

Interpreting Deep Visual Representations via Network Dissection

Bolei Zhou¹, David Bau¹, Aude Oliva, and Antonio Torralba

Abstract—The success of recent deep convolutional neural networks (CNNs) depends on learning hidden representations that can summarize the important factors of variation behind the data. In this work, we describe *Network Dissection*, a method that interprets networks by providing meaningful labels to their individual units. The proposed method quantifies the interpretability of CNN representations by evaluating the alignment between individual hidden units and visual semantic concepts. By identifying the best alignments, units are given interpretable labels ranging from colors, materials, textures, parts, objects and scenes. The method reveals that deep representations are more transparent and interpretable than they would be under a random equivalently powerful basis. We apply our approach to interpret and compare the latent representations of several network architectures trained to solve a wide range of supervised and self-supervised tasks. We then examine factors affecting the network interpretability such as the number of the training iterations, regularizations, different initialization parameters, as well as networks depth and width. Finally we show that the interpreted units can be used to provide explicit explanations of a given CNN prediction for an image. Our results highlight that interpretability is an important property of deep neural networks that provides new insights into what hierarchical structures can learn.

Index Terms—Convolutional neural networks, network interpretability, visual recognition, interpretable machine learning

1 INTRODUCTION

OBSERVATIONS of hidden units in deep neural networks have revealed that human-interpretable concepts can emerge as individual latent variables within those networks. For example, object detector units emerge within networks trained to recognize places [1], part detectors emerge in object classifiers [2] and object detectors emerge in generative video networks [3]. This internal structure has appeared in situations where the networks are not constrained to decompose problems in any interpretable way.

The emergence of interpretable structure suggests that deep networks may be spontaneously learning disentangled representations. While a network can learn an efficient encoding that makes economical use of hidden variables to distinguish between inputs, the appearance of a disentangled representation is not well understood. A disentangled representation aligns its variables with a meaningful factorization of the underlying problem structure [4], or units that have a semantic interpretation (a face, wheel, green color, etc). Here, we address the following key issues:

- What is a disentangled representation of neural networks, and how can its factors be detected and quantified?
- Do interpretable hidden units reflect a special alignment of feature space?

- What differences in network architectures, data sources, and training conditions lead to the internal representations with greater or lesser entanglement?

We propose a general analytic framework, *Network Dissection*, for interpreting deep visual representations and quantifying their interpretability. Using a broadly and densely labeled dataset named Broden, our framework identifies hidden units' semantics for any given CNN, and aligns them with interpretable concepts.

Building upon [5], we provide a description of the methodology of Network Dissection in detail, and how it is used to interpret deep visual representations trained with different network architectures (AlexNet, VGG, GoogLeNet, ResNet, DenseNet) and supervision tasks (ImageNet for object recognition, Places for scene recognition, as well as other self-taught supervision tasks). We show that interpretability is an axis-aligned property of a representation that can be destroyed by rotation without affecting discriminative power. We further examine how interpretability is affected by different training datasets, training regularizations such as dropout [6] and batch normalization [7], as well as fine-tuning between different data sources. Our experiments reveal that units emerge as semantic detectors in the intermediate layers of most deep visual representations, while the degree of interpretability can vary widely across changes in architecture and training sets. We conclude that representations learned by deep networks are more interpretable than previously thought, and that measurements of interpretability provide insights about the structure of deep visual representations that are not revealed by their classification power alone.¹

1. Code, data, and more dissection results are available at the project page <http://netdissect.csail.mit.edu/>.

• The authors are with CSAIL, MIT, Cambridge, MA 02139.
E-mail: {bzhou, davidbau, oliva, torralba}@csail.mit.edu.

Manuscript received 14 Sept. 2017; revised 4 Apr. 2018; accepted 12 June 2018. Date of publication 22 July 2018; date of current version 13 Aug. 2019. (Corresponding author: Bolei Zhou.)

Recommended for acceptance by A. Vedaldi.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2858759

1.1 Related Work

Visualizing Deep Visual Representations. Though CNN models are often said to be black boxes, their behavior can be visualized at the *local individual unit level* by sampling image patches that maximize activation of hidden individual units [1], [8], [9], or the *global feature space level* by using variants of backpropagation to identify or generate salient image features [10], [11]. Back-propagation together with a natural image prior can be used to invert a CNN layer activation [12], and an image generation network can be trained to invert the deep features by synthesizing the input images [13]. [14] further synthesizes the prototypical images for individual units by learning a feature code for the image generation network from [13]. These visualizations reveal the visual patterns that have been learned and provide a qualitative guide to unit interpretation. In [1], human evaluation of visualizations is used to determine which individual units behave as object detectors in a network trained to classify scenes. However, human evaluation is not scalable to increasingly large networks such as ResNet [15]. Here, we introduce a scalable method to go from qualitative visualization to quantitative interpretation of large networks.

Analyzing the Properties of Deep Visual Representations. Much work has studied the power of CNN layer activations as generic visual features for classification [16], [17]. While transferability of layer activations has been explored, higher layer units remain most often specialized to the target task [18]. Susceptibility to adversarial input has shown that discriminative CNN models are fooled by particular visual patterns [19], [20]. Analysis of correlation between different random initialized networks reveals that many units converge to the same set of representations after training [21]. The question of how representations generalize has been investigated by showing that a CNN can easily fit a random labeling of training data even under explicit regularization [22].

Unsupervised Learning of Deep Visual Representations. Unsupervised learning or self-supervised learning works exploit the correspondence structure that comes for free from unlabeled images to train networks from scratch [23], [24], [25], [26], [27]. For example, CNNs have been trained by predicting image context [23], by colorizing gray images [28], [29], by solving image puzzle [24], and by associating the images with ambient sounds [30]. The resulting deep visual representations learned from different unsupervised learning tasks are compared by evaluating them to generic visual features on classification datasets such as Pascal VOC. Here, we provide an alternative approach to compare deep visual representations in terms of their interpretability, beyond their discriminative power.

2 FRAMEWORK OF NETWORK DISSECTION

The notion of a disentangled representation rests on human perception of what it means for a concept to be mixed up. Thus, we define the *interpretability* of deep visual representation as the degree of alignment with human-interpretable concepts. Our quantitative measurement of interpretability proceeds in three steps:

- 1) Identify a broad set of human-labeled visual concepts.
- 2) Gather the response of the hidden variables to known concepts.

- 3) Quantify alignment of hidden variable–concept pairs.

This three-step process of *network dissection* is reminiscent of neuroscientists' methods to characterize biological neurons [31]. Since our purpose is to measure the level to which a representation is disentangled, we focus on quantifying the correspondence between a single latent variable and a visual concept.

In a fully interpretable local coding such as a one-hot-encoding, each variable will match with one human-interpretable concept. Although we expect a network to learn partially nonlocal representations in interior layers [4], as past experience shows that an emergent concept will often align with a combination of a several hidden units [2], [17], our aim is to assess how well a representation is disentangled. Therefore we measure the alignment between single units and single interpretable concepts. This does not gauge the discriminative power of the representation; rather it quantifies its disentangled interpretability. As we will show in Section 3.2, it is possible for two representations of perfectly equivalent discriminative power to have different levels of interpretability.

To assess the interpretability of CNNs, we draw concepts from a new labeled image dataset that unifies visual concepts from a heterogeneous collection of datasets, see Section 2.1. We then measure the alignment of each CNN hidden unit with each concept by evaluating the feature activation of each individual unit as a segmentation model for each concept. To quantify the interpretability of a whole layer, we count the number of distinct concepts that are aligned with a unit, as detailed in Section 2.2.

2.1 Broden: Broadly and Densely Labeled Dataset

To ascertain alignment with both low-level concepts such as colors and higher-level concepts such as objects, we assembled the *Broadly and Densely Labeled Dataset (Broden)* which unifies several densely labeled image datasets: ADE [32], OpenSurfaces [33], Pascal-Context [34], Pascal-Part [35], and the Describable Textures Dataset [36]. These datasets contain examples of a broad range of colors, materials, textures, parts, objects and scenes. Most examples are segmented down to the pixel level except textures and scenes, which cover full images. Every pixel is also annotated automatically with one of eleven color names commonly used by humans [37].

Broden provides a ground truth set of exemplars for a set of visual concepts (see examples in Fig. 1). The concept labels in Broden are merged from their original datasets so that every class corresponds to an English word. Labels are merged based on shared synonyms, disregarding positional distinctions such as 'left' and 'top' and avoiding a blacklist of 29 overly general synonyms (such as 'machine' for 'car'). Multiple Broden labels can apply to the same pixel. A pixel that has the Pascal-Part label 'left front cat leg' has three labels in Broden: a unified 'cat' label representing cats across datasets; a similar unified 'leg' label; and the color label 'black'. Only labels with at least 10 samples are included. Table 1 shows the number of classes per dataset and the average number of image samples per label class, for a total of 1197 classes.

2.2 Scoring Unit Interpretability

The proposed network dissection method evaluates every individual convolutional unit in a CNN as a solution to a

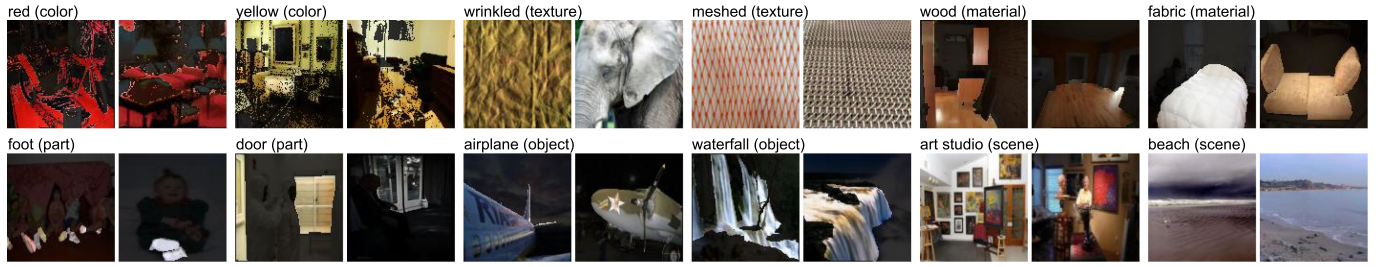


Fig. 1. Samples from the Broden Dataset. The ground truth for each concept is a pixel-wise dense annotation.

binary segmentation task to every visual concept in Broden (see Fig. 3). Our method can be applied to any CNN using a forward pass without the need for training or backpropagation. For every input image \mathbf{x} in the Broden dataset, the activation map $A_k(\mathbf{x})$ of every internal convolutional unit k is collected. Then the distribution of individual unit activations a_k is computed. For each unit k , the top quantile level T_k is determined such that $P(a_k > T_k) = 0.005$ over every spatial location of the activation map in the dataset.

