Network Dissection: Quantifying Interpretability of Deep Visual Representations

David Bau*, Bolei Zhou*, Aditya Khosla, Aude Oliva, and Antonio Torralba
CSAIL, MIT
{davidbau, bzhou, khosla, oliva, torralba}@csail.mit.edu

Abstract

We propose a general framework called Network Dissection for quantifying the interpretability of latent representations of CNNs by evaluating the alignment between individual hidden units and a set of semantic concepts. Given any CNN model, the proposed method draws on a broad data set of visual concepts to score the semantics of hidden units at each intermediate convolutional layer. The units with semantics are given labels across a range of objects, parts, scenes, textures, materials, and colors. We use the proposed method to show that individual units are significantly more interpretable than random linear combinations of units, then we apply our method to compare the latent representations of various networks when trained to solve different supervised and self-supervised training tasks. We further analyze the effect of training iterations, compare networks trained with different initializations, examine the impact of network depth and width, and measure the effect of dropout and batch normalization on the interpretability of deep visual representations. We demonstrate that the proposed method can shed light on characteristics of CNN models and training methods that go beyond measurements of their discriminative power.

1. Introduction

Observations of hidden units in large deep neural networks have revealed that human-interpretable concepts sometimes emerge as individual latent variables within those networks: for example, object detector units emerge within networks trained to recognize places [40]; part detectors emerge in object classifiers [11]; and object detectors emerge in generative video networks [32] (Fig. 1). 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 learning disentangled representations spontaneously. While it is commonly understood that a network can learn an efficient encoding that makes economical use of hidden variables to distinguish its states, 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, and encouraging disentangled representations is a significant area of research [5]. If the internal representation of a deep network is partly disentangled, one possible path for understanding its mechanisms is to detect disentangled structure, and simply read out the separated factors.

Figure 1. Unit 13 in [40] (classifying places) detects table lamps. Unit 246 in [11] (classifying objects) detects bicycle wheels. A unit in [32] (self-supervised for generating videos) detects people.

However, this proposal raises questions which we address in this paper:

- What is a disentangled representation, and how can its factors be quantified and detected?
- Do interpretable hidden units reflect a special alignment of feature space, or are interpretations a chimera?
- What conditions in state-of-the-art training lead to representations with greater or lesser entanglement?

To examine these issues, we propose a general analytic framework, *network dissection*, for interpreting deep visual representations and quantifying their interpretability. Using Broden, a broadly and densely labeled data set, our framework identifies hidden units' semantics for any given CNN, then aligns them with human-interpretable concepts. We evaluate our method on various CNNs (AlexNet, VGG, GoogLeNet, ResNet) trained on object and scene recognition, and show that emergent 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 training data sets, training techniques like dropout [28] and batch normalization [13], and supervision by different primary tasks.

∗ indicates equal contribution
Source code and data available at http://netdissect.csail.mit.edu

1.1. Related Work

A growing number of techniques have been developed to understand the internal representations of convolutional neural networks through visualization. The behavior of a CNN can be visualized by sampling image patches that maximize activation of hidden units [37, 40], or by using variants of backpropagation to identify or generate salient image features [17, 26, 37]. The discriminative power of hidden layers of CNN features can also be understood by isolating portions of networks, transferring them or limiting them, and testing their capabilities on specialized problems [35, 24, 2]. Visualizations digest the mechanisms of a network down to images which themselves must be interpreted; this motivates our work which aims to match representations of CNNs with labeled interpretations directly and automatically.

Most relevant to our current work are explorations of the roles of individual units inside neural networks. In [40] human evaluation was used to determine that individual units behave as object detectors in a network that was trained to classify scenes. [20] automatically generated prototypical images for individual units by learning a feature inversion mapping; this contrasts with our approach of automatically assigning concept labels. Recently [3] suggested an approach to testing the intermediate layers by training simple linear probes, which analyzes the information dynamics among layers and its effect on the final prediction.

2. Network Dissection

How can we quantify the clarity of an idea? The notion of a disentangled representation rests on the human perception of what it means for a concept to be mixed up. Therefore when we quantify interpretability, we define it in terms of alignment with a set of human-interpretable concepts. Our measurement of interpretability for deep visual representations proceeds in three steps:

1. Identify a broad set of human-labeled visual concepts.
2. Gather hidden variables' response to known concepts.
3. Quantify alignment of hidden variable−concept pairs.

