Surprising Effectiveness of Few-Image Unsupervised Feature Learning

Yuki M. Asano Christian Rupprecht Andrea Vedaldi

Visual Geometry Group
University of Oxford
{yuki,chrisr,vedaldi}@robots.ox.ac.uk

Abstract

State-of-the-art methods for unsupervised representation learning can train well the first few layers of standard convolutional neural networks, but they are not as good as supervised learning for deeper layers. This is likely due to the generic and relatively simple nature of shallow layers; and yet, these approaches are applied to millions of images, scalability being advertised as their major advantage since unlabelled data is cheap to collect. In this paper we question this practice and ask whether so many images are actually needed to learn the layers for which unsupervised learning works best. Our main result is that a few or even a single *image together with strong data augmentation are sufficient to nearly saturate performance. Specifically, we provide an analysis for three different self-supervised feature learning methods (BiGAN, RotNet, DeepCluster) vs number of training images (1, 10, 1000) and show that we can top the accuracy for the first two convolutional layers of common networks using just a single unlabelled training image and obtain competitive results for other layers. We further study and visualize the learned representation as a function of which (single) image is used for training. Our results are also suggestive of which type of information may be captured by shallow layers in deep networks.*

1. Introduction

Despite tremendous progress in supervised learning, the ability of machines to learn without external supervision remains limited. In fact, the need for large annotated datasets is a major obstacle to deploying machine learning systems to new applications. On the other hand, representations learned by convolutional neural networks (CNNs) have been shown to transfer well between tasks [48, 41, 17, 21]. Unsupervised learning methods such as self-supervision exploit this capability by pre-training CNNs models using pretext tasks that do not require expensive manual annotations, and then transfer the representations to a target task with a modest amount of labelled data.

Figure 1: **Single-image self-supervision.** We show that several self-supervision methods can be used to train the first few layers of a deep neural networks using a *single training image*, such as this Image A, as input. This is achieved by modulating data augmentation, and it is possible because these layers are sufficiently generic.

The degree of success of unsupervised representation learning depends on the nature of the image representation. Deep neural networks extract a sequence of increasingly more abstract representations of images, with shallower layers capturing lower-level and thus more generically-applicable image statistics. These representations can be assessed individually by how well they can solve a reference task such as ImageNet classification. Using this as a criterion, authors have shown that self-supervision is about as good as manual supervision for training the first few convolutional layers of a neural network, but that the performance degrades for deeper layers.

Usually this is explained by the fact that the first few layers extract low-level information [56], making high-level semantic information — which is captured by manual anno-

tations — less important to learn them. However, unsupervised learning methods still use hundreds of thousands if not millions of images to train these low-level layers. Just as manual labels are unnecessary, it is not clear if these large datasets are truly required to learn simpler features. More in general, studying the number of images required to learn different parts of a neural network can shed some light on the complexity of the statistics captured by these components.Thus, in this work we aim at answering a simple question: "*how many unlabelled images do we need to learn different parts of a neural network?*"

Our main finding is that the shallower layers of common deep networks can be learned using surprisingly few example images. In fact, we show that *a single training image* is sufficient to perfectly train the early layers in a standard architecture such as AlexNet. This strongly confirms the very generic nature of the early convolutional layers, and that training from large datasets is in this case entirely unnecessary. This is more than just a theoretical finding, as these low-level feature extractors are useful for upstream tasks such as template matching [27, 50] and style transfer [14, 26], which currently rely on pre-training with millions of images.

While the main aim of the paper is to study the ultra low-data regime in self-supervision, which is complementary to the typical approach of using increasingly larger datasets, as a byproduct we also show how self-supervised learning can be applied to this regime effectively. The key is to modulate the amount of data augmentation. When a single training image is considered, in particular, we extract a large number of random crops from it, and apply standard augmentation transforms to those crops; this is equivalent to modifying the statistics of data augmentations used to train standard neural networks, making them more extreme. This also allows to interpolate between two data augmentation settings, one for the ultra low-data regime, and the other for big data.

We also study which self-supervised methods can operate successfully in the low-data regime. To this end, we investigate BiGAN, RotNet and DeepCluster as three methods representative of different approaches to unsupervised learning. We find that performance as a function of the amount of data is highly dependent on the method, but all three methods analyzed can indeed leverage a single image to learn the first few layers of a deep network well.

2. Related Work

Our paper relates to two broad areas of research: (a) self-supervised/unsupervised learning, and (b) learning from a single sample. We discuss closely related works for each.

Self-supervised learning: A wide variety of proxy tasks requiring no manual annotations have been proposed for the self-training of deep convolutional neural networks. These methods use various cues and tasks namely, in-painting [44], patch context and jigsaw puzzles [7, 37, 39, 36], clustering [4], noise-as-targets [3], colorization [58, 31], generation [25, 45, 8], geometry [9, 16] and counting [38]. The idea is that the pretext task can be constructed automatically and easily on images alone. Thus, methods often modify information in the images and require the network to recover them. In-painting or colorization techniques fall in this category. However these methods have the downside that the features are learned on modified images which potentially harms the generalization to unmodified ones. For example, colorization uses a gray scale image as input, thus the network cannot learn to extract color information, which can be important for other tasks.

