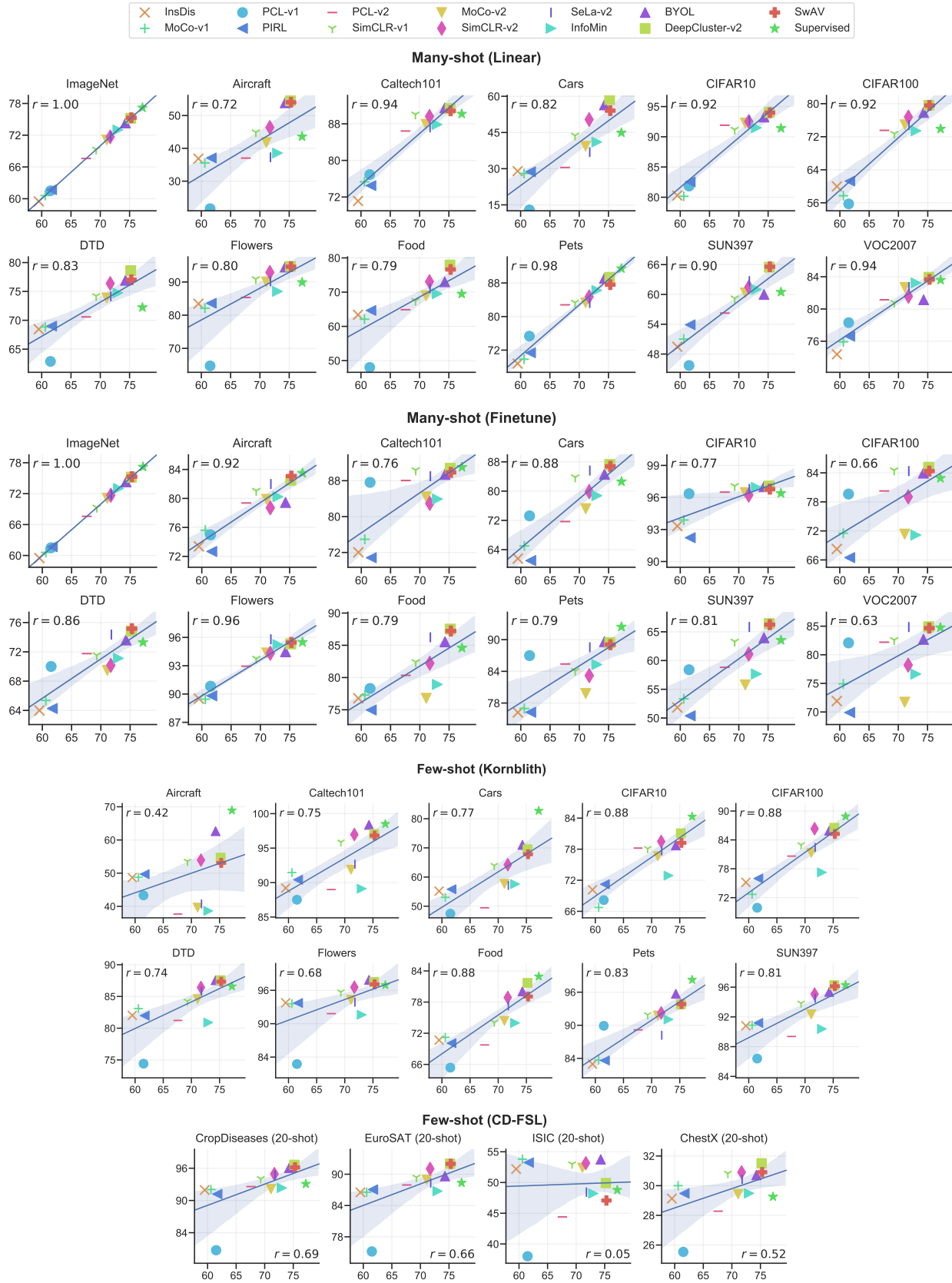
Figure 5. Individual plots of transfer correlations between ImageNet accuracy on the x-axis and target performance on the y-axis.