To compare a low-resolution unit’s activation map to the input-resolution annotation mask L_c for some concept c , the activation map is scaled up to the mask resolution $S_k(\mathbf{x})$ from $A_k(\mathbf{x})$ using bilinear interpolation, anchoring interpolants at the center of each unit’s receptive field.

$S_k(\mathbf{x})$ is then thresholded into a binary segmentation: $M_k(\mathbf{x}) \equiv S_k(\mathbf{x}) \geq T_k$, selecting all regions for which the activation exceeds the threshold T_k . These segmentations are evaluated against every concept c in the dataset by computing intersections $M_k(\mathbf{x}) \cap L_c(\mathbf{x})$, for every (k, c) pair.

The score of each unit k as segmentation for concept c is reported as a the Intersection over Union score (IoU) across all the images in the dataset,

$$IoU_{k,c} = \frac{\sum |M_k(\mathbf{x}) \cap L_c(\mathbf{x})|}{\sum |M_k(\mathbf{x}) \cup L_c(\mathbf{x})|}, \quad (1)$$

where $|\cdot|$ is the cardinality of a set. Because the dataset contains some types of labels which are not present on some subsets of inputs, the sums are computed only on the subset of images that have at least one labeled concept of the same category as c . The value of $IoU_{k,c}$ is the accuracy of unit k in detecting concept c ; we consider one unit k as a *detector* for concept c if $IoU_{k,c}$ exceeds a threshold (> 0.04). Our qualitative results are insensitive to the IoU threshold: different thresholds denote different numbers of units as concept detectors across all the networks but relative orderings remain stable. Given that one unit might be the detector for multiple concepts, here we choose the top ranked label. To quantify the interpretability of a layer, we count the number of unique concepts aligned with units, i.e., *unique detectors*.

TABLE 1
Statistics of Each Label Type Included in the Dataset

Category	Classes	Sources	Avg sample
scene	468	ADE [32]	38
object	584	ADE [32], Pascal-Context [34]	491
part	234	ADE [32], Pascal-Part [35]	854
material	32	OpenSurfaces [33]	1,703
texture	47	DTD [36]	140
color	11	Generated	59,250

Fig. 2 summarizes the whole process of scoring unit interpretability: By segmenting the annotation mask using the receptive field of units for the top activated images, we compute the IoU for each concept. Importantly, the IoU which evaluates the quality of the segmentation of a unit is an objective confidence score for interpretability that is *comparable across networks*, enabling us to compare interpretability of different representations and so lays the basis for the experiments below. Note that network dissection results depends on the underlying vocabulary: if a unit matches a human-understandable concept that is absent from Broden, that unit will not score well for interpretability. Future versions of Broden will include a larger vocabulary of visual concepts.

3 EXPERIMENTS OF INTERPRETING DEEP VISUAL REPRESENTATIONS

In this section, we conduct a series of experiments to interpret the internal representations of deep visual representations. In Section 3.1, we validate our method using human evaluation. In Section 3.2 we use random unitary rotations of a learned representation to test whether interpretability of CNNs is an axis-independent property; we find that it is not, and we conclude that interpretability is not an inevitable result of the discriminative power of a representation. In Section 3.3 we analyze all the convolutional layers of AlexNet as trained on ImageNet [38] and Places [39]. We confirm that our method reveals detectors for higher-level semantic concepts at higher layers and lower-level concepts at lower layers; and that more detectors for higher-level concepts emerge under scene training. Then, we show that different network architectures such as AlexNet, VGG, and ResNet

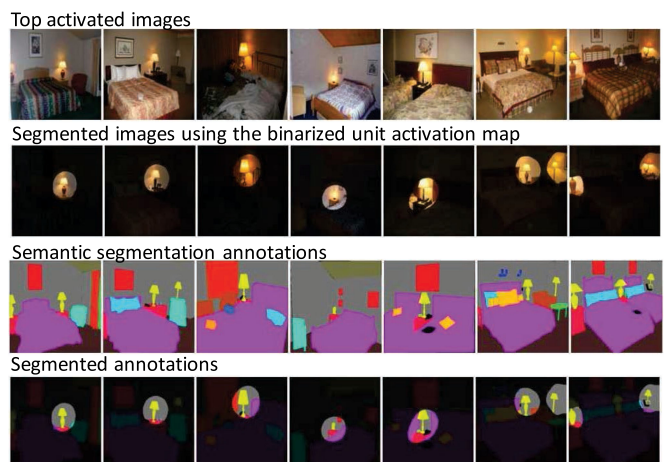


Fig. 2. Scoring unit interpretability by evaluating the unit for semantic segmentation.

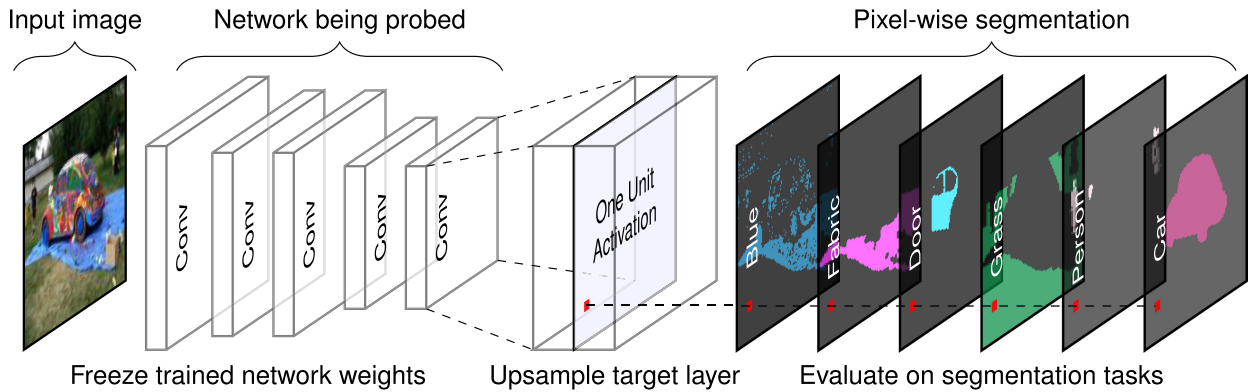


Fig. 3. Illustration of network dissection for measuring semantic alignment of units in a given CNN. Here one unit of the last convolutional layer of a given CNN is probed by evaluating its performance on various segmentation tasks. Our method can probe any convolutional layer.

yield different interpretability, and differently supervised training tasks and self-supervised training tasks also yield a variety of levels of interpretability in Section 3.4. Additionally in Section 3.5 we show the interpretability of model trained from captioning images. Another set of experiments shows the impact of different training conditions in Section 3.6 and what happens during the transfer learning in Section 3.7. We further examine the relationship between discriminative power and interpretability in Section 3.9, and investigate a possible way to improve the interpretability of CNNs by increasing their width in Section 3.8. Finally in Section 3.10, we utilize the interpretable units as explanatory factors to the prediction given by a CNN.

For testing we used CNN models with different architectures and primary tasks (Table 2), including AlexNet [38], GoogLeNet [40], VGG [41], ResNet [15], and DenseNet [42]. For supervised training, the models are trained from scratch (i.e., not pretrained) on ImageNet [43], Places205 [39], and Places365 [44]. ImageNet is an object-centric dataset, which contains 1.2 million images from 1000 object classes. Places205 (2.4 million images from 205 scene classes) and Places365 (1.6 million images from 365 scene classes) are two subsets the scene-centric dataset Places. “Hybrid” network refers to a combination of ImageNet and Places365. The self-supervised networks are introduced in Section 3.4.

3.1 Human Evaluation of Interpretations

Using network dissection, we analyzed the interpretability of units within all the convolutional layers of Places-

TABLE 2
Collection of Tested CNN Models

Training	Network	dataset or task
none	AlexNet	random
Supervised	AlexNet	ImageNet, Places205, Places365, Hybrid.
	GoogLeNet	ImageNet, Places205, Places365.
	VGG-16	ImageNet, Places205, Places365, Hybrid.
	ResNet-152	ImageNet, Places365.
	DenseNet-161	ImageNet, Places365.
Self	AlexNet	context, puzzle, egomotion, tracking, moving, videoorder, audio, crosschannel, colorization. objectcentric, transinv.

AlexNet and ImageNet-AlexNet, then compared with human interpretation. Places-AlexNet is trained for scene classification on Places205 [39], while ImageNet-AlexNet is the identical architecture trained for object classification on ImageNet [38].

Our evaluation was done by raters on Amazon Mechanical Turk (AMT). As a baseline, we used the descriptions from [1], where three independent raters wrote short phrases and gave a confidence score, to describe the meaning of a unit, based on seeing the top image patches for that unit. As a canonical description, we chose the most common description of a unit (when raters agreed), and the highest-confidence description (when raters did not agree). To identify non-interpretable units, raters were shown the canonical descriptions of visualizations and asked whether the description was valid. Units with validated descriptions are taken as interpretable units. To compare these baseline descriptions to network-dissection-derived labels, raters were shown a visualization of top images patches for an interpretable unit, along with a word or short phrase, and asked to vote (yes/no) whether the phrase was descriptive of most of the patches (Fig. 4). The baseline human-written descriptions were randomized with the labels from net

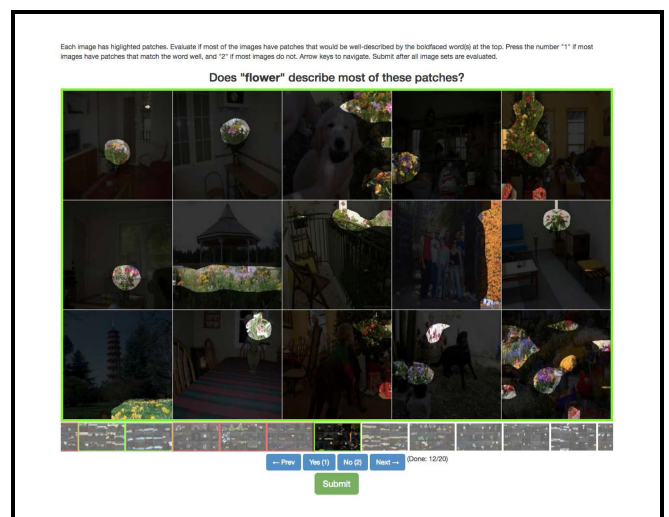


Fig. 4. The annotation interface used by human raters on Amazon Mechanical Turk. Raters are shown descriptive text in quotes together with fifteen images, each with highlighted patches, and must evaluate whether the quoted text is a good description for the highlighted patches.