Slightly less related are methods that use additional information to learn features. Here, often temporal information is used in the form of videos. Typical pretext tasks are based on temporal-context [35, 55, 32, 47], spatio-temporal cues [22, 13, 54], foreground-background segmentation via video segmentation [43], optical-flow [12, 33], future-frame synthesis [49], audio prediction from video [5, 42], audio-video alignment [2], ego-motion estimation [23], slow feature analysis with higher order temporal coherence [24], transformation between frames [1] and patch tracking in videos [53]. Since we are interested in learning features from as little data as one image, we cannot make use of methods that rely on video input.

Our contribution inspects three unsupervised feature learning methods that use very different means of extracting information from the data: BiGAN [8] utilizes a generative adversarial task, RotNet [16] exploits the photographic bias in the dataset and DeepCluster [4] learns stable feature representations under a number of image transformations by proxy labels obtained from clustering. These are described in more detail in the Methods section.

Learning from a single sample: In some applications of computer vision, the bold idea of learning from a single sample comes out of necessity. For general object tracking, methods such as max margin correlation filters [46] learn robust tracking templates from a single sample of the patch. Single sample learning was pursued by the semi-parametric exemplar SVM model [34]. They learn one SVM per positive sample separating it from all negative patches mined from the background. While only one sample is used for the positive set, the negative set consists of thousands of images and is a necessary component of their method. The negative space was approximated by a multi-dimensional Gaussian by the Exemplar LDA [20]. These SVMs, one per positive sample, are pooled together using a max aggregation. We differ from both of these approaches in that we do not use a large collection of negative images to train our model. Instead we restrict ourselves to a single or a few

images with a systematic augmentation strategy.

3. Methods

We discuss first our data augmentation strategy and selection of source images (Section 3.1); then, we describe three diverse methods for unsupervised feature learning which we compare in the experimental section (Section 3.2). As we will see, these methods differ in their ability to learn well from augmented data.

3.1. Data

Our goal is to understand the performance of representation learning methods as a function of the image data used to train them. To make comparisons as fair as possible, we develop a protocol where only the nature of the training data is changed, but all other parameters remain fixed.

In order to do so, given a baseline method trained on d source images, we replace those with another set of d images. Of these, now only $N \ll d$ are *source* images (i.e. i.i.d. samples), while the remaining $d-N$ are augmentations of the source ones. Thus, the amount of information in the training data is controlled by N and we can generate a continuum of datasets that vary from one extreme, utilizing a single source image $N=1$, to the other extreme, using all $N=d$ original training set images. For example, if the baseline method is trained on ImageNet, then $d = 1{,}281{,}167$. When $N=1$, it means that we train the method using a single source image and generate the remaining $1{,}281{,}166$ images via augmentation. Other baselines use CIFAR-10/100 images, so in those cases $d=50{,}000$ instead.

The data augmentation protocol, detailed next, is an extreme version of augmentations already employed by most deep learning protocols. Each method we test, in fact, already performs some data augmentation internally. Thus, when the method is applied on our augmented data, this can be equivalently thought of as incrementing these "native" augmentations by concatenating them with our own.

Choice of augmentations. Next, we describe how the N source images are expanded to additional $d-N$ images so that the models can be trained on exactly d images, independent from the choice of N. The idea is to use an aggressive form of data augmentation involving cropping, scaling, rotation, contrast changes, and adding noise. These transformations are representative of invariances that one may wish to incorporate in the features. Augmentation can be seen as imposing a prior on how we expect the manifold of natural images to look like. When training with very few images, these priors become more important since the model cannot extract them directly from data.

Given a source image of size size $H \times W$, we first extract a certain number of random patches of size (w, h), with $w \leq W, h \leq H$, and the constraints

$$\beta \leq \frac{wh}{WH}, \quad \gamma \leq \frac{h}{w} \leq \gamma^{-1}.$$

Thus, the smallest size of the crops is limited to be at least βWH and at most the whole image. Additionally, changes to the aspect ratio are limited by γ. In practice we use $\beta = 10^{-3}$ and $\gamma = \frac{3}{4}$.

Second, good features should not change much by small image rotations, so images are rotated (before cropping to avoid border artifacts) by $\alpha \in (-35, 35)$ degrees. Due to symmetry in image statistics, images are also flipped left-to-right with 50% probability.