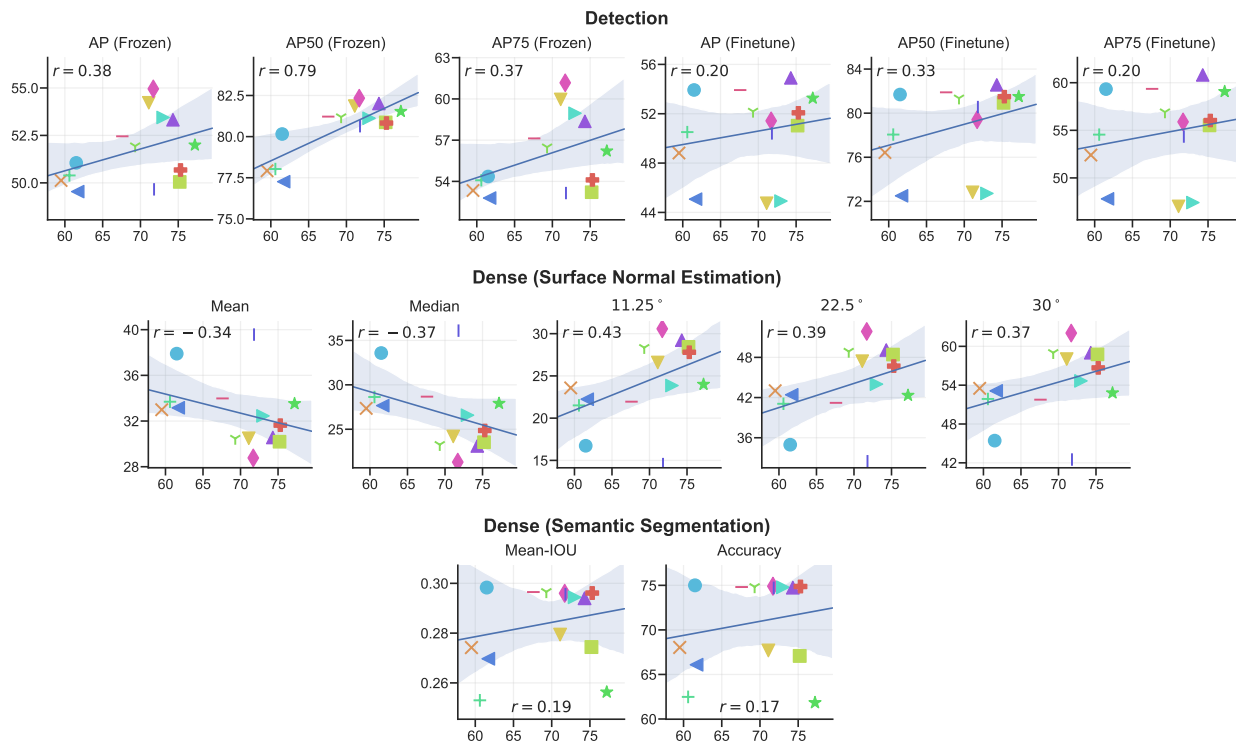
Figure 6. Individual plots of transfer correlations between ImageNet accuracy on the x-axis and target performance on the y-axis.
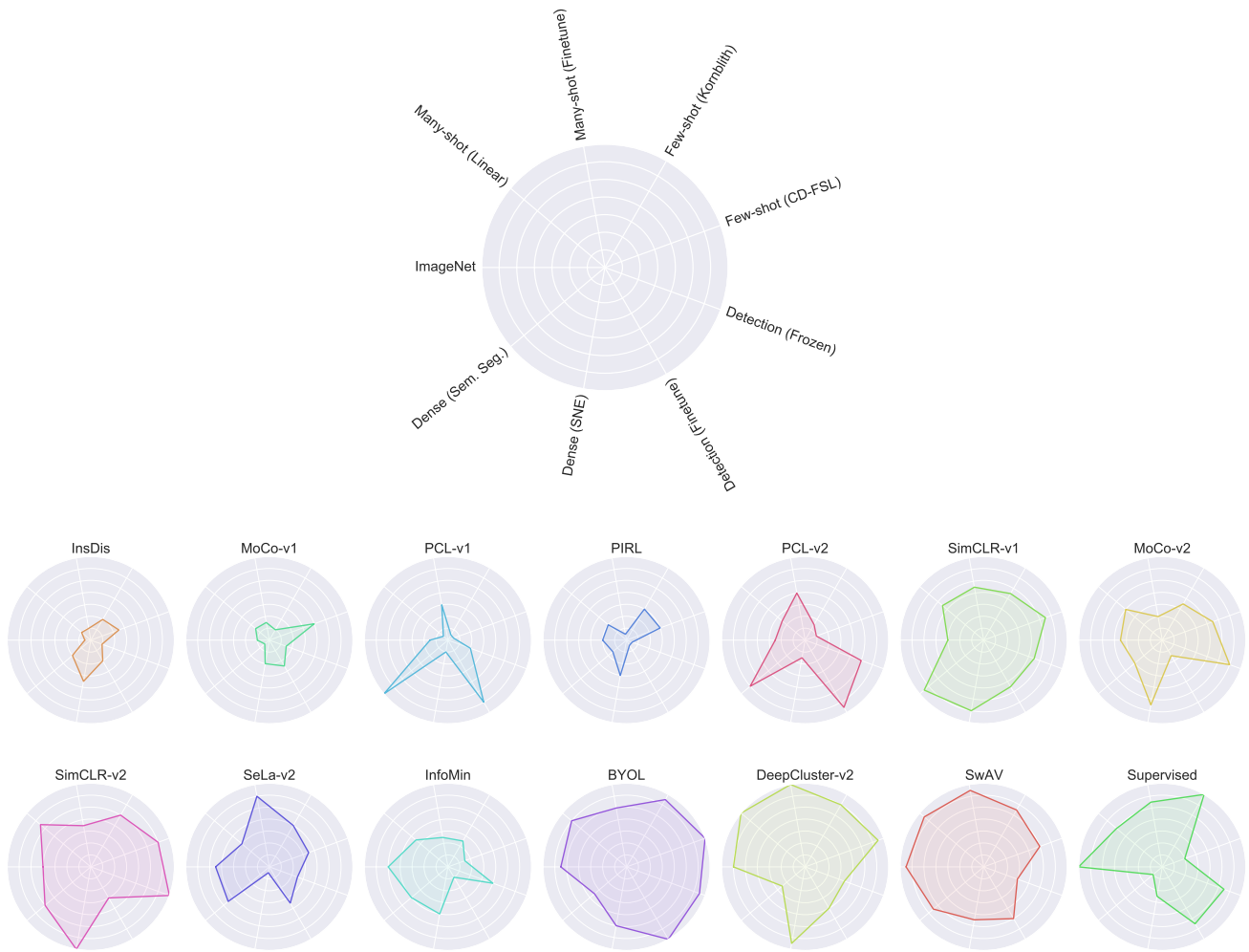
Figure 7. Radar charts of model performance on ImageNet and our eight different evaluation settings. In each setting we compute the rankings of the models (from averaged performance where there are multiple datasets). In each plot above, a higher rank (better performance) places