Quantifying Interpretations of Deep Visual Representations

Bolei Zhou [*1] **David Bau** [*1] **Antonio Torralba** [1]

Abstract

We propose Network Dissection[1], a framework for quantifying interpretability of the units inside a deep convolutional neural networks (CNNs). Different vocabularies of interpretable units as object detectors have emerged from the networks trained for object recognition on ImageNet and scene classification on Places respectively. We reveal that the interpretations of units evolve over the training iterations both in the stage of the train-from-scratch and the stage of the fine-tuning between data sources. The interpretations of units is further used as explanatory factors for the deep features used for visual recognition.

1. Introduction

Previous efforts to interpret the internals of a convolutional neural network have focused on visualizations, for example, visualizing image patches that maximize individual unit activations (Zeiler & Fergus, 2014; Zhou et al., 2015); or using optimization to generate patterns and regions salient to a unit (Mahendran & Vedaldi, 2015; Simonyan et al., 2014; Zeiler & Fergus, 2014; Nguyen et al., 2016); or rendering representation space using dimensionality reduction (Maaten & Hinton, 2008; Jolliffe, 2002). Though the visualizations give us the intuition about what image patterns the units are supposed to detect, the results are rather qualitative and cannot be interpreted quantitatively, *i.e.* which concept label the detected image patterns belong to and how accurate the unit detects that concept. Therefore they leave open the question of how to quantify and compare interpretations of the deep visual representations.

Recently we propose a framework called Network Dissection to quantify the interpretability of any given CNNs (Bau et al., 2017). Our work quantifies interpretability by defining a benchmark for the emergence of detectors for interpretable visual concepts. Quantifying interpretability allows us to ask and answer whether interpretability is a property of the embedding or the overall representation; and whether and how different network architectures, training supervisions, and training regularization affect the interpretability of learned representations.

2. Overview of Network Dissection

To measure interpretability, we evaluate the ability of each hidden unit to solve segmentation problems from a dictionary of human-interpretable visual concepts.

2.1. Broden: Broadly and Densely Labeled Dataset

As a dictionary of visual concepts, we construct the Broadly and Densely Labeled Dataset (**Broden**), which unifies several densely labeled image data sets: ADE (Zhou et al., 2017), OpenSurfaces (Bell et al., 2014), Pascal-Context (Mottaghi et al., 2014), Pascal-Part (Chen et al., 2014), and the Describable Textures Dataset (Cimpoi et al., 2014), containing a broad range of labeled classes of objects, scenes, object parts, textures, and materials, with most examples labeled at the pixel level.

2.2. Scoring Unit Interpretability

Let c denote any concept within the Broden dataset and let k denote any convolutional unit in a CNN. Network dissection defines the quality of the interpretation c for unit k by quantifying the ability of k to solve the segmentation problem given by c using this IoU score:

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

In the above, $\mathbf{x}$ represents an image in the Broden dataset, $L_c(\mathbf{x})$ is the set of pixels labeled with concept c, and $M_k(\mathbf{x})$ is binary mask selecting those pixels that lie within areas of highest activation of unit k. M_k is computed by (bilinearly) upsampling the activation of k on input $\mathbf{x}$, and applying a threshold T_k that selects a fixed quantile (0.5%) of the pixels over the entire dataset. Because the data set contains some categories of labels (such as textures) which are not present on some subsets of inputs, the sums are computed

[*]Equal contribution [1]CSAIL, MIT. Correspondence to: Bolei Zhou <bolei@mit.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

[1]The complete paper and code are available at `http://netdissect.csail.mit.edu`