This three-step process of *network dissection* is reminiscent of the procedures used by neuroscientists to understand similar representation questions in biological neurons [23]. 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 exactly with one human-interpretable concept. Although we expect a network to learn partially nonlocal representations in interior layers [5], and past experience shows that an emergent concept will often align with a combination of a several hidden units [11, 2], our present 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 Sec. 3.2, it is possible for two representations of perfectly equivalent discriminative power to have very different levels of interpretability.

Figure 2. Samples from the **Broden** Dataset. The ground truth for each concept is a pixel-wise dense annotation.

To assess the interpretability of any given CNN, we draw concepts from a new broadly and densely labeled image data set that unifies labeled visual concepts from a heterogeneous collection of labeled data sources, described in Sec. 2.1. We then measure the alignment of each hidden unit of the CNN 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 layer as a whole, we count the number of distinct visual concepts that are aligned with a unit in the layer, as detailed in Sec. 2.2.

2.1. Broden: Broadly and Densely Labeled Dataset

To be able to ascertain alignment with both low-level concepts such as colors and higher-level concepts such as objects, we have assembled a new heterogeneous data set.

The Broadly and Densely Labeled Dataset (**Broden**) unifies several densely labeled image data sets: ADE [43], OpenSurfaces [4], Pascal-Context [19], Pascal-Part [6], and the Describable Textures Dataset [7]. These data sets contain examples of a broad range of objects, scenes, object parts, textures, and materials in a variety of contexts. Most examples are segmented down to the pixel level except textures and scenes which are given for full-images. In addition, every image pixel in the data set is annotated with one of the eleven common color names according to the human perceptions classified by van de Weijer [31]. A sample of the types of labels in the Broden dataset are shown in Fig. 2.

The purpose of Broden is to provide a ground truth set of exemplars for a broad set of visual concepts. The concept labels in Broden are normalized and merged from their original data sets 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

Table 1. Statistics of each label type included in the data set.

Category	Classes	Sources	Avg sample
scene	468	ADE [43]	38
object	584	ADE [43], Pascal-Context [19]	491
part	234	ADE [43], Pascal-Part [6]	854
material	32	OpenSurfaces [4]	1,703
texture	47	DTD [7]	140
color	11	Generated	59,250

avoiding a blacklist of 29 overly general synonyms (such as 'machine' for 'car'). Multiple Broden labels can apply to the same pixel: for example, a black pixel that has the Pascal-Part label 'left front cat leg' has three labels in Broden: a unified 'cat' label representing cats across data sets; a similar unified 'leg' label; and the color label 'black'. Only labels with at least 10 image samples are included. Table 1 shows the average number of image samples per label class.

2.2. Scoring Unit Interpretability

The proposed network dissection method evaluates every individual convolutional unit in a CNN as a solution to a binary segmentation task to every visual concept in Broden (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 data set.

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 data set 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 data-set-wide intersection over union score

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

where $|\cdot|$ is the cardinality of a set. Because the data set 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. 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. For our comparisons we report a detector if $IoU_{k,c} > 0.04$. Note that one unit might be the detector for multiple concepts; for the purpose of our analysis, we choose the top ranked label. To quantify the interpretability of a layer, we count the number unique concepts aligned with units. We call this the number of *unique detectors*.

Table 2. Tested CNNs Models

Training	**Network**	**Data set 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.
Self	AlexNet	`context`, `puzzle`, `egomotion`, `tracking`, `moving`, `videoorder`, `audio`, `crosschannel`,`colorization`. `objectcentric`.

The IoU evaluating the quality of the segmentation of a unit is an objective confidence score for interpretability that is *comparable across networks*. Thus this score enables us to compare interpretability of different representations and lays the basis for the experiments below. Note that network dissection works only as well as the underlying data set: if a unit matches a human-understandable concept that is absent in Broden, then it will not score well for interpretability. Future versions of Broden will be expanded to include more kinds of visual concepts.

3. Experiments

For testing we prepare a collection of CNN models with different network architectures and supervision of primary tasks, as listed in Table 2. The network architectures include AlexNet [15], GoogLeNet [29], VGG [27], and ResNet [12]. For supervised training, the models are trained from scratch (i.e., not pretrained) on ImageNet [25], Places205 [42], and Places365 [41]. ImageNet is an object-centric data set, which contains 1.2 million images from 1000 classes. Places205 and Places365 are two subsets of the Places Database, which is a scene-centric data set with categories such as kitchen, living room, and coast. Places205 contains 2.4 million images from 205 scene categories, while Places365 contains 1.6 million images from 365 scene categories. "Hybrid" refers to a combination of ImageNet and Places365. For self-supervised training tasks, we select several recent models trained on predicting context (`context`) [9], solving puzzles (`puzzle`) [21], predicting ego-motion (`egomotion`) [14], learning by moving (`moving`) [1], predicting video frame order (`videoorder`) [18] or tracking (`tracking`) [33], detecting object-centric alignment (`objectcentric`) [10], colorizing images (`colorization`) [38], predicting cross-channel (`crosschannel`) [39], and predicting ambient sound from frames (`audio`) [22]. The self-supervised models we analyze are comparable to each other in that they all use AlexNet