TABLE 3
Human Evaluation of Our Network Dissection Approach

	conv1	conv2	conv3	conv4	conv5
Interpretable units	57/96	126/256	247/384	258/384	194/256
color units	36	45	44	19	12
texture units	19	53	64	72	23
material units	0	2	2	9	8
part units	0	0	13	17	16
object units	2	22	109	127	114
scene units	0	4	15	14	21
Human consistency	82%	76%	83%	82%	91%
on color units	92%	80%	82%	84%	100%
on texture units	68%	81%	83%	81%	96%
on material units	n/a	50%	100%	78%	100%
on part units	n/a	n/a	92%	94%	88%
on object units	50%	68%	84%	83%	90%
on scene units	n/a	25%	67%	71%	81%
Network Dissection	37%	56%	54%	59%	71%
on color units	44%	53%	55%	42%	67%
on texture units	26%	58%	42%	54%	39%
on material units	n/a	50%	50%	89%	75%
on part units	n/a	n/a	85%	71%	75%
on object units	0%	59%	57%	65%	75%
on scene units	n/a	50%	53%	29%	86%

dissection, and the origin of the labels was not revealed to the raters. Table 3 summarizes the results. The number of interpretable units is shown for each layer and type of description. As expected, color and texture concepts dominate in the lower layers conv1 and conv2 while part, object and scene detectors are more frequent at conv4 and conv5. Average positive votes for descriptions of interpretable units are shown, both for human-written labels and network-dissection-derived labels. Human labels are most highly consistent for units of conv5, suggesting that humans have no trouble identifying high-level visual concept detectors, while lower-level detectors, particularly textures, are more difficult to label. Similarly, labels given by network dissection are best at conv5 and for high-level concepts, and are found to be

less descriptive for lower layers and textures. In Fig. 5, a sample of units is shown together with both automatically inferred interpretations and manually assigned interpretations taken from [1]. The predicted labels match the human annotation well, though sometimes they capture a different description of a concept, like the ‘crosswalk’ predicted by the algorithm compared to ‘horizontal lines’ given by human for the third unit in conv4 of Places-AlexNet in Fig. 5.

3.2 Measurement of Axis-Aligned Interpretability

Two hypotheses can explain the emergence of interpretability in individual hidden layer units:

Hypothesis 1. Interpretability is a property of the representation as a whole, and individual interpretable units emerge because interpretability is a generic property of typical directions of representational space. Under this hypothesis, projecting to *any* direction would typically reveal an interpretable concept, and interpretations of single units in the natural basis would not be more meaningful than interpretations that can be found in any other direction.

Hypothesis 2. Interpretable alignments are unusual, and interpretable units emerge because learning converges to a special basis that aligns explanatory factors with individual units. In this model, the natural basis represents a meaningful decomposition learned by the network.

Hypothesis 1 is the default assumption: in the past it has been found [19] that with respect to interpretability “there is no distinction between individual high level units and random linear combinations of high level units.” Network dissection allows us to re-evaluate this hypothesis. Thus, we conduct an experiment to determine whether it is meaningful to assign an interpretable concept to an individual unit. We apply random changes in basis to a representation learned by AlexNet. Under hypothesis 1, the overall level of interpretability should not be affected by a change in basis, even as

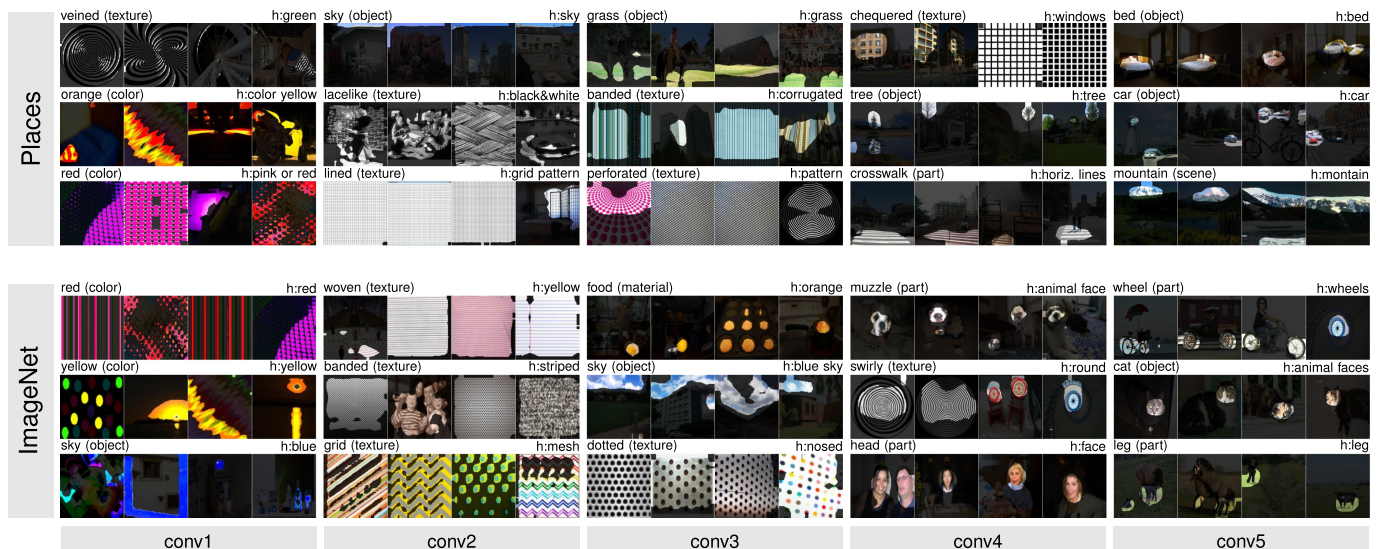


Fig. 5. Comparison of the interpretability of the convolutional layers of AlexNet, trained on classification tasks for Places (top) and ImageNet (bottom). Five units in each layer are shown with their semantics. The segmentation generated by each unit is shown on the three Broden images with highest activation. Top-scoring labels are shown above to the left, and human-annotated labels are shown above to the right. There is some disagreement: for example, raters mark the first conv4 unit on Places as a ‘windows’ detector, while the algorithm matches the ‘chequered’ texture.

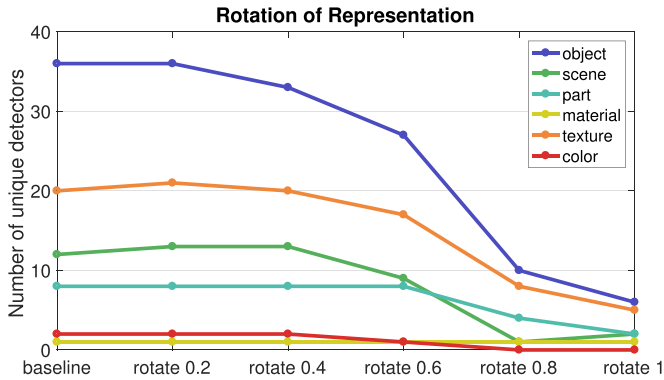


Fig. 6. Interpretability over changes in basis of the representation of AlexNet_{conv5} trained on Places. The vertical axis shows the number of unique interpretable concepts that match a unit in the representation. The horizontal axis shows α , which quantifies the degree of rotation.

rotations cause the specific set of represented concepts to change. Under hypothesis 2, the overall level of interpretability is expected to drop under a change in basis.

We begin with the representation of the 256 convolutional units of AlexNet_{conv5} trained on Places205 and examine the effect of a change in basis. To avoid any issues of conditioning or degeneracy, we change basis using a random orthogonal transformation Q . The rotation Q is drawn uniformly from $SO(256)$ by applying Gram-Schmidt on a normally-distributed $QR = A \in \mathbf{R}^{256^2}$ with positive-diagonal right-triangular R , as described by [45]. Interpretability is summarized as the number of unique visual concepts aligned with units, as defined in Section 2.2.

Denoting AlexNet_{conv5} as $f(x)$, we found that the number of unique detectors in $Qf(x)$ is 80 percent fewer than the number of unique detectors in $f(x)$. Our finding is inconsistent with hypothesis 1 and consistent with hypothesis 2.

We also tested smaller perturbations of basis using Q^α for $0 \leq \alpha \leq 1$, where the fractional powers $Q^\alpha \in SO(256)$ are chosen to form a minimal geodesic gradually rotating from I to Q ; these intermediate rotations are computed using a Schur decomposition. Fig. 6 shows that interpretability of

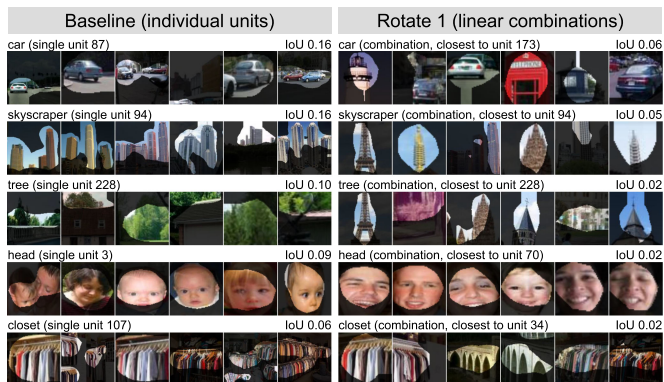


Fig. 7. Visualizations of the best single-unit concept detectors of five concepts taken from individual units of AlexNet_{conv5} trained on Places (left), compared with the best linear-combination detectors of the same concepts taken from the same representation under a random rotation (right). For most concepts, both the IoU and the visualization of the top activating image patches confirm that individual units match concepts better than linear combinations. In other cases, (e.g., head detectors) visualization of a linear combination appears highly consistent, but the IoU reveals lower consistency when evaluated over the whole dataset.