Illumination changes are common in natural images, we thus expect image features to be robust to color and contrast changes. Thus, we employ a set of linear transformations in RGB space to model this variability in real data. Additionally, the color/intensity of single pixels should not affect the feature representation, as this does not change the contents of the image. To this end, color jitter with additive brightness, contrast and saturation are sampled from three uniform distributions in $(0.6, 1.4)$ and hue noise from $(-0.1, 0.1)$ is applied to the image patches. Finally, the cropped and transformed patches are scaled to the color range $(-1, 1)$ and then rescaled to full $S \times S$ resolution to be supplied to each representation learning method, using bilinear interpolation. This formulation ensures that the patches are created in the target resolution S, independent from the size and aspect ratio W, H of the source image.

Real samples. The images used for the $N=1$ and $N=10$ experiments are shown in Figures 1 to 3, respectively (this is *all* the training data used in such experiments). For the special case of using a single training image, i.e. $N = 1$, we have chosen one photographic (2560×1920) and one drawn image (600×225), which we call *Image* `A` and *Image* `B`, respectively. The two images were selected as they contain rich texture and are diverse, but are otherwise not optimized for performance. We test only two images due to the cost of running a full set of experiments (each image is expanded up to 1.2M times for training some of the models, as explained above). However, this is sufficient to prove our main points. While resolution matters to some extent as a bigger image contains more pixels, the information within is still far more correlated, and thus more redundant than sampling several smaller images. In particular, the resolution difference in Image `A` and `B` appears to be negligible in our experiments. For CIFAR-10, where $S=32$ we only use Image `B` due to the resolution difference. In direct comparison, Image `B` is the size of about 132 CIFAR images which is still much less than $d=50{,}000$. For $N>1$, we select the source images randomly from each method's training set.

Figure 2: Image B used for the $N = 1$ experiments.

Figure 3: ImageNet images for the $N = 10$ experiments.

3.2. Representation Learning Methods

We give a brief overview of the three self-supervised methods analyzed in this work.

Generative models. Generative Adversarial Networks (GANs) [18] learn to generate images using an adversarial objective: a generator network maps noise samples to image samples, approximating a target image distribution and a discriminator network is tasked with distinguishing generated and real samples. Generator and discriminator are pitched one against the other and learned together; when an equilibrium is reached, the generator produces images indistinguishable (at least from the viewpoint of the discriminator) from real ones. Bidirectional Generative Adversarial Networks (BiGAN) [8, 10] are a recent extension of GANs designed to learn a useful image representation as an approximate inverse of the generator through joint inference on an encoding and the image. This method's native augmentation uses random crops and random horizontal flips to learn features from $S = 128$ sized images. As opposed to the other two methods discussed below it employs leaky ReLU non-linearities as is typical in GAN discriminators.

Rotation. Most image datasets contain pictures that are 'upright' as this is how humans prefer to take and look at them. This photographer bias can be understood as a form of implicit data labelling. RotNet [16] exploits this by tasking a network with predicting the upright direction of a picture after applying to it a random rotation multiple of 90 degrees (in practice this is formulated as a 4-way classification problem). The authors reason that the concept of 'upright' requires learning high level concepts in the image and hence this method is not vulnerable to exploiting low-level visual information, encouraging the network to learn more abstract features. In our experiments, we test this hypothesis by learning from impoverished datasets that may lack the photographer bias. The native augmentations that RotNet uses on the $S = 256$ inputs only comprise horizontal flips and non-scaled random crops to 224×224.

Table 1: **Ablating data augmentation using MonoGAN.** Training a linear classifier on the features extracted at different depths of the network for CIFAR-10.

	CIFAR-10			
	conv1	conv2	conv3	conv4
(a) Fully sup.	66.5	70.1	72.4	75.9
(b) Random feat.	57.8	55.5	54.2	47.3
(c) No aug.	57.9	56.2	54.2	47.8
(d) Jitter	58.9	58.0	57.0	49.8
(e) Rotation	61.4	58.8	56.1	47.5
(f) Scale	67.9	69.3	67.9	59.1
(g) Rot. & jitter	64.9	63.6	61.0	53.4
(h) Rot. & scale	67.6	69.9	68.0	60.7
(i) Jitter & scale	68.1	71.3	69.5	62.4
(j) All	**68.1**	**72.3**	**70.8**	**63.5**

Clustering. DeepCluster [4] is a recent state-of-the-art unsupervised representation learning method. This approach alternates k-means clustering to produce pseudo-labels for the data and feature learning to fit the representation to these labels. The authors attribute the success of the method to the prior knowledge ingrained in the structure of the convolutional neural network [52].

The method alternatives between a clustering step, in which k-means is applied on the PCA-reduced features with $k = 10^3$, and a learning step, in which the network is trained to predict the cluster ID for each image under a set of augmentations (random resized crops with $\beta = 0.08, \gamma = \frac{3}{4}$ and horizontal flips) that constitute its native augmentations used on top of the $S = 256$ input images.