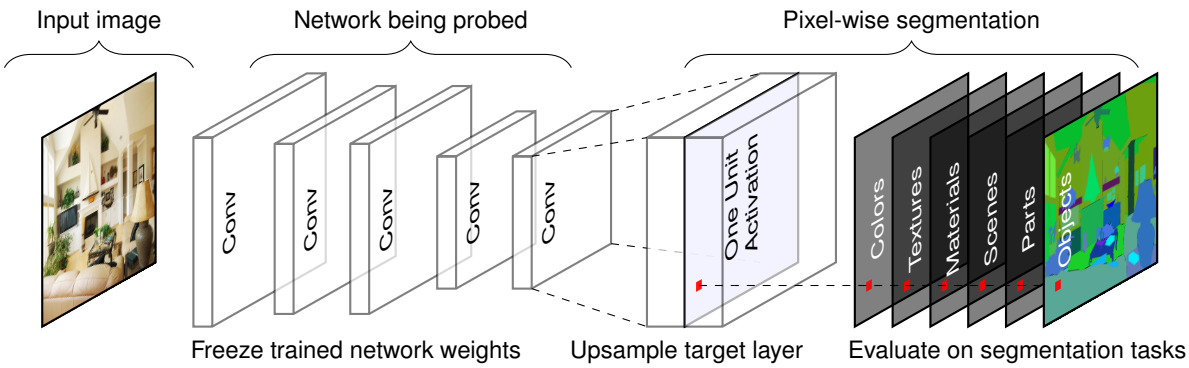

Figure 3. Illustration of network dissection for measuring semantic alignment of units in a given CNN. Here one unit of the last convolutional layer of a given CNN is probed by evaluating its performance on 1197 segmentation tasks. Our method can probe any convolutional layer.

or an AlexNet-derived architecture.

In the following experiments, we begin by validating our method using human evaluation. Then, 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. Next, we analyze all the convolutional layers of AlexNet as trained on ImageNet [15] and as trained on Places [42], and confirm that our method reveals detectors for higher-level 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 yield different interpretability, while differently supervised training tasks and self-supervised training tasks also yield a variety of levels of interpretability. Finally we show the impact of different training conditions, examine the relationship between discriminative power and interpretability, and investigate a possible way to improve the interpretability of CNNs by increasing their width.

3.1. Human Evaluation of Interpretations

We evaluate the quality of the unit interpretations found by our method using Amazon Mechanical Turk (AMT). Raters were shown 15 images with highlighted patches showing the most highly-activating regions for each unit in AlexNet trained on Places205, and asked to decide (yes/no) whether a given phrase describes most of the image patches.

Table 3 summarizes the results. First, we determined the set of interpretable units as those units for which raters agreed with ground-truth interpretations from [40]. Over this set of units, we report the portion of interpretations generated by our method that were rated as descriptive. Within this set we also compare to the portion of ground-truth labels that were found to be descriptive by a second group of raters. The proposed method can find semantic labels for units that are comparable to descriptions written by human annotators at the highest layer. At the lowest layer, the low-level color and texture concepts available in Broden are only sufficient to match good interpretations for a minority of units. Human consistency is also highest at `conv5`, which suggests that humans are better at recognizing and agreeing upon high-level visual concepts such as objects and parts, rather than the shapes and textures that emerge at lower layers.

Table 3. Human evaluation of our Network Dissection approach. Interpretable units are those where raters agreed with ground-truth interpretations. Within this set we report the portion of interpretations assigned by our method that were rated as descriptive. Human consistency is based on a second evaluation of ground-truth labels.

	conv1	conv2	conv3	conv4	conv5
Interpretable units	57/96	126/256	247/384	258/384	194/256
Human consistency	82%	76%	83%	82%	91%
Network Dissection	37%	56%	54%	59%	71%

3.2. Measurement of Axis-Aligned Interpretability

We conduct an experiment to determine whether it is meaningful to assign an interpretable concept to an individual unit. Two possible hypotheses can explain the emergence of interpretability in individual hidden layer units:

Hypothesis 1. Interpretable units emerge because interpretable concepts appear in most directions in representation space. If the representation localizes related concepts in an axis-independent way, projecting to *any* direction could reveal an interpretable concept, and interpretations of single units in the natural basis may not be a meaningful way to understand a representation.