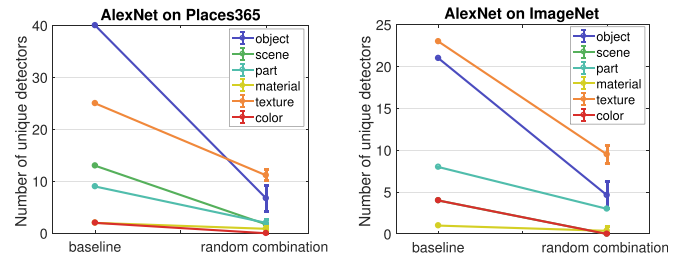


Fig. 8. Complete rotation ($\alpha = 1$) repeated on AlexNet trained on Places365 and ImageNet respectively. Rotation reduces the interpretability significantly for both of the networks.

$Q^\alpha f(x)$ decreases as larger rotations are applied. Fig. 7 shows some examples of the linearly combined units.

Each rotated representation has the same discriminative power as the original layer. Writing the original network as $g(f(x))$, note that $g(r) \equiv g(Q^T r)$ defines a neural network that processes the rotated representation $r = Qf(x)$ exactly as the original g operates on $f(x)$. Furthermore, we verify that a network can learn to solve a task given a rotated representation. Starting with AlexNet trained to solve places365, we freeze the bottom layers up to pool15 and retrain the top layers of the network under two conditions: one in which the representation at pool15 is randomly rotated ($\alpha = 1$) before passing to fc6, and the other where the representation up to pool15 is left unchanged. Then we reinitialize and retrain the fc6-fc8 layers of an AlexNet on places365. Under both the unrotated and rotated conditions, reinitializing and retraining the top layers improves performance, and the improvement is similar regardless of whether the pool15 representation is rotated. Initial accuracy is 50.3 percent. After retraining the unrotated representation, accuracy improves to 51.9 percent; after retraining the rotated representation, accuracy is 51.7 percent. Thus the network learns to solve the task even when the representation is randomly rotated. Since a network can be transformed into an equivalent network with the same discriminative ability but with lower interpretability, we conclude that interpretability must be measured separately from discrimination ability.

We repeated the measurement of interpretability upon complete rotation ($\alpha = 1$) on Places365 and ImageNet 10 times; see results in Fig. 8. There is a drop of interpretability for both. Alexnet on Places365 drops more, which can be explained due to that network starting with a higher number of interpretable units.

3.3 Network Architectures with Supervised Learning

How do different network architectures affect disentangled interpretability of the learned representations? For simplicity, the following experiments focus on the last convolutional layer of each CNN, where semantic detectors emerge most.

Results showing the number of unique detectors that emerge from various network architectures trained on ImageNet and Places, and the ratio of unique detectors (the number of unique detectors normalized by the total number of units at that layer) are plotted in Fig. 9. Interpretability in terms of the number of unique detectors, can be compared as follows: ResNet > DenseNet > VGG > GoogLeNet >

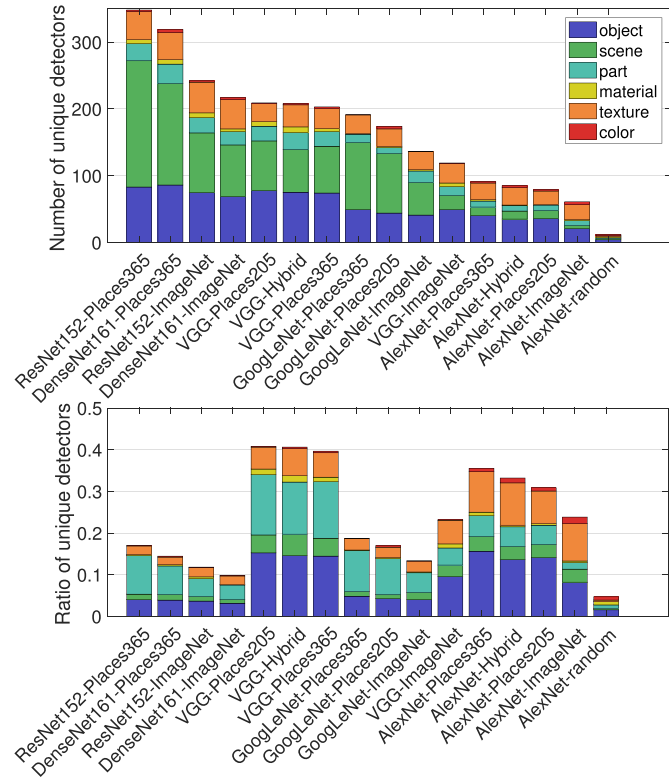


Fig. 9. Interpretability across different architectures trained on ImageNet and Places. Plot above shows the number of unique detectors, plot below shows the ratio of unique detectors (number of unique detectors divided by the total number of units).

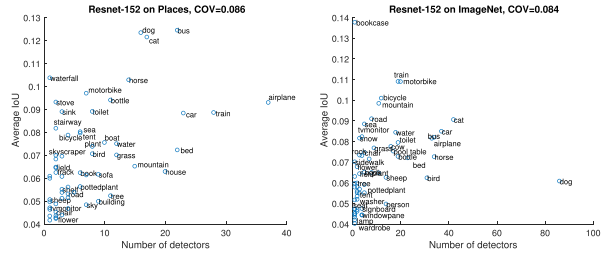


Fig. 10. Average IoU versus the number of detectors for the object class in Resnet152 trained on Places and ImageNet respectively. For a set of units detecting the same object class, we average their IoU.

AlexNet. Deeper architectures seem to have greater interpretability, though individual layer structure is different across architectures. Comparing training datasets, we find Places > ImageNet. As discussed in [1], scenes are composed of multiple objects, with more object detectors emerging in CNNs trained to recognize places. In terms of ratio of unique detectors, VGG architecture is highest. We consider the number of unique detectors as the metric of interpretability for a network as it better measures the diversity and coverage of emergent interpretable concepts.

Fig. 10 shows the plot of average IoU versus the number of detectors for the object detectors in Resnet152 trained on Places and ImageNet. Note the weak positive correlation between the two ($r = 0.08$), i.e., the higher average IoU the more detectors for that class.

Fig. 11 shows some object detectors grouped by object categories. For the same object category, the visual appearance

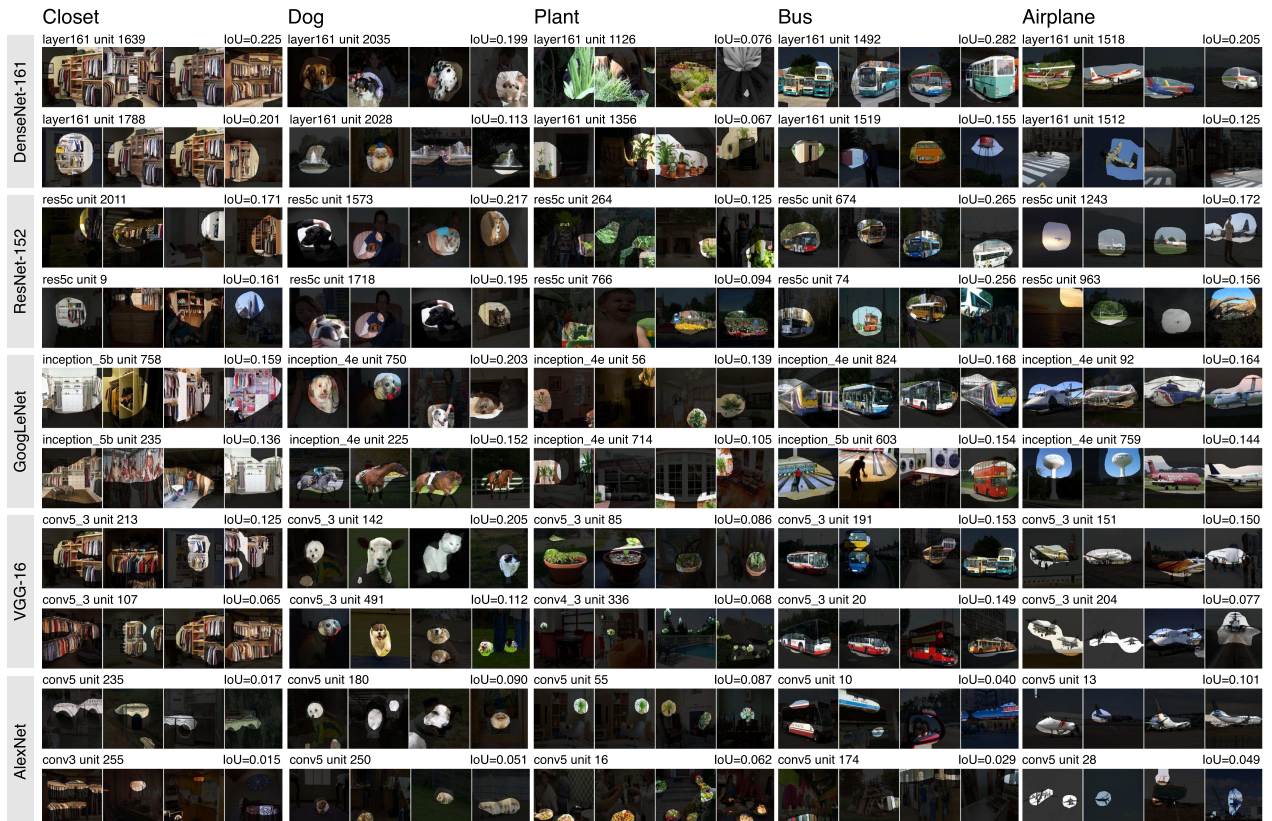


Fig. 11. Comparison of several visual concept detectors identified by network dissection in DenseNet, ResNet, GoogLeNet, VGG, and AlexNet. Each network is trained on Places365. The two highest-IoU matches among convolutional units of each network is shown. The segmentation generated by each unit is shown on the four maximally activating Broden images. Some units activate on concept generalizations, e.g., GoogLeNet 4e’s unit 225 on horses and dogs, and 759 on white ellipsoids and jets.