4. Experiments

We evaluate the representation learning methods on ImageNet and CIFAR-10/100 using linear probes (Section 4.1). After ablating various choices of transformations in our augmentation protocol (Section 4.2), we move to the core question of the paper: whether a large dataset is beneficial to unsupervised learning, especially for learning early convolutional features (Section 4.3).

Table 2: **ImageNet LSVRC-12 linear probing evaluation.** A linear classifier is trained on the (downsampled) activations of each layer in the pretrained model. We report classification accuracy averaged over 10 crops. The column [ref] indicates which publication the reported numbers are borrowed from.

	Method	#images	[ref]	ILSVRC-12 conv1	conv2	conv3	conv4	conv5
(a)	Full-supervision	1,281,167	[59]	19.3	36.3	44.2	48.3	50.5
(b)	Random	0	[59]	11.6	17.1	16.9	16.3	14.1
(c)	Krähenbühl *et al.* [28]: Random Rescaled	0	[28]	17.5	23.0	24.5	23.2	20.6
(d)	Donahue *et al.* [8]: BiGAN	1,281,167	[59]	17.7	24.5	31.0	29.9	28.0
(e)	mono, Image `A`	1		20.4	30.9	33.4	28.4	16.0
(f)	mono, Image `B`	1		20.5	30.4	31.6	27.0	16.8
(g)	deka	10		16.2	16.5	16.5	13.1	7.5
(h)	kilo	1,000		16.1	17.7	18.3	17.6	13.5
(i)	Gidaris *et al.* [16]: RotNet	1,281,167	[16]	18.8	31.7	38.7	38.2	36.5
(j)	mono, Image `A`	1		19.9	30.2	30.6	27.6	21.9
(k)	mono, Image `B`	1		17.8	27.6	27.9	25.4	20.2
(l)	deka	10		19.6	30.7	32.6	28.9	22.6
(m)	kilo	1,000		21.0	33.5	36.5	34.0	29.4
(n)	Caron *et al.* [4]: DeepCluster	1,281,167	[4]	18.0	32.5	39.2	37.2	30.6
(o)	mono, Image `A`	1		20.7	31.5	32.5	28.5	21.0
(p)	mono, Image `B`	1		19.7	30.1	31.6	28.5	20.4
(q)	deka	10		18.5	29.0	31.1	28.2	21.9
(r)	kilo	1,000		19.5	29.8	33.0	31.7	26.8

4.1. Linear probes and baseline architecture

In order to quantify if a neural network has learned useful feature representations, we follow the standard approach of using linear probes [59]. This amounts to solving a difficult task such as ImageNet classification by training a linear classifier on top of pre-trained feature representations, which are kept fixed. Linear classifiers heavily rely on the quality of the representation since their discriminative power is low.

We apply linear probes to all intermediate convolutional layers of networks and train on the ImageNet LSVRC-12 [6] and CIFAR-10/100 [29] datasets, which are the standard benchmarks for evaluation in self-supervised learning. Our base encoder architecture is AlexNet [30], since this is a good representative model and is most often used in other unsupervised learning work for the purpose of benchmarking. This model has five convolutional blocks (each comprising a linear convolution later followed by ReLU and optionally max pooling). We insert the probes right after the ReLU layer in each block, and denote these entry points `conv1` to `conv5`. Applying the linear probes at each convolutional layer allows studying the quality of the representation learned at different depths of the network.

Details. While linear probes are conceptually straightforward, there are several technical details that can affect the final accuracy by a few percentage points. Unfortunately, prior work has used several slightly different setups, so that comparing numbers between different publications must be done with caution. To make matter more difficult, not all papers released evaluation source code. Upon acceptance, we will release the implementation of all of our experiments, including the evaluation code.

In our implementation, we follow the original proposal [59] in pooling each representation to a vector with $9600, 9216, 9600, 9600, 9216$ dimensions for `conv1-5` using adaptive max-pooling, and absorb the batch normalization weights into the preceding convolutions. For evaluation on ImageNet we follow RotNet to train linear probes: images are resized such that the shorter edge has a length of 256 pixels, random crops of 224×224 are computed and flipped horizontally with 50% probability. Learning lasts for 36 epochs and the learning rate schedule starts from 0.01 and is divided by five at epochs 5, 15 and 25. The top-1 accuracy of the linear classifier is then measured on the ImageNet validation subset. This uses DeepCluster's protocol, extracting 10 crops for each validation image (four at the corners and one at the center along with their horizontal flips) and averaging the prediction scores before the accu-

racy is computed. For CIFAR-10/100 data, we follow the same learning rate schedule and for both training and evaluation we do not reduce the dimensionality of the representations and keep the images' original size of 32×32.