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 [30] 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. We apply random changes in basis to a representation

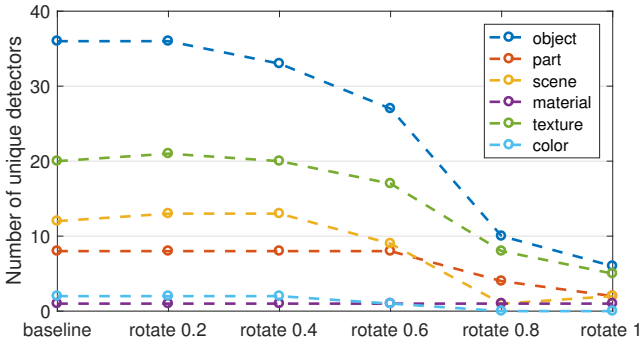

Figure 4. Interpretability over changes in basis of the representation of AlexNet conv5 trained on Places. The vertical axis shows the number of unique interpretable concepts that match a unit in the representation. The horizontal axis shows α, which quantifies the degree of rotation.

learned by AlexNet. Under hypothesis 1, the overall level of interpretability should not be affected by a change in basis, even as 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 [8]. Interpretability is summarized as the number of unique visual concepts aligned with units, as defined in Sec. 2.2.

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

We also test 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. 4 shows that interpretability of $Q^\alpha f(x)$ decreases as larger rotations are applied.

Each rotated representation has exactly 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 $f(x)$. We conclude that interpretability is neither an inevitable result of discriminative power, nor is it a prerequisite to discriminative power. Instead, we find that interpretability is a different quality that must be measured separately to be understood.

3.3. Disentangled Concepts by Layer

Using network dissection, we analyze and compare the interpretability of units within all the convolutional layers of Places-AlexNet and ImageNet-AlexNet. Places-AlexNet is trained for scene classification on Places205 [42], while ImageNet-AlexNet is the identical architecture trained for object classification on ImageNet [15].

The results are summarized in Fig. 5. A sample of units are shown together with both automatically inferred interpretations and manually assigned interpretations taken from [40]. We can see that the predicted labels match the human annotation well, though sometimes they capture a different description of a visual concept, such as the 'crosswalk' predicted by the algorithm compared to 'horizontal lines' given by the human for the third unit in conv4 of Places-AlexNet in Fig. 5. Confirming intuition, color and texture concepts dominate at lower layers conv1 and conv2 while more object and part detectors emerge in conv5.

3.4. Network Architectures and Supervisions

How do different network architectures and training supervisions affect disentangled interpretability of the learned representations? We apply network dissection to evaluate a range of network architectures and supervisions. 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 are plotted in Fig. 7, with examples shown in Fig. 6. In terms of network architecture, we find that interpretability of ResNet > VGG > GoogLeNet > AlexNet. Deeper architectures appear to allow greater interpretability. Comparing training data sets, we find Places > ImageNet. As discussed in [40], one scene is composed of multiple objects, so it may be beneficial for more object detectors to emerge in CNNs trained to recognize scenes.

Results from networks trained on various supervised and self-supervised tasks are shown in Fig. 8. Here the network architecture is AlexNet for each model, We observe that training on Places365 creates the largest number of unique detectors. Self-supervised models create many texture detectors but relatively few object detectors; apparently, supervision from a self-taught primary task is much weaker at inferring interpretable concepts than supervised training on a large annotated data set. 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. We hypothesize that emergent units represent concepts required to solve the primary task.