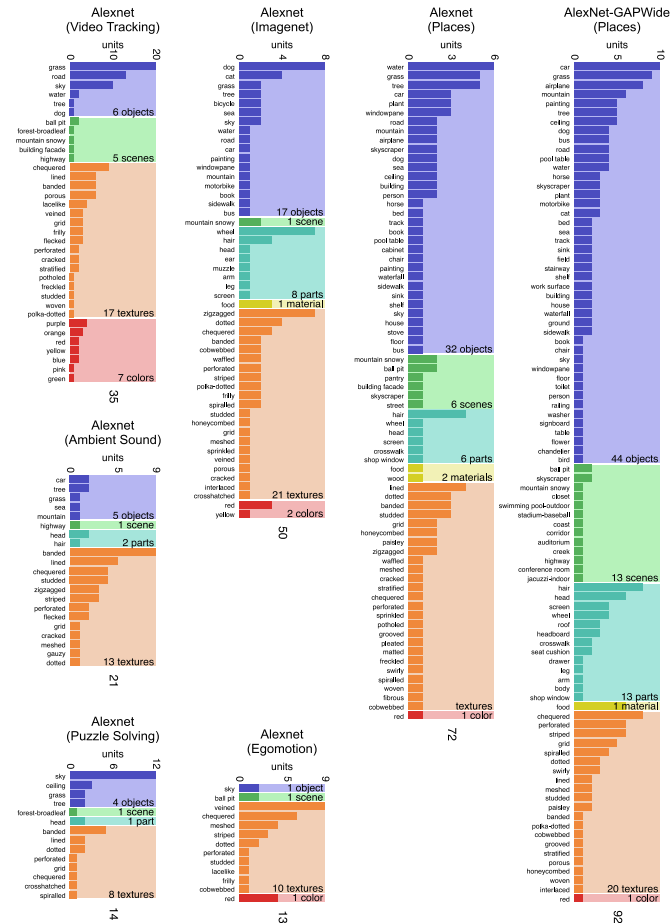


Fig. 12. Comparison of unique detectors of all types on a variety of training tasks. More results, including comparisons across architectures, are at the project page.

of the unit as detector varies within the same network and across different networks. DenseNet and ResNet have such good detectors for bus and airplane with $\text{IoU} > 0.25$. Fig. 12 compares interpretable units on a variety of training tasks.

Fig. 13 shows the interpretable detectors for different layers and network architectures trained on Places365. More object and scene detectors emerge at the higher layers across all architectures, suggesting that representational ability increases with layer depth.

Because of the compositional structure of the CNN layers, the deeper layers should have higher capacity to represent concepts with larger visual complexity such as objects and scene parts. Our measurements confirm this, and we conclude that higher network depth encourages the emergence of visual concepts with higher semantic complexity.

3.4 Representations from Self-Supervised Learning

Recently several works have explored a novel paradigm for unsupervised learning of CNNs without using millions of annotated images, namely self-supervised learning. Here, we investigated 12 networks trained for different self-supervised learning tasks: for predicting context (context) [23], solving puzzles (puzzle) [24], predicting ego-motion (egomotion) [25], learning by moving (moving) [26], predicting video frame order (videorder) [46] or tracking (tracking) [27], detecting object-centric alignment (objectcentric) [47], colorizing images (colorization) [28], inpainting (contextencoder) [48], predicting cross-channel (crosschannel) [29], predicting ambient sound from frames (audio) [30], and tracking invariant patterns in videos (transinv) [49]. The self-supervised models all used AlexNet or an AlexNet-derived architecture, with one exception model transinv which uses VGG as the base network.

How do different supervisions affect internal representations? We compared the interpretability resulting from self-supervised learning and supervised learning. We kept the network architecture to AlexNet for each model (one exception is the recent model transinv which uses VGG as the base network). Results are shown in Fig. 14: training on Places365 creates the largest number of unique detectors. Self-supervised models create many texture detectors but relatively few object detectors; apparently, supervision

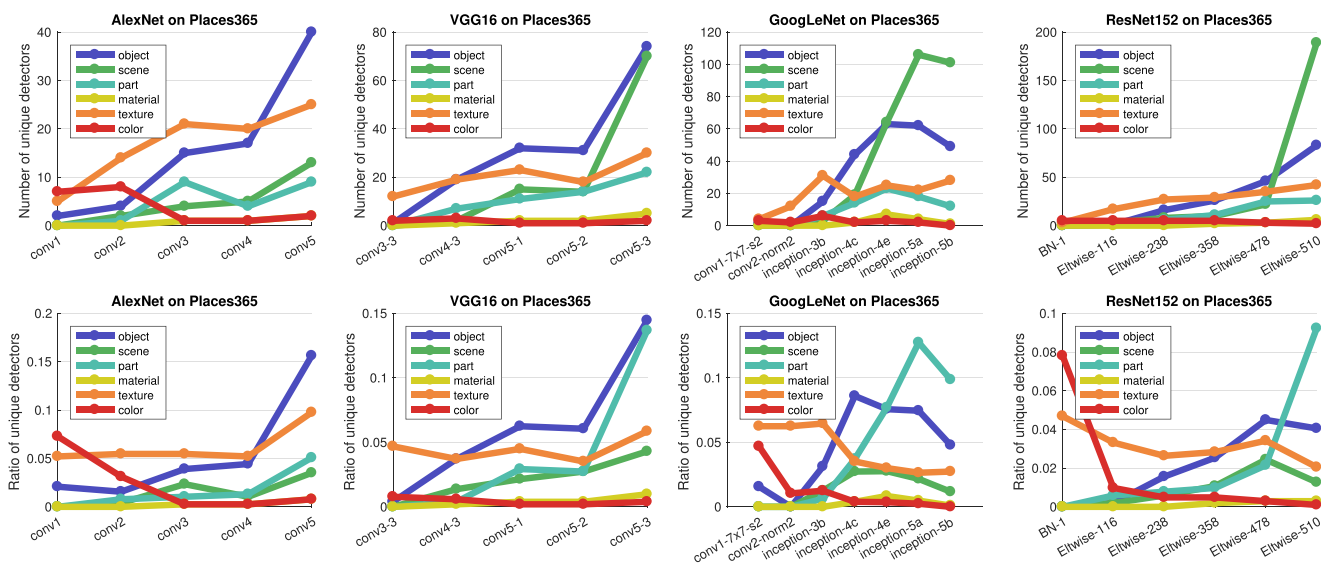


Fig. 13. Comparison of interpretability of the layers for AlexNet, VGG16, GoogLeNet, and ResNet152 trained on Places365. All five conv layers of AlexNet and the selected layers of VGG, GoogLeNet, and ResNet are included. Plot above shows the number of unique detectors and the plot below show the ratio of unique detectors.

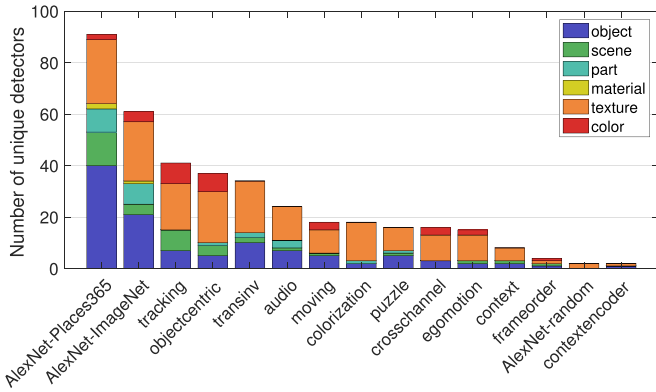


Fig. 14. Semantic detectors emerge across different supervision of the primary training task. All these models use the AlexNet architecture and are tested at conv5.

from a self-taught primary task is much weaker at inferring interpretable concepts than supervised training on a large annotated dataset. The form of self-supervision makes a difference: for example, the colorization model is trained on colorless images, and almost no color detection units emerge. This suggests that emergent units represent concepts required to solve a primary task.

Fig. 15 shows typical detectors identified in the self-supervised CNN models. For the models *audio* and *puzzle*, some part and object detectors emerge. Those detectors may be useful for CNNs to solve primary tasks: the *audio* model is trained to associate objects with a sound source, so it may be useful to recognize people and cars; while the *puzzle* model is trained to align the different parts of objects and scenes in an image. For *colorization* and *tracking*, recognizing textures might be good enough for the CNN to solve primary tasks such as colorizing a desaturated natural image; thus it is unsurprising that the texture detectors dominate.

3.5 Representations from Captioning Images

To further compare supervised learning and self-supervised learning, we trained a CNN from scratch using the supervision of captioning images, which generates natural language sentence to describe contents. We used the image captioning data from COCO dataset [50], with five captions per image. We then trained a CNN plus LSTM as the image captioning model similar to [51]. Features of ResNet18 are used as input to the LSTM for generating captions. The CNN+LSTM architecture and the network dissection results on the last convolutional layer of the ResNet18 are shown in Fig. 16: Many

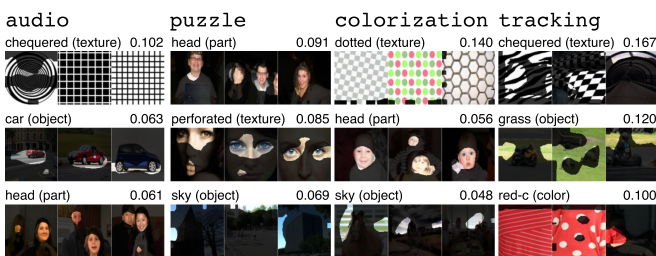


Fig. 15. The top ranked concepts in the three top categories in four self-supervised networks. Some object and part detectors emerge in audio. Detectors for person heads also appear in puzzle and colorization. A variety of texture concepts dominate models with self-supervised training.