4.2. Effect of Augmentations

In order to better understand which image transformations are important to learn a good feature representations, we analyze the impact of augmentation settings. For speed, these experiments are conducted using the CIFAR-10 images ($d = 50,000$ in the training set) and with the smaller source Image `B` and a GAN using the Wasserstein GAN formulation with gradient penalty [19]. The encoder is a smaller AlexNet-like CNN consisting of four convolutional layers (kernel sizes: $7, 5, 3, 3$; strides: $3, 2, 2, 1$) followed by a single fully connected layer as the discriminator. Given that the GAN is trained on a single image (plus augmentations), we call this setting *MonoGAN*.

Table 1 reports all 2^3 combinations of the three main augmentations (scale, rotation, and jitter) and a randomly initialized network baseline (see Table 1 (b)) using the linear probes protocol discussed above. Without data augmentation the model only achieves marginally better performance than the random network (which also achieves a non-negligible level of performance [51, 4]). This is understandable since the dataset literally consists of a single training image cloned d times. Color jitter and rotation slightly improve the performance of all probes by 1-2% points, but random rescaling adds at least ten points at every depth (see Table 1 (f,h,i)) and is the most important single augmentation. A similar conclusion can be drawn when two augmentations are combined, although there are diminishing returns as more augmentations are combined. Overall, we find all three types of augmentations are of importance when training in the ultra-low data setting.

4.3. Benchmark evaluation

We analyze how performance varies as a function N, the number of actual samples that are used to generated the augmented datasets, and compare it to the gold-standard setup (in terms of choice of training data) defined in the papers that introduced each method. The evaluation is again based on linear probes (Section 4.1).

Mono is enough. From Table 2 we make the following observations. Training with just a single source image (e,f,i,k,o,p) is much better than random initialization (b) for all layers. More importantly, when comparing within pretext task, even with one image we are able to improve the quality of `conv1`–`conv3` features compared to full (unsupervised) ImageNet training for GAN based self-supervision (d-h). For the other methods (i-m, n-r) we reach and also surpass the performance for the first layer and are within 1.5% points for the second. Given that the best unsupervised performance for `conv2` is 32.5, our method using a single source Image `A` (Table 2, o) is remarkably close with 31.5.

Table 3: **CIFAR-10.** Accuracy of linear classifiers on features extracted at different depths of the network.

	CIFAR-10			
	`conv1`	`conv2`	`conv3`	`conv4`
Fully supervised	66.5	70.1	72.4	75.9
Random	57.8	55.5	54.2	47.3
RotNet	64.4	65.6	65.6	59.1
GAN (CIFAR-10)	67.7	**73.0**	**72.5**	**69.2**
MonoGAN	**68.1**	72.3	70.8	63.5

Image contents. While we surpass the GAN based approach of [8] for both single source images, we find more nuanced results for the other two methods: For RotNet, as expected, the photographic bias cannot be extracted from a single image. Thus its performance is low with little training data and increases together with the number of images (Table 2, i-m). When comparing Image `A` and `B` trained networks for RotNet, we find that the photograph yields better performance than the hand drawn animal image. This indicates that the method can extract rotation information from low level image features such as patches which is at first counter intuitive. Considering that the hand-drawn image does not work well, we can assume that lighting and shadows even in small patches can indeed give important cues on the up direction which can be learned even from a single (real) image. DeepCluster shows poor performance in `conv1` which we can improve upon in the single image setting (Table 2, n-p).

More than one image. While BiGAN fails to converge for $N \in \{10, 1000\}$, most likely due to issues in learning from a distribution which is neither whole images nor only patches, we find that both RotNet and DeepCluster improve their performance in deeper layers when increasing the number of training images. However, for `conv1` and `conv2`, a single image is enough. In deeper layers, DeepCluster seems to require large amounts of source images to yield the reported results as the deka- and kilo- variants start improving over the single image case (Table 2, n-r). This need for data also explains the gap between the two input images which have different resolutions. Summarizing Table 2, we can conclude that learning `conv1`, `conv2` and for the most part `conv3` (33.4 vs. 39.4) on over 1M images does not yield a significant performance increase over using one single training image — a highly unexpected result.

Table 4: **CIFAR-100.** Accuracy of linear classifiers on features extracted at different depths of the network.

	CIFAR-100			
	conv1	conv2	conv3	conv4
Fully supervised	38.7	43.6	44.4	46.5
Random	30.9	29.8	28.6	24.1
RotNet	36.0	35.9	34.2	25.8
GAN (CIFAR-100)	38.7	43.6	44.4	**46.5**
GAN (CIFAR-10)	39.6	46.0	**45.1**	39.9
MonoGAN	**39.9**	**46.9**	44.5	38.8

Generalization. In Table 3, we show the results of training linear classifiers for the CIFAR-10 dataset and compare against various baselines. We can find that the GAN trained on the smaller Image B outperforms all other methods including the fully-supervised trained one for the first convolutional layer. We also outperform the same architecture trained on the full CIFAR-10 training set using RotNet, which might be due to the fact that either CIFAR images do not contain much information about the orientation of the picture or because they do not contain as many objects as in ImageNet. While the GAN trained on the whole dataset outperforms the MonoGAN on the deeper layers, the gap stays very small until the last layer. These findings are also reflected in the experiments on the CIFAR-100 dataset shown in Table 4. Here we find that our method obtains the best performance for the first two layers, even against the fully supervised version. The gap between our mono variant and the other methods increases again with deeper layers, hinting to the fact that we cannot learn very high level concepts in deeper layers from just one single image. These results corroborate the finding that our method allows learning very generalizable early features that are not domain dependent.