the line closer to the outer edge of the circle. A larger total area roughly corresponds to better performance across a wide range of transfer settings. The rankings are based on average accuracy in the many-shot and few-shot settings, AP50 for frozen and finetuned detection, mean error for surface normal estimation and mean IOU for semantic segmentation.
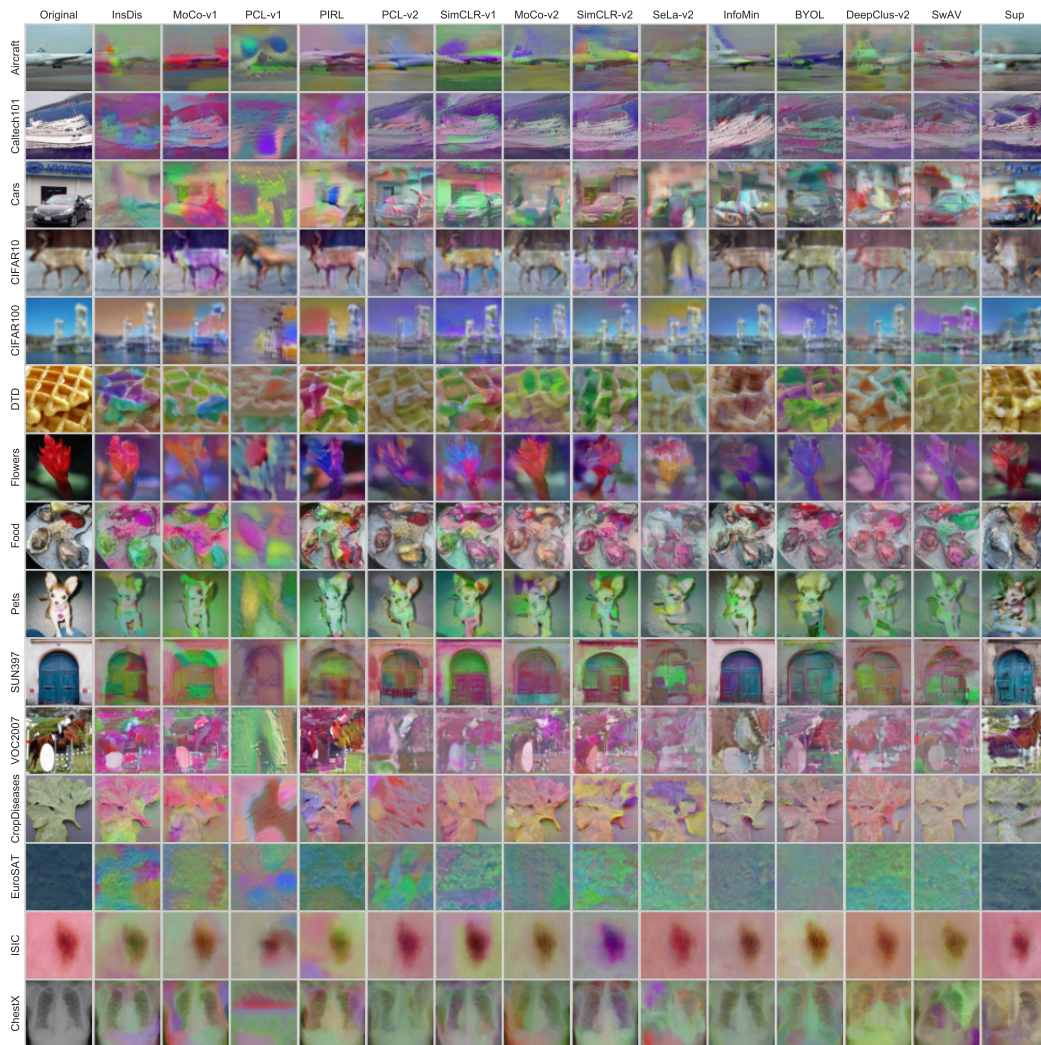
Figure 8. Deep image prior reconstructions on one image for each of 15 datasets.