Fig. 9 shows some typical visual detectors identified in the self-supervised CNN models. For the models audio and puzzle, some object and part detectors emerge. Those detectors may be useful for CNNs to solve the primary tasks:

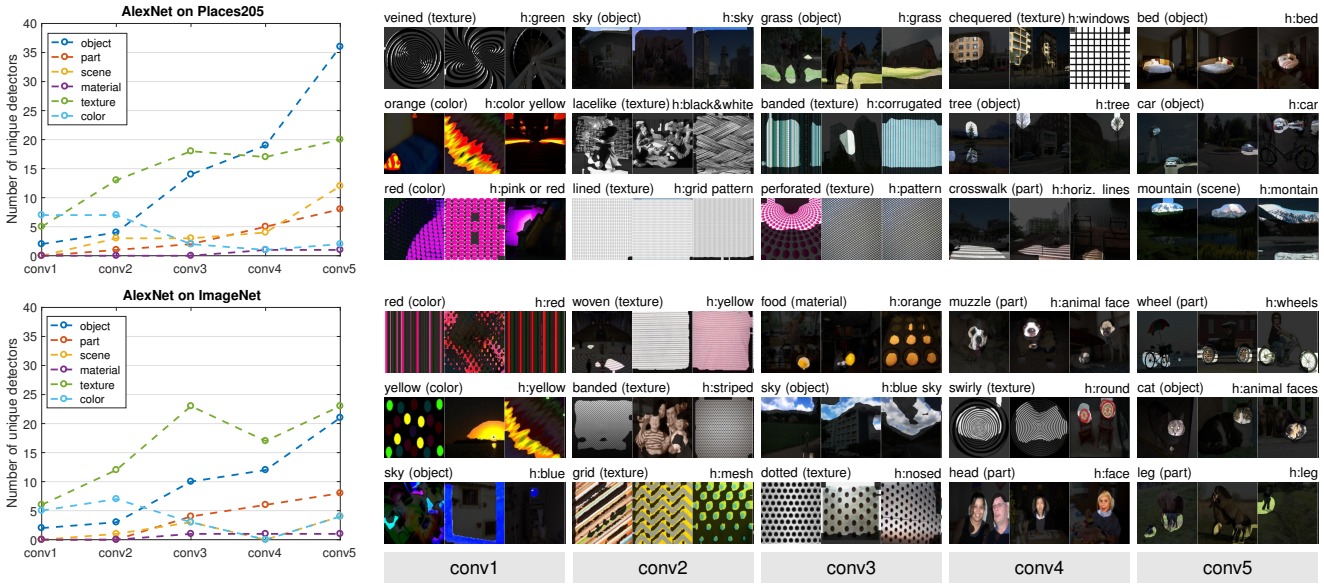

Figure 5. A comparison of the interpretability of all five convolutional layers of AlexNet, as trained on classification tasks for Places (top) and ImageNet (bottom). At right, three examples of units in each layer are shown with identified semantics. The segmentation generated by each unit is shown 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. Some disagreement can be seen for the dominant judgment of meaning. For example, human annotators mark the first conv4 unit on Places as a 'windows' detector, while the algorithm matches the 'chequered' texture.

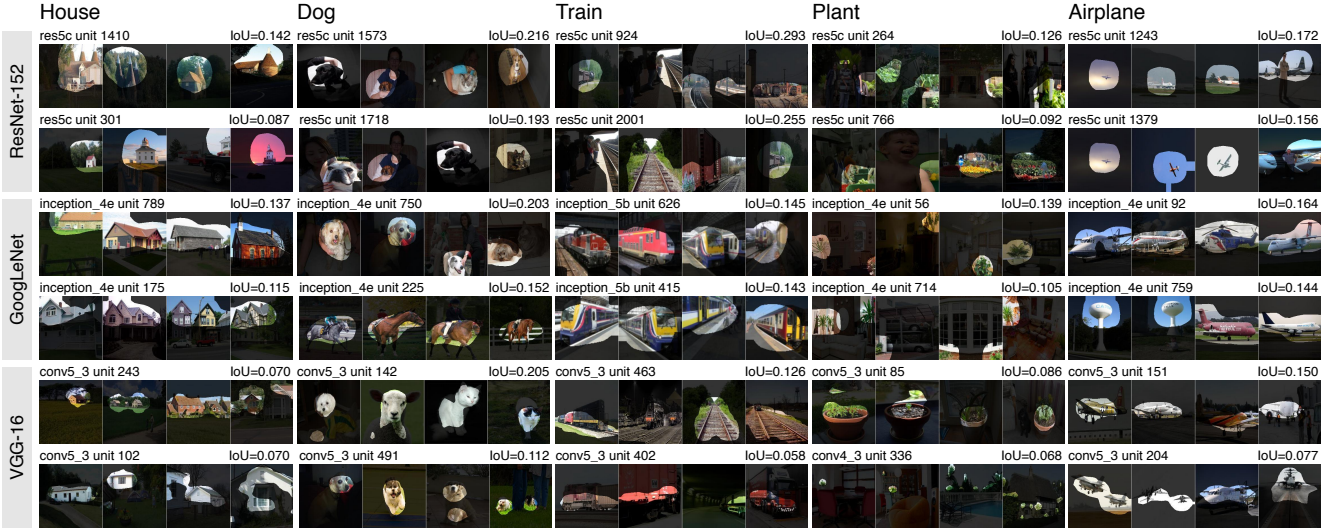

Figure 6. A comparison of several visual concept detectors identified by network dissection in ResNet, GoogLeNet, and VGG. Each network is trained on Places365. The two highest-IoU matches among convolutional units of each network is shown. The segmentation generated by each unit is shown on the four 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.