4.4. Qualitative Analysis

Visual comparison of weights. In Figure 4, we compare the learned filters of all first-layer convolutions of an AlexNet trained with the different methods and a single image. First, we find that the filters closely resemble those obtained via supervised training: Gabor-like edge detectors and various color blobs. Second, we find that the look is not easily predictive of its performance, e.g. while generatively learned filters (BiGAN) show many edge detectors, its linear probes performance is about the same as that of DeepCluster which seems to learn many somewhat redundant point features. However, we also find that some edge detectors are required, as we can confirm from RotNet and DeepCluster trained on Image B, which yield less crisp filters and worse performances.

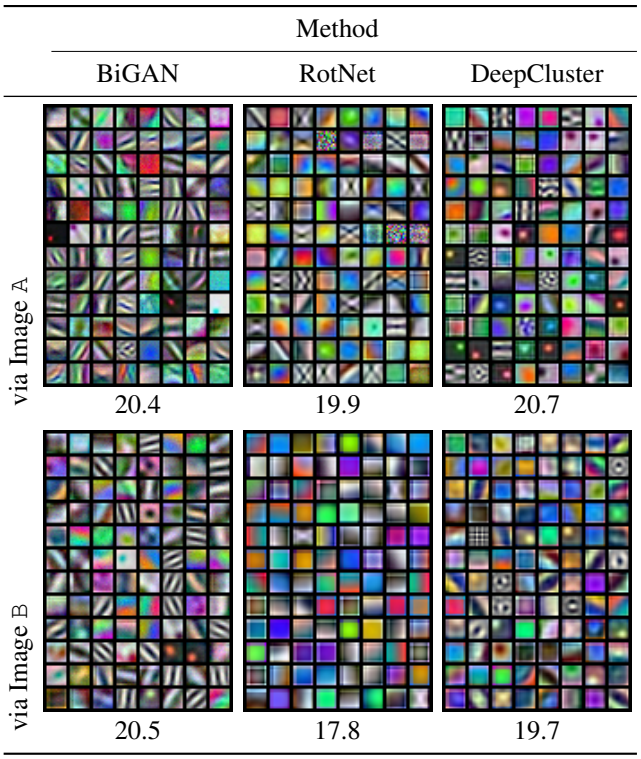

Figure 4: **conv1 filters trained using a single image.** The 96 learned $(3\times11\times11)$ filters for the first layer of AlexNet are shown for each single training image and method along with their linear classifier performance. For visualization, each filter is normalized to be in the range of $(-1, 1)$.

Visual comparison of filters. In order to understand what deeper neurons are responding to in our model, we visualize random neurons via activation maximization [11, 57] in each layer. Additionally, we retrieve the top-9 images in the ImageNet training set that activate each neuron most in Figure 5. Since the mono networks are not trained on the ImageNet dataset, it can be used here for visualization. From the first convolutional layer we find typical neurons strongly reacting to oriented edges. In layers 2-4 we find patterns such as grids (conv2:3), and textures such as leopard skin (conv2:2) and round grid cover (conv4:4). Confirming our hypothesis that the neural network is only extracting patterns and not semantic information, we do not find any neurons particularly specialized to certain objects even in higher levels as for example dog faces or similar which can be fund in supervised networks. This finding aligns with the observations of other unsupervised methods [4, 59]. As most neurons extract simple patterns and textures, the surprising effectiveness of training a network using a single image can be explained by the recent finding that even CNNs trained on ImageNet rely on texture (as opposed to shape) information to classify [15].