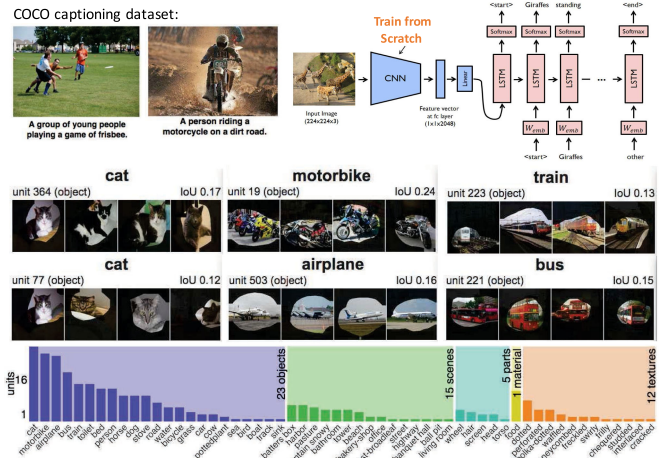


Fig. 16. Example images in the COCO captioning dataset, the CNN+LSTM image captioning model, and the network dissection result. Training ResNet18 from scratch using the supervision from captioning images leads to a lot of emergent object detectors.

object detectors emerge, suggesting that supervision from natural language captions contains high-level semantics.

3.6 Training Conditions

The number of training iterations, dropout [6], batch normalization [7], and random initialization [21], are known to affect the representation learned by neural networks. To analyze the effect of training conditions on interpretability, we took Places205-AlexNet as the baseline model and prepared several variants of it, all using the same AlexNet architecture. For the variants *Repeat1*, *Repeat2* and *Repeat3*, we randomly initialized the weights and trained them with the same number of iterations. For the variant *NoDropout*, we removed the dropout in the FC layers of the baseline model. For the variant *BatchNorm*, we applied batch normalization at each convolutional layer of the baseline model. *Repeat1*, *Repeat2*, *Repeat3* all have nearly the same top-1 accuracy 50.0 percent on the validation set. The variant without dropout has top-1 accuracy 49.2 percent. The variant with batch norm has top-1 accuracy 50.5 percent.

Fig. 17 shows the results: 1) Comparing different random initializations, the models converge to similar levels of interpretability, both in terms of unique detector number and total detector number; this matches observations of convergent learning discussed in [21]. 2) For the network without dropout, more texture detectors but fewer object detectors, emerge. 3) Batch normalization seems to decrease interpretability significantly.

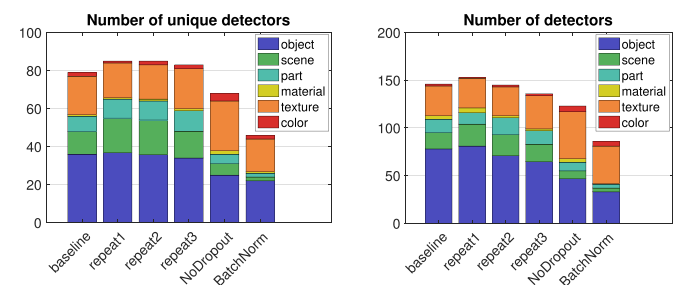


Fig. 17. Effect of regularizations on the interpretability of CNNs.

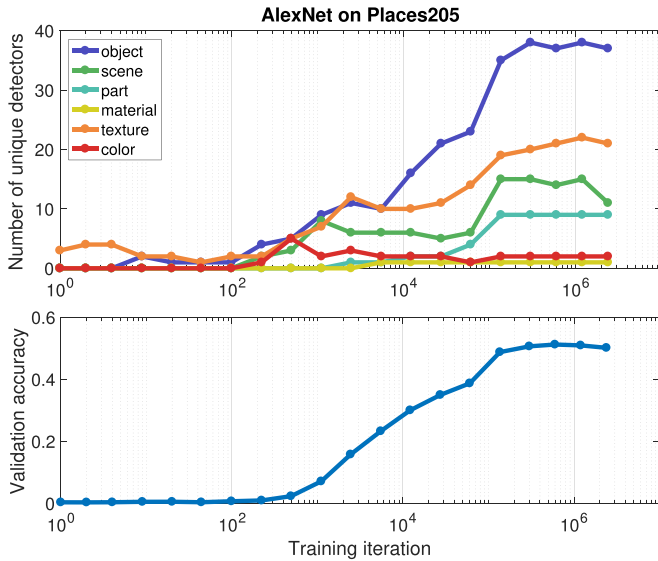


Fig. 18. The evolution of the interpretability of conv5 of Places205-AlexNet over 3,000,000 training iterations. The accuracy on the validation at each iteration is also plotted. The baseline model is trained to 300,000 iterations (marked at the red line).

The batch normalization result serves as a caution that discriminative power is not the only property of a representation that should be measured. Our intuition here is that the batch normalization ‘whitens’ the activation at each layer, which smooths out scaling issues and allows a network to easily rotate axes of intermediate representations during training. While whitening apparently speeds training, it may also have an effect similar to random rotations analyzed in Section 3.2 which destroy interpretability. As discussed in Section 3.2, however, interpretability is neither a prerequisite nor an obstacle to discriminative power. Finding ways to capture the benefits of batch normalization without destroying interpretability is an important area for future work.

Fig. 18 plots the interpretability of snapshots of the baseline model at different training iterations along with the accuracy on the validation set. We can see that object detectors and part detectors begin emerging at about 10,000 iterations (each iteration processes a batch of 256 images). We do not find evidence of transitions across different concept categories during training. For example, units in conv5 do not turn into texture or material detectors before becoming object or part detectors. In Fig. 19, we keep track of six units over different training iteration. We observe that some units start converging to the semantic concept at early stage. For example, unit138 starts detecting mountain snowy as early as at iteration 2446. We also observe that units evolve over

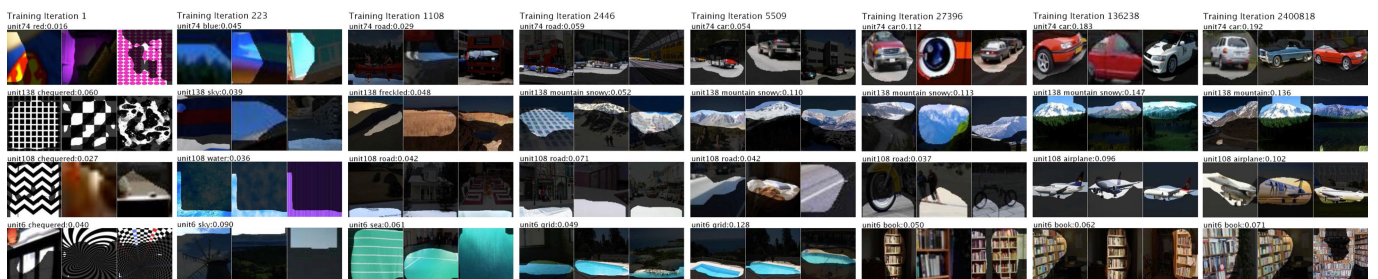


Fig. 19. The interpretations of units change over iterations. Each row shows the interpretation of one unit.

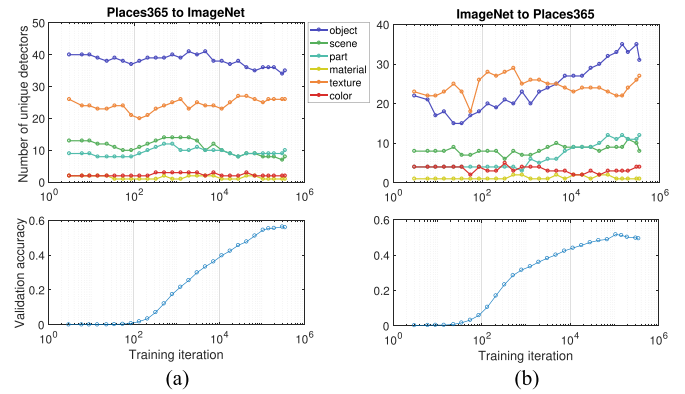


Fig. 20. a) Fine-tune AlexNet from ImageNet to Places365. b) Fine-tune AlexNet from Places365 to ImageNet.

time: unit74 and unit108 detect road first before they start detecting car and airplane respectively.

3.7 Transfer Learning between Places and ImageNet

Fine-tuning a pre-trained network to a target domain is commonly used in transfer learning. The deep features from the pre-trained network show good generalization across different domains. The pre-trained network also makes the training converge faster and results in better accuracy, especially if there is not enough training data for the target domain. Here we analyze how unit interpretation evolve during transfer learning.

To see how individual units evolve across domains, we run two experiments: fine-tuning Places-AlexNet to ImageNet and fine-tuning ImageNet-AlexNet to Places. The interpretability results of the model checkpoints at different fine-tuning iteration are plotted in Fig. 20. The training indeed converges faster compared to the network trained from scratch on Places in Fig. 18. The interpretations of the units also change over fine-tuning. For example, the number of unique object detectors first drop then keep increasing for the network trained on ImageNet being fine-tuned to Places365, while it is slowly dropping for the network trained on Places being fine-tuned to ImageNet.

Fig. 21 shows some examples of the individual unit evolution happening in the networks trained from ImageNet to Places365 and from Places365 to ImageNet, at the beginning and at the end of fine-tuning. In the ImageNet to Places365 network, unit15 which detects white dogs initially, evolves to detect waterfall; unit136 and unit144 which detect dogs first, evolve to detect horse and cow respectively (note a lot of scene categories in Places like pasture and corral contain

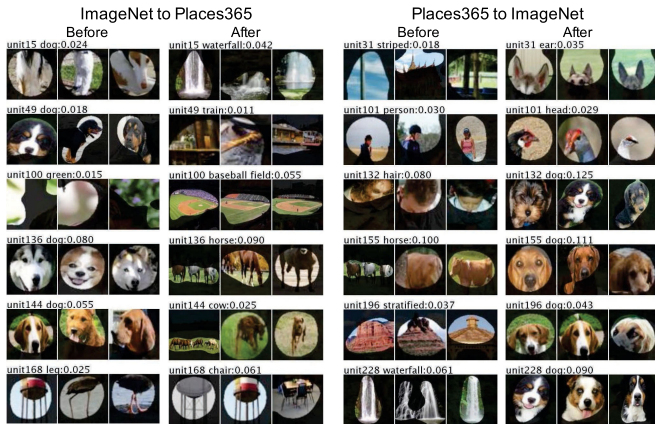


Fig. 21. Units evolve from a) the network fine-tuned from ImageNet to Places365 and b) the network fine-tuned from Places365 to ImageNet. Six units are shown with their semantics at the beginning of the fine-tuning and at the end of the fine-tuning.

these animals). In the Places365 to ImageNet network, several units evolve to be dog detectors, given ImageNet distribution of categories. While units evolve to detect different concepts, the before and after- concepts often share low-level image similarity such as colors and textures.