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. Training Conditions vs. Interpretability

Training conditions such as the number of training iterations, dropout [28], batch normalization [13], and random initialization [16], are known to affect the representation learning of neural networks. To analyze the effect of training conditions on interpretability, we take the Places205-AlexNet as the baseline model and prepare several variants of it, all using the same AlexNet architecture. For the vari-

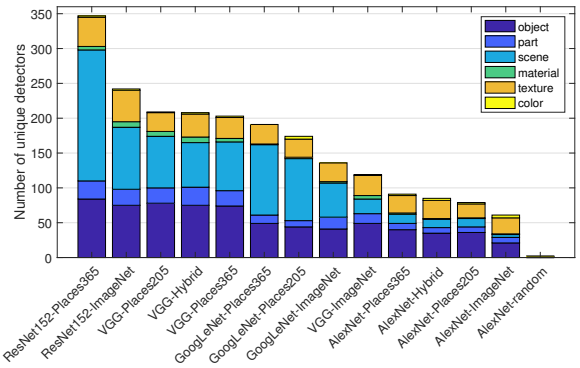

Figure 7. Interpretability across different architectures and training.

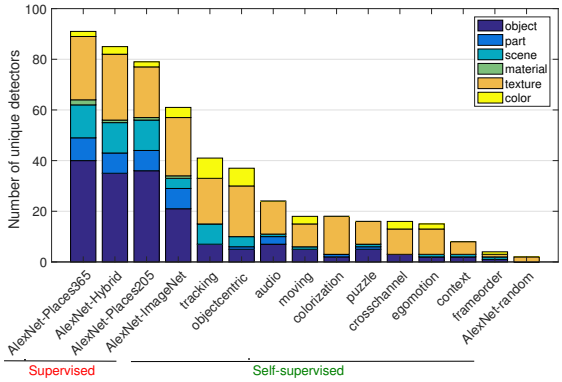

Figure 8. Semantic detectors emerge across different supervision of the primary training task. All these models use the AlexNet architecture and are tested at conv5.

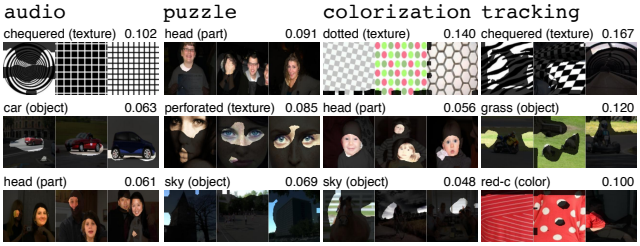

Figure 9. The top ranked concepts in the three top categories in four self-supervised networks. Some object and part detectors emerge in audio. Detectors for person heads also appear in puzzle and colorization. A variety of texture concepts dominate models with self-supervised training.

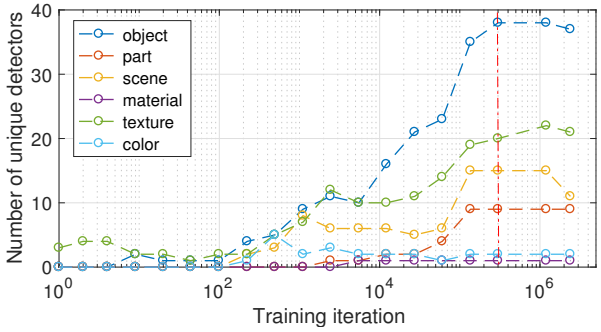

Figure 10. The evolution of the interpretability of conv5 of Places205-AlexNet over 2,400,000 training iterations. The baseline model is trained to 300,000 iterations (marked at the red line).