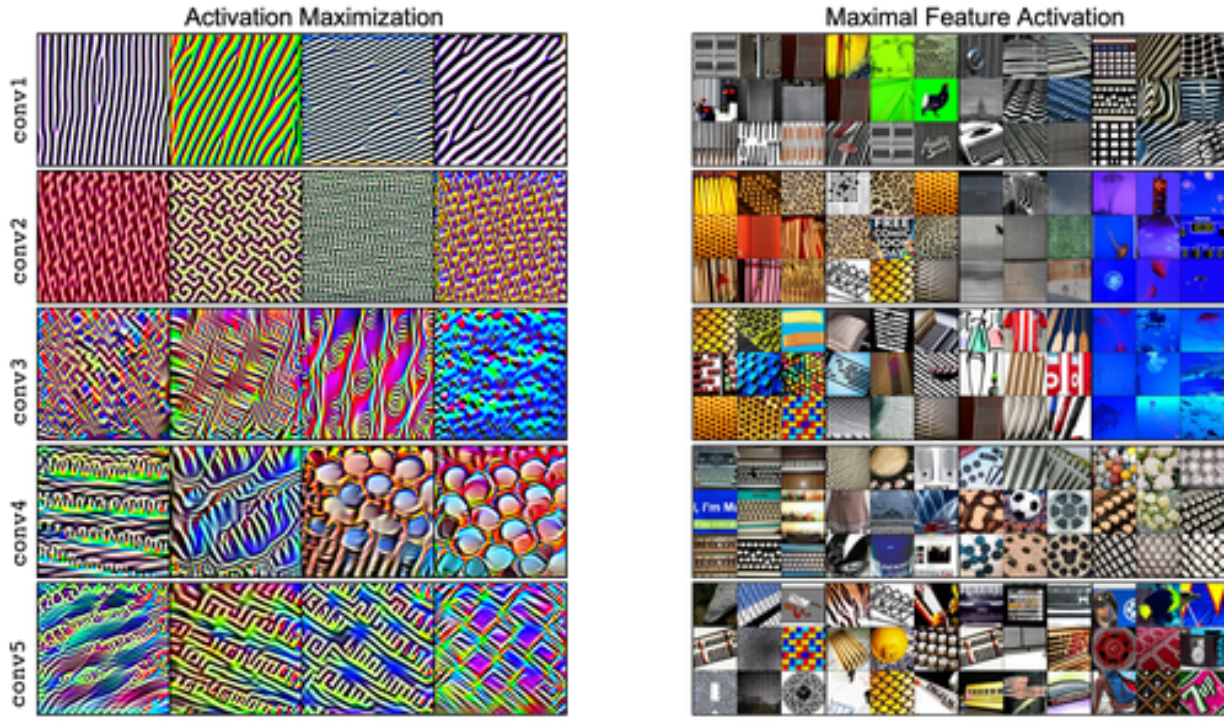

Figure 5: **Filter visualization.** We show activation maximization (left) and retrieval of top 9 activated images from the validation set of ImageNet (right) for four random non-cherrypicked target filters. From top to bottom: `conv1-5` of the BiGAN trained on a single image `A`. The filter visualization is obtained by learning a (regularized) input image that maximizes the response to the target filter using the library Lucid [40].

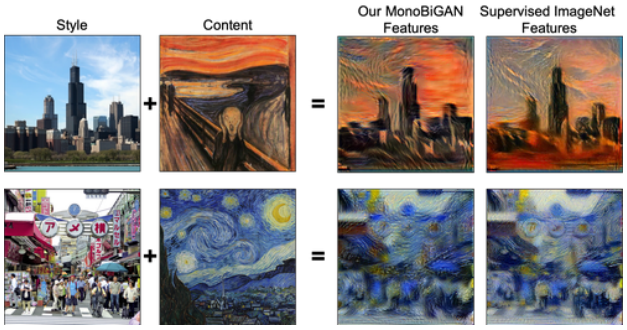

Figure 6: **Style transfer with single-image pretraining.** We show two style transfer results using the Image `A` trained BiGAN and the ImageNet pretrained AlexNet.

Neural style transfer. Lastly, we show how our features of a neural network trained on only a single image can be used for other applications. In Figure 6 we show two basic style transfers using the method of [14]. Image content and style are separated and the style is transferred from the source to target image using CNN features. We visually compare the results of using our features and from full ImageNet supervision. We find almost no visual differences in the stylized images and can conclude that our early features are equally powerful as fully supervised ones for this task.

5. Conclusions

While current research mostly focuses on using more and more data during training, we find it valuable to analyze the opposite scenario where we reduce the data as much as possible. We have made the surprising observation that we can learn good and generalizable features through self-supervision from one single source image. Additionally, we confirm that deeper layers do indeed need more training data to be able to learn meaningful object features. To extract as much information as possible out of a single training image, we need to perform aggressive augmentations to provide the network with enough variety for learning.

For the future, we are interested in two directions. 0-image learning could be achieved by training with procedurally generated images such as fractals. Second, one can search or optimize for the perfect single training image — conceptually, *the* prototypical image — which includes as much information about the visual world as possible.

Acknowledgements.