The fine-tuned model achieves almost the same classification accuracy as the train-from-scratch model, but the training converges faster due to the feature reuse. For the ImageNet to Places network, 139 out of 256 units (54.4 percent) at conv5 layer keep the same concepts during the finetuning, while for the network fine-tuned from Places to ImageNet, 135 out of 256 units (52.7 percent) at conv5 stay have the same concepts. We further categorized the unit evolution into five types based on the similarity between the concepts before and after fine-tuning. Out of the 117 units which evolved in the network fine-tuned from Imagenet to Places, 47 units keep a similar type of shape, 31 units have a similar texture, 18 units have similar colors, 13 units have a similar type of object, and 8 units do not have a clear pattern of similarities (see Fig. 22). Fig. 23 illustrates the evolution history for two units of each model. Units can switch their top ranked label several times before converging to a concept: unit15 in the fine-tuning of ImageNet to Places365 flipped to white, crystalline, before stabilizing to a waterfall concept. Other units converge earlier: unit132 in the fine-tuning of Places365 to ImageNet goes from hair to dog at an early stage of fine-tuning.

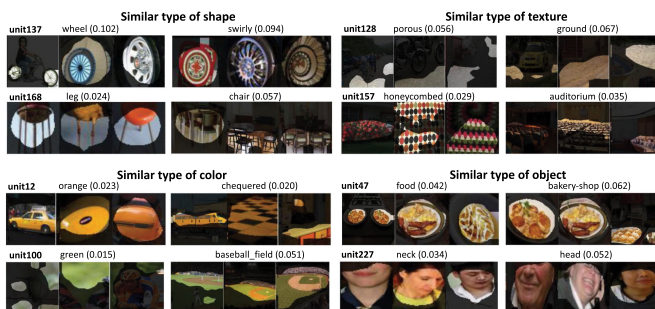


Fig. 22. Examples from four types of unit evolutions. Types are defined based on the concept similarity.

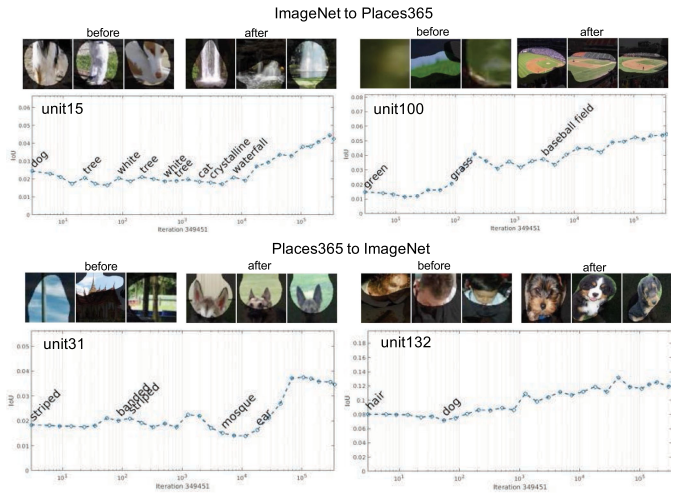


Fig. 23. The history of one unit evolution during the fine-tuning from ImageNet to Places365 (top) and Places365 to ImageNet (low).

3.8 Layer Width versus Interpretability

From AlexNet to ResNet, CNNs have grown deeper in the quest for higher classification accuracy. Depth is important for high discrimination ability, and as shown in Section 3.3, interpretability increases with depth. However, the role of the width of layers (the number of units per layer) has been less explored. One reason is that increasing the number of convolutional units in a layer significantly increases computational cost while yielding only marginal classification accuracy improvements. Nevertheless, some recent work [52] suggests that a carefully designed wide residual network can achieve classification accuracy superior to the commonly used thin and deep counterparts.

To test how width affects emergence of interpretable detectors, we removed the FC layers of AlexNet, then tripled the number of units at the conv5, i.e., from 256 to 768 units, as *AlexNet-GAP-Wide*. We further tripled the number of units for all the previous conv layers except conv1 for the standard AlexNet, as *AlexNet-GAP-WideAll*. Finally we put a global average pooling layer after conv5 and fully connected the pooled 768-feature activations to the final class prediction. After training on Places365, the *AlexNet-GAP-Wide* and the *AlexNet-GAP-WideAll* have similar classification accuracy on the validation set as the standard AlexNet (~ 0.5 percent top1 accuracy lower and higher): however many more emergent unique concept detectors at conv5 are found for *AlexNet-GAP-Wide* and all the conv layers for *AlexNet-GAP-WideAll* (see Fig. 24). Increasing the number of units to 1024 and 2048 at conv5, did not significantly increase the unique concepts. This may indicate either a limit on the capacity of AlexNet to separate explanatory factors, or a limit on the number of disentangled concepts that are helpful to solve the primary task of scene classification.

3.9 Discrimination versus Interpretability

Activations from the higher layers of pre-trained CNNs are often used as generic visual features (noted as deep features), generalizing well to other image datasets [16], [39]. It is interesting to bridge the notion of generic visual features with their interpretability. Here we first benchmarked the deep features from several networks on several image classification datasets for their discriminative power. For each

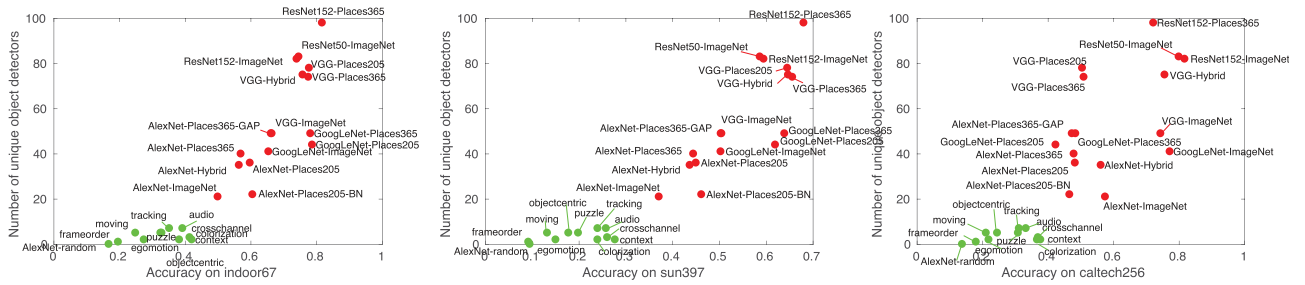


Fig. 26. The number of unique object detectors in the last convolutional layer compared to each representations classification accuracy on three datasets. Supervised (in red) and unsupervised (in green) representations clearly form two clusters.

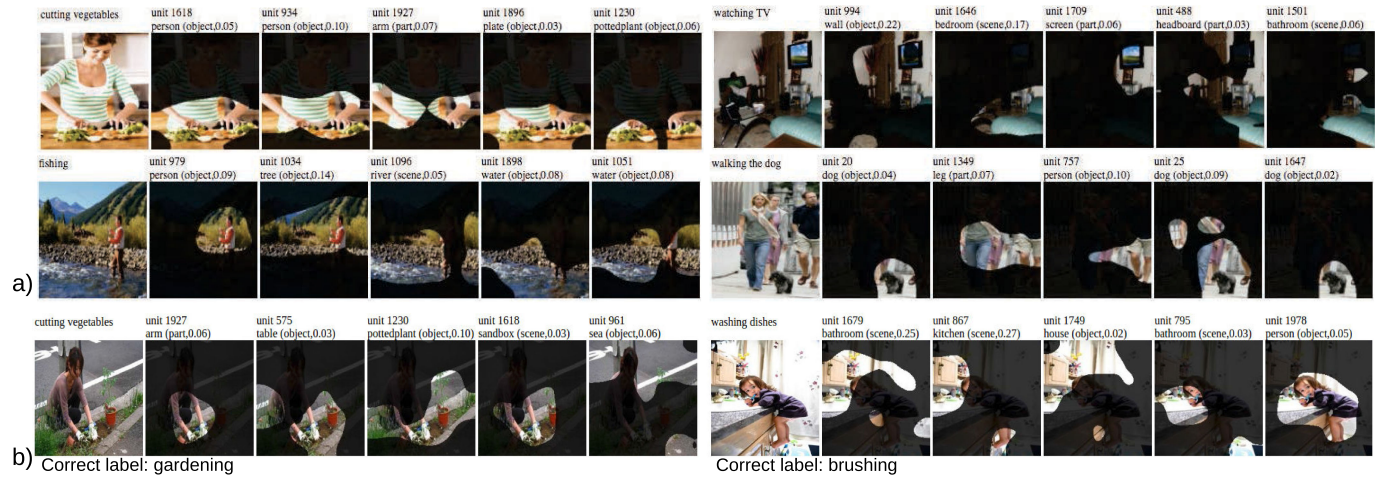


Fig. 27. Segmenting images using top activated units weighted by the class label from ResNet152-Places365 deep feature. a) the correctly predicted samples. b) the incorrectly predicted samples.

selectivity of individual units. The effect of choosing tighter and looser τ on the interpretation of units across a whole representation is shown in Fig. 28. A τ smaller than 0.005 identifies fewer objects because some objects will be missed by the small threshold. On the other hand, a larger τ , or using no threshold at all, associates units with general concepts such as colors, textures, and large regions, rather than capturing the sensitivity of units on more specific concepts. Fig. 29

shows the effect of varying τ on specific units' IoU. Although the two units are sensitive to paintings and horses, respectively, they are also both generally sensitive to the color brown when considered at a larger τ . The tight $\tau = 0.005$ reveals the sensitivity of the units to fine-grained concepts.