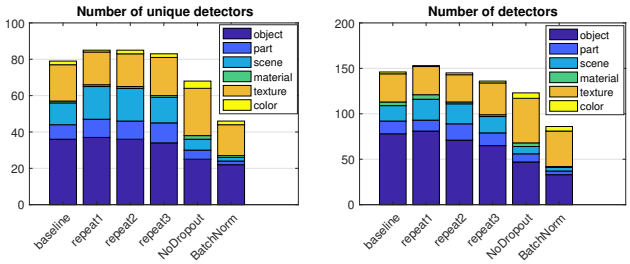

Figure 11. Effect of regularizations on the interpretability of CNNs.

ants *Repeat1*, *Repeat2* and *Repeat3*, we randomly initialize the weights and train them with the same number of iterations. For the variant *NoDropout*, we remove the dropout in the FC layers of the baseline model. For the variant *BatchNorm*, we apply batch normalization at each convolutional layers of the baseline model. Repeat1, Repeat2, Repeat3 all have nearly the same top-1 accuracy 50.0% on the validation set. The variant without dropout has top-1 accuracy 49.2%. The variant with batch norm has top-1 accuracy 50.5%.

In Fig. 10 we plot the interpretability of snapshots of the baseline model at different training iterations. 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.

Fig. 11 shows the interpretability of units in the CNNs over different training conditions. We find several effects: 1) Comparing different random initializations, the models converge to similar levels of interpretability, both in terms of the unique detector number and the total detector number; this matches observations of convergent learning discussed in [16]. 2) For the network without dropout, more texture detectors emerge but fewer object detectors. 3) Batch normalization seems to decrease interpretability significantly.

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 for the loss of interpretability under batch normalization 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 Sec. 3.2 which destroy interpretability. As discussed in Sec. 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.

3.6. Discrimination vs. Interpretability

Activations from the higher layers of CNNs are often used as generic visual features, showing great discrimination

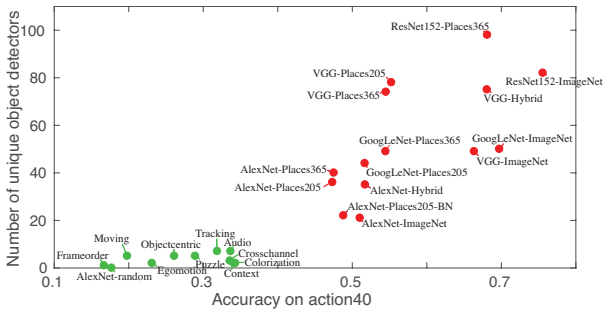

Figure 12. The number of unique object detectors in the last convolutional layer compared to each representations classification accuracy on the action40 data set. Supervised and unsupervised representations clearly form two clusters.

and generalization ability [42, 24]. Here we benchmark deep features from several networks trained on several standard image classification data sets for their discrimination ability on a new task. For each trained model, we extract the representation at the highest convolutional layer, and train a linear SVM with $C = 0.001$ on the training data for action40 action recognition task [34]. We compute the classification accuracy averaged across classes on the test split.

Fig. 12 plots the number of the unique object detectors for each representation, compared to that representation's classification accuracy on the action40 test set. We can see there is positive correlation between them. Thus the supervision tasks that encourage the emergence of more concept detectors may also improve the discrimination ability of deep features. Interestingly, the best discriminative representation for action40 is the representation from ResNet152-ImageNet, which has fewer unique object detectors compared to ResNet152-Places365. We hypothesize that the accuracy on a representation when applied to a task is dependent not only on the number of concept detectors in the representation, but on the suitability of the set of represented concepts to the transfer task.

3.7. Layer Width vs. Interpretability

From AlexNet to ResNet, CNNs for visual recognition have grown deeper in the quest for higher classification accuracy. Depth has been shown to be important to high discrimination ability, and we have seen in Sec. 3.4 that interpretability can increase with depth as well. However, the width of layers (the number of units per layer) has been less explored. One reason is that increasing the number of convolutional units at a layer significantly increases computational cost while yielding only marginal improvements in classification accuracy. Nevertheless, some recent work [36] shows that a carefully designed wide residual network can achieve classification accuracy superior to the commonly used thin and deep counterparts.