We thank Aravindh Mahendran for fruitful discussions. Yuki Asano gratefully acknowledges support from the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems (EP/L015897/1). Financial support was provided by ERC IDIU-638009.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, pages 37–45. IEEE, 2015. 2

[2] R. Arandjelović and A. Zisserman. Look, listen and learn. In *Proc. ICCV*, 2017. 2

[3] P. Bojanowski and A. Joulin. Unsupervised learning by predicting noise. In *Proc. ICML*, pages 517–526. PMLR, 2017. 2

[4] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proc. ECCV*, 2018. 2, 4, 5, 6, 7

[5] V. R. de Sa. Learning classification with unlabeled data. In *NIPS*, pages 112–119, 1994. 2

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 5

[7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. ICCV*, pages 1422–1430, 2015. 2

[8] J. Donahue, P. Krhenbhl, and T. Darrell. Adversarial feature learning. *Proc. ICLR*, 2017. 2, 4, 5, 6

[9] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *IEEE PAMI*, 38(9):1734–1747, Sept 2016. 2

[10] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016. 4

[11] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, Jun 2009. 7

[12] C. Gan, B. Gong, K. Liu, H. Su, and L. J. Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *Proc. CVPR*, 2018. 2

[13] R. Gao, D. Jayaraman, and K. Grauman. Object-centric representation learning from unlabeled videos. In *Proc. ACCV*, 2016. 2

[14] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. CVPR*, pages 2414–2423, 2016. 2, 8

[15] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 7

[16] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. ICLR*, 2018. 2, 4, 5

[17] R. B. Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 1

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 4

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5767–5777, 2017. 6

[20] B. Hariharan, J. Malik, and D. Ramanan. Discriminative decorrelation for clustering and classification. In *Proc. ECCV*, 2012. 2

[21] M. Huh, P. Agrawal, and A. A. Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 1

[22] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. Learning visual groups from co-occurrences in space and time. In *Proc. ICLR*, 2015. 2

[23] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. In *Proc. ICCV*, 2015. 2

[24] D. Jayaraman and K. Grauman. Slow and steady feature analysis: higher order temporal coherence in video. In *Proc. CVPR*, 2016. 2

[25] S. Jenni and P. Favaro. Self-supervised feature learning by learning to spot artifacts. In *Proc. CVPR*, 2018. 2

[26] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. ECCV*, pages 694–711, 2016. 2

[27] R. Kat, R. Jevnisek, and S. Avidan. Matching pixels using co-occurrence statistics. In *Proc. ICCV*, pages 1751–1759, 2018. 2

[28] P. Krähenbühl, C. Doersch, J. Donahue, and T. Darrell. Data-dependent initializations of convolutional neural networks. *Proc. ICLR*, 2016. 5

[29] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 5

[31] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *Proc. CVPR*, 2017. 2

[32] H.-Y. Lee, J.-B. Huang, M. K. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequence. In *Proc. ICCV*, 2017. 2

[33] A. Mahendran, J. Thewlis, and A. Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In *Proc. ACCV*, 2018. 2

[34] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proc. ICCV*, 2011. 2

[35] I. Misra, C. L. Zitnick, and M. Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016. 2

[36] T. Mundhenk, D. Ho, and B. Y. Chen. Improvements to context based self-supervised learning. In *Proc. CVPR*, 2017. 2

[37] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, pages 69–84. Springer, 2016. 2

[38] M. Noroozi, H. Pirsiavash, and P. Favaro. Representation learning by learning to count. In *Proc. ICCV*, 2017. 2

[39] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *Proc. CVPR*, 2018. 2

[40] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 3(3):e10, 2018. 8

[41] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks. In *Proc. CVPR*, 2014. 1

[42] A. Owens, P. Isola, J. H. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *Proc. CVPR*, pages 2405–2413, 2016. 2

[43] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *Proc. CVPR*, 2017. 2

[44] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. CVPR*, pages 2536–2544, 2016. 2

[45] Z. Ren and Y. J. Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. In *Proc. CVPR*, 2018. 2

[46] A. Rodriguez, V. N. Boddeti, B. V. Kumar, and A. Mahalanobis. Maximum margin correlation filter: A new approach for localization and classification. *IEEE Transactions on Image Processing*, 22(2):631–643, 2013. 2

[47] P. Sermanet et al. Time-contrastive networks: Self-supervised learning from video. In *Proc. Intl. Conf. on Robotics and Automation*, 2018. 2

[48] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE Trans. on Medical Imaging*, 35(5):1285–1298, May 2016. 1

[49] N. Srivastava, E. Mansimov, and R. Salakhudinov. Unsupervised learning of video representations using lstms. In *Proc. ICML*, 2015. 2

[50] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. In *Proc. CVPR*, pages 175–183, 2017. 2

[51] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proc. CVPR*, 2017. 6

[52] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proc. CVPR*, 2018. 4

[53] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. ICCV*, pages 2794–2802, 2015. 2

[54] X. Wang, K. He, and A. Gupta. Transitive invariance for self-supervised visual representation learning. In *Proc. ICCV*, 2017. 2

[55] D. Wei, J. Lim, A. Zisserman, and W. T. Freeman. Learning and using the arrow of time. In *Proc. CVPR*, 2018. 2

[56] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *NIPS*, pages 3320–3328, 2014. 1

[57] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, 2014. 7

[58] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*, pages 649–666. Springer, 2016. 2

[59] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. CVPR*, 2017. 5, 7