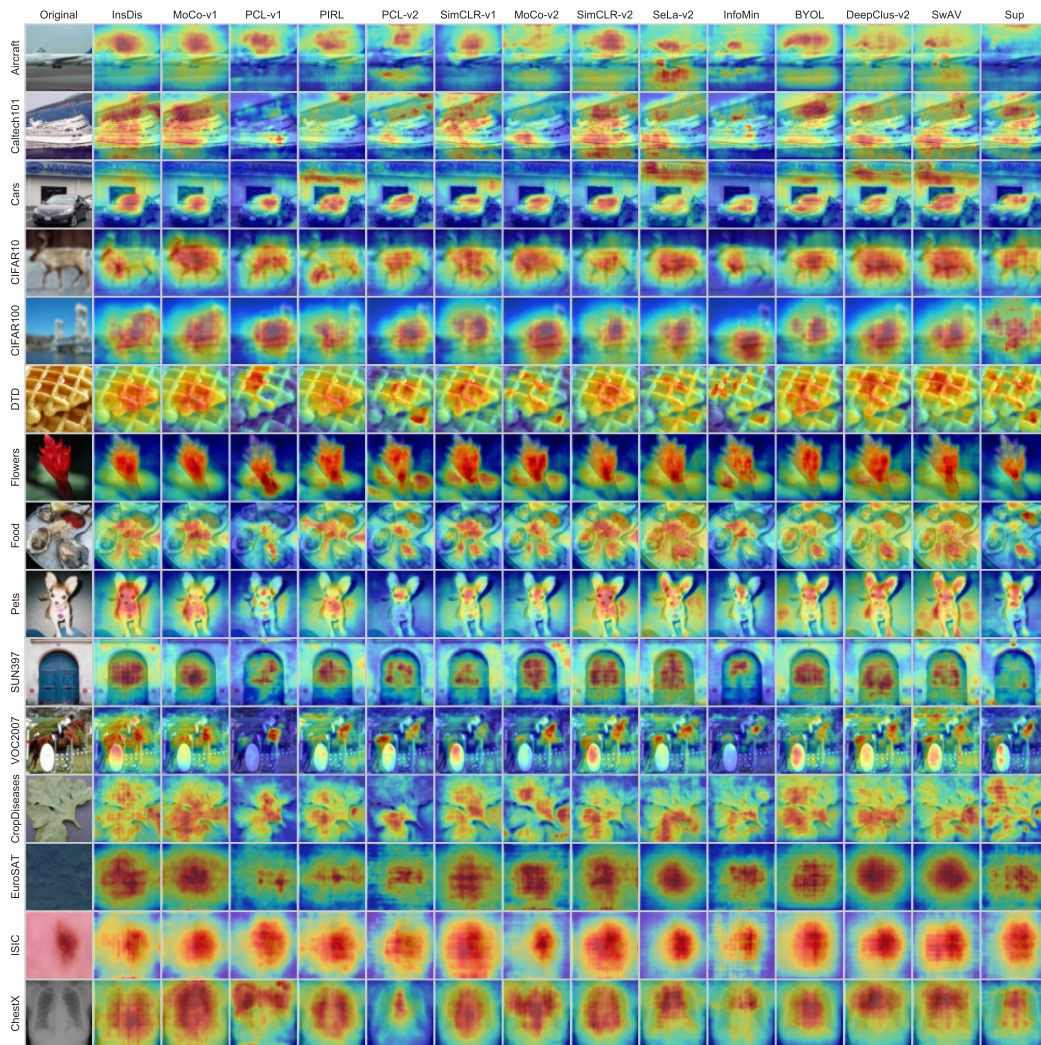
Figure 9. Saliency maps for all models on one image for each of 15 datasets.