To explore how the width of layers affects interpretability of CNNs, we do a preliminary experiment to test how width

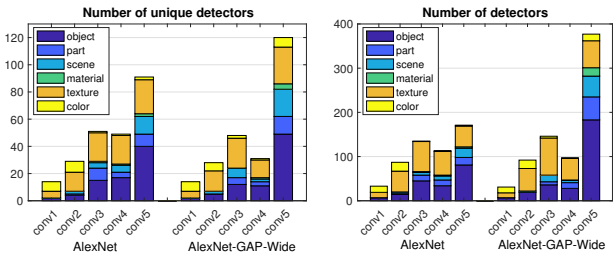

Figure 13. Comparison between standard AlexNet and AlexNet-GAP-Wide (AlexNet with wider `conv5` layer and GAP layer) through the number of unique detectors (the left plot) and the number of detectors (the right plot). Widening the layer brings the emergence of more detectors. Networks are trained on Places365.

affects emergence of interpretable detectors: we remove the FC layers of the AlexNet, then triple the number of units at the `conv5`, *i.e.*, from 256 units to 768 units. Finally we put a global average pooling layer after `conv5` and fully connect the pooled 768-feature activations to the final class prediction. We call this model *AlexNet-GAP-Wide*.

After training on Places365, the AlexNet-GAP-Wide obtains similar classification accuracy on the validation set as the standard AlexNet (0.5% top1 accuracy lower), but it has many more emergent concept detectors, both in terms of the number of unique detectors and the number of detector units at `conv5`, as shown in Fig. 13. We have also increased the number of units to 1024 and 2048 at `conv5`, but the number of unique concepts does not significantly increase further. This may indicate a limit on the capacity of AlexNet to separate explanatory factors; or it may indicate that a limit on the number of disentangled concepts that are helpful to solve the primary task of scene classification.

4. Conclusion

This paper proposed a general framework, network dissection, for quantifying interpretability of CNNs. We applied network dissection to measure whether interpretability is an axis-independent phenomenon, and we found that it is not. This is consistent with the hypothesis that interpretable units indicate a partially disentangled representation. We applied network dissection to investigate the effects on interpretability of state-of-the art CNN training techniques. We have confirmed that representations at different layers disentangle different categories of meaning; and that different training techniques can have a significant effect on the interpretability of the representation learned by hidden units.

Acknowledgements. This work was partly supported by the National Science Foundation under Grant No. 1524817 to A.T.; the Vannevar Bush Faculty Fellowship program sponsored by the Basic Research Office of the Assistant Secretary of Defense for Research and Engineering and funded by the Office of Naval Research through grant 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.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *Proc. ICCV*, 2015.

[2] P. Agrawal, R. Girshick, and J. Malik. Analyzing the performance of multilayer neural networks for object recognition. *Proc. ECCV*, 2014.

[3] G. Alain and Y. Bengio. Understanding intermediate layers using linear classifier probes. *arXiv:1610.01644*, 2016.

[4] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 2014.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[6] 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. CVPR*, 2014.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proc. CVPR*, 2014.

[8] P. Diaconis. What is a random matrix? *Notices of the AMS*, 52(11):1348–1349, 2005.

[9] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. CVPR*, 2015.

[10] R. Gao, D. Jayaraman, and K. Grauman. Object-centric representation learning from unlabeled videos. *arXiv:1612.00500*, 2016.

[11] A. Gonzalez-Garcia, D. Modolo, and V. Ferrari. Do semantic parts emerge in convolutional neural networks? *arXiv:1607.03738*, 2016.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016.

[13] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.

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

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. Hopcroft. Convergent learning: Do different neural networks learn the same representations? *arXiv:1511.07543*, 2015.

[17] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. *Proc. CVPR*, 2015.

[18] I. Mikjjsra, C. L. Zitnick, and M. Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Proc. ECCV*, 2016.

[19] 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. CVPR*, 2014.

[20] 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 Advances in Neural Information Processing Systems*, 2016.

[21] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. ECCV*, 2016.

[22] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *Proc. ECCV*, 2016.

[23] R. Q. Quiroga, L. Reddy, G. Kreiman, C. Koch, and I. Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107, 2005.

[24] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv:1403.6382*, 2014.

[25] 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'l Journal of Computer Vision*, 2015.

[26] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations Workshop*, 2014.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

[28] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[29] 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. CVPR*, 2015.

[30] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013.

[31] J. Van De Weijer, C. Schmid, J. Verbeek, and D. Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

[32] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. *arXiv:1609.02612*, 2016.

[33] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *Proc. CVPR*, 2015.

[34] 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. ICCV*, 2011.

[35] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? *In Advances in Neural Information Processing Systems*, 2014.

[36] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv:1605.07146*, 2016.

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

[38] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *Proc. ECCV*. Springer, 2016.

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

[40] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *International Conference on Learning Representations*, 2015.

[41] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An image database for deep scene understanding. *arXiv:1610.02055*, 2016.

[42] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. *In Advances in Neural Information Processing Systems*, 2014.

[43] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. *Proc. CVPR*, 2017.