Potential Biases in the Interpretations. Several potential biases might occur to our method as follows: 1) Our method will not identify units that detect concepts that do not appear in the Broden dataset, including some difficult-to-name concepts such as 'the corner of a room'; 2) Some units might detect a very fine-grained concept, such as a

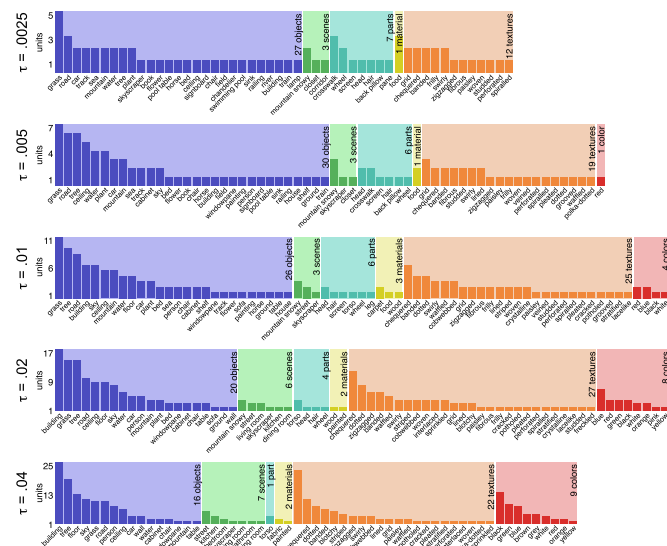


Fig. 28. Labels that appear in Alexnet-conv5 on Places205 as τ is varied from 0.0025 to 0.04. At wider thresholds, more units are assigned to labels for generic concepts such as colors and textures.

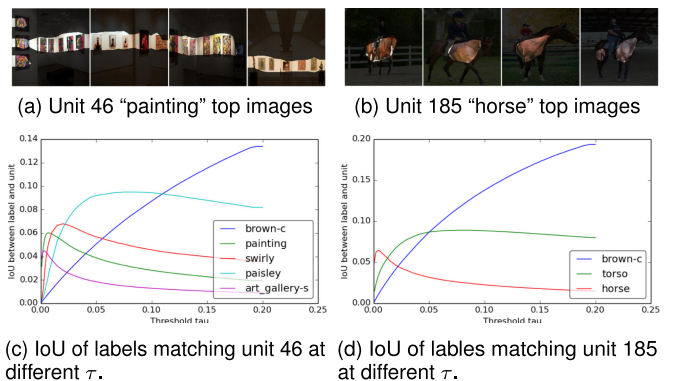


Fig. 29. Typical relationships between τ and IoU for different labels. In (c) and (d), IoU is shown on the y axis and τ is on the x axis, and every concept in Broden which maximizes IoU for some τ is shown. For loose thresholds, the same general concept "brown-c" maximizes IoU for both units even though the units have remarkable distinctive selectivity at tighter thresholds.

wooden stool chair leg, which are more specific than concepts in Broden, thus yielding a low IoU on the ‘chair’ category. Such units might not be counted as a concept detector. 3) Our method measures the degree of alignment between individual unit activations and a visual concept, so it will not identify a group of units that might jointly represent one concept; 4) Units might not be centered within their receptive fields so that the upsampled activation maps may be misaligned by a few pixels. 5) The “number of unique detectors” metric might favor large networks in comparing their network interpretability.

5 CONCLUSION

Network Dissection translates qualitative visualizations of representation units into quantitative interpretations and measurements of interpretability. Here we show that the units of a deep representation are significantly more interpretable than expected for a basis of the representation space. We investigate the interpretability of deep visual representations resulting from different architectures, training supervisions, and training conditions. We also show that interpretability of deep visual representations is relevant to the power of the representation as a generalizable visual feature. We conclude that interpretability is an important property of deep neural networks that provides new insights into their hierarchical structure. Our work motivates future work towards building more interpretable and explainable AI systems.

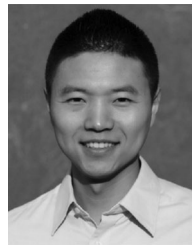
ACKNOWLEDGMENTS

This work was funded by DARPA XAI program No. FA8750-18-C-0004 and NSF Grant No. 1524817 to A.T., NSF grant No. 1532591 to A.O and A.T.; the Vannevar Bush Faculty Fellowship program funded by the ONR grant No. N00014-16-1-3116 to A.O.; the MIT Big Data Initiative at CSAIL, the Toyota Research Institute MIT CSAIL Joint Research Center, Google and Amazon Awards, and a hardware donation from NVIDIA Corporation. B.Z. is supported by a Facebook Fellowship. Bolei Zhou and David Bau contributed equally to this work.

REFERENCES

- [1] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Object detectors emerge in deep scene cnns,” in *Proc. Int. Conf. Learn. Representations*, 2015.
- [2] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari, “Do semantic parts emerge in convolutional neural networks?” *Int. J. Comput. Vis.*, vol. 126, no. 5, pp. 476–494, 2018.
- [3] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” *Advances Neural Inf. Process. Syst.*, vol. 23, pp. 613–621, 2016.
- [4] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.
- [5] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, “Network dissection: Quantifying interpretability of deep visual representations,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3310–3327.
- [6] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [7] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. IEEE Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
- [8] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 818–833.
- [9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Region-based convolutional networks for accurate object detection and segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.
- [10] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 5188–5196.
- [11] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” in *Proc. Int. Conf. Learn. Representations Workshop*, 2014.
- [12] A. Mahendran and A. Vedaldi, “Understanding deep image representations by inverting them,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015.
- [13] A. Dosovitskiy and T. Brox, “Generating images with perceptual similarity metrics based on deep networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 658–666.
- [14] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune, “Synthesizing the preferred inputs for neurons in neural networks via deep generator networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3395–3403.
- [15] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [16] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, “Cnn features off-the-shelf: an astounding baseline for recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 806–813.
- [17] P. Agrawal, R. Girshick, and J. Malik, “Analyzing the performance of multilayer neural networks for object recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 329–344.
- [18] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep neural networks?” in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.
- [19] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.
- [20] A. Nguyen, J. Yosinski, and J. Clune, “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 427–436.
- [21] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. E. Hopcroft, “Convergent learning: Do different neural networks learn the same representations?” in *FE@NIPS*, 2015, pp. 196–212.
- [22] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *Proc. Int. Conf. Learn. Representations*, 2017.
- [23] C. Doersch, A. Gupta, and A. A. Efros, “Unsupervised visual representation learning by context prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1191–1200.
- [24] M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 69–84.
- [25] D. Jayaraman and K. Grauman, “Learning image representations tied to ego-motion,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1413–1421.
- [26] P. Agrawal, J. Carreira, and J. Malik, “Learning to see by moving,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 37–45.
- [27] X. Wang and A. Gupta, “Unsupervised learning of visual representations using videos,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2794–2802.
- [28] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *Proc. Eur. Conf. Comput. Vis.*, 2016.
- [29] R. Zhang, P. Isola, and A. A. Efros, “Split-brain autoencoders: Unsupervised learning by cross-channel prediction,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.
- [30] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, “Ambient sound provides supervision for visual learning,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 801–816.
- [31] R. Q. Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried, “Invariant visual representation by single neurons in the human brain,” *Nature*, vol. 435, no. 7045, pp. 1102–1107, 2005.
- [32] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ade20k dataset,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5122–5130.

- [33] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *ACM Trans. Graph.*, vol. 33, 2014, Art. no. 159.
- [34] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 891–898.
- [35] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille, "Detect what you can: Detecting and representing objects using holistic models and body parts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1979–1986.
- [36] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3606–3613.
- [37] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus, "Learning color names for real-world applications," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1512–1523, Jul. 2009.
- [38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [39] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 487–495.
- [40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.
- [42] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2261–2269.
- [43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
- [44] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2017.
- [45] P. Diaconis, "What is a random matrix?" *Notices AMS*, vol. 52, no. 11, pp. 1348–1349, 2005.
- [46] I. Mikijšra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: Unsupervised learning using temporal order verification," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 527–544.
- [47] R. Gao, D. Jayaraman, and K. Grauman, "Object-centric representation learning from unlabeled videos," in *Asian Conf. Comput. Vis.*, Springer, Cham, 2016, pp. 248–263.
- [48] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2536–2544.
- [49] X. Wang, K. He, and A. Gupta, "Transitive invariance for self-supervised visual representation learning," in *Proc. of Int'l Conf. Comput. Vis. (ICCV)*, 2017, pp. 1338–1347.
- [50] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [51] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.
- [52] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. British Mach. Vis. Conf. (BMVC)*, BMVA Press, Sep. 2016, pp. 87.1–87.12.
- [53] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 413–420.
- [54] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.
- [55] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. 7694, 2007. [Online]. Available: <http://authors.library.caltech.edu/7694>
- [56] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2921–2929.
- [57] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. Guibas, and L. Fei-Fei, "Human action recognition by learning bases of action attributes and parts," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1331–1338.



Bolei Zhou received the BEng degree in biomedical engineering from Shanghai Jiao Tong University, in 2010 and the MPhil degree in information engineering from the Chinese University of Hong Kong. He is working toward the PhD degree from Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology. His research interests include computer vision and machine learning. He is an award recipient of Facebook Fellowship, Microsoft Research Asia Fellowship, and MIT Greater China Fellowship.



David Bau received the AB degree in mathematics from Harvard, in 1992 and the MS degree in computer science from Cornell, in 1994. He is working toward the PhD degree at the MIT Computer Science and Artificial Intelligence Laboratory (CSAIL). He coauthored a textbook on numerical linear algebra. He was a software engineer with Microsoft and Google and developed ranking algorithms for Google Image Search. His research interest includes interpretable machine learning.



Aude Oliva received baccalaureate in physics and mathematics and the MSc and PhD degrees in cognitive sciences from the Institut National Polytechnique of Grenoble, France. She is a principal research scientist with the MIT Computer Science and Artificial Intelligence Laboratory. She received the 2006 National Science Foundation Career award, the 2014 Guggenheim and the 2016 Vannevar Bush awards. Her research spans cognitive science, neuroscience and computer vision.



Antonio Torralba received the degree in telecommunications engineering from Telecom BCN, Spain, in 1994 and the PhD degree in signal, image, and speech processing from the Institut National Polytechnique de Grenoble, France, in 2000. From 2000 to 2005, he spent postdoctoral training at the Brain and Cognitive Science Department and the Computer Science and Artificial Intelligence Laboratory, MIT. He is now a professor of electrical engineering and computer science with the Massachusetts Institute of Technology (MIT). He is an associate editor of the *International Journal in Computer Vision*, and has served as program chair for the Computer Vision and Pattern Recognition conference, in 2015. He received the 2008 National Science Foundation (NSF) Career award, the best student paper award at the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2009, and the 2010 J. K. Aggarwal Prize from the International Association for Pattern Recognition (IAPR).

▷ For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.