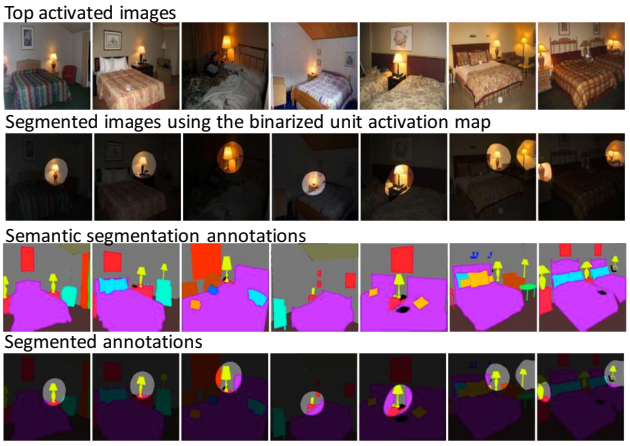

Figure 1. Scoring unit interpretability by evaluating the unit activation for semantic segmentation. Unit activation map is used to segment the top activated images, localizing the favorite image patterns for that unit. The activation map is further used to segment the annotation mask to compute the IoU.

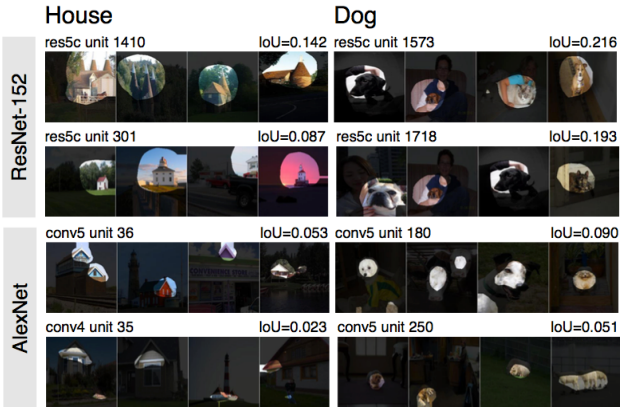

Figure 2. For each of the two concepts House and Dog, two hidden units from the ResNet and AlexNet are shown as object detectors respectively. The IoU scores are shown above the visualization.

only on the subset of images that have at least one labeled concept of the same category as c. Figure 1 gives one example of computing the IoU over the top activated images with semantic segmentation annotations.

The value of $IoU_{k,c}$ is the accuracy of unit k in detecting concept c. In our analysis, we consider a unit k as a detector for concept c if $IoU_{k,c} > 0.04$, and when a unit detects more than one concept, we choose the top scoring label. To quantify the interpretability of a layer, we count the distinct concepts detected, i.e., the number of *unique detectors*.

Network dissection is applied to the last convolutional layer of the testing networks. Figure 3 shows the histogram of units identified as object detectors from the AlexNet and ResNet trained on ImageNet and Places respectively. Each class might have several detectors. For example, for the networks trained on ImageNet, the most frequent detectrors are dog detectors. For the networks trained on Places, the most frequent detector in AlexNet is water detector, while the most frequent detector in ResNet is airplane detector. Comparing the networks trained on ImageNet and Places, we can see those networks learn quite different set of object vocabularies. If we keep the network architecture the same, there are more object detectors emerged in the network trained for scene classification (Places). Figure 2 shows some exemplar detectors from the two networks trained on Places and ImageNet.

3. Applications

3.1. The emergence of concepts over training iterations

Figure 4 plots the interpretability of snapshots of the baseline model (AlexNet trained on Places205) 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. From the plot with the validation accuracy,

In Figure 5, we keep track of two units over different training iteration. We observe that the units start converging to the semantic concept at early stage. For example the first unit detects road first before they start detecting car.

3.2. Transfer learning between Places and ImageNet

Fine-tuning the pre-trained network such as ImageNet-CNN to another target source is a transfer learning technique commonly used. It makes the training converge faster and results better accuracy especially if there is not enough training data at the target source. Here we would like to see how the interpretation of the internal units evolve during different stages of the transfer learning.

Given well trained Places-AlexNet and ImageNet-AlexNet, we fine-tune the Places-AlexNet on ImageNet and fine-tune the ImageNet-AlexNet on Places respectively. The interpretability results of the snapshots of the networks over the fine-tuning iterations are plotted in Figure 6. We can see that the training indeed converges faster compared to the network trained from scratch on Places in Figure 4. The semantics of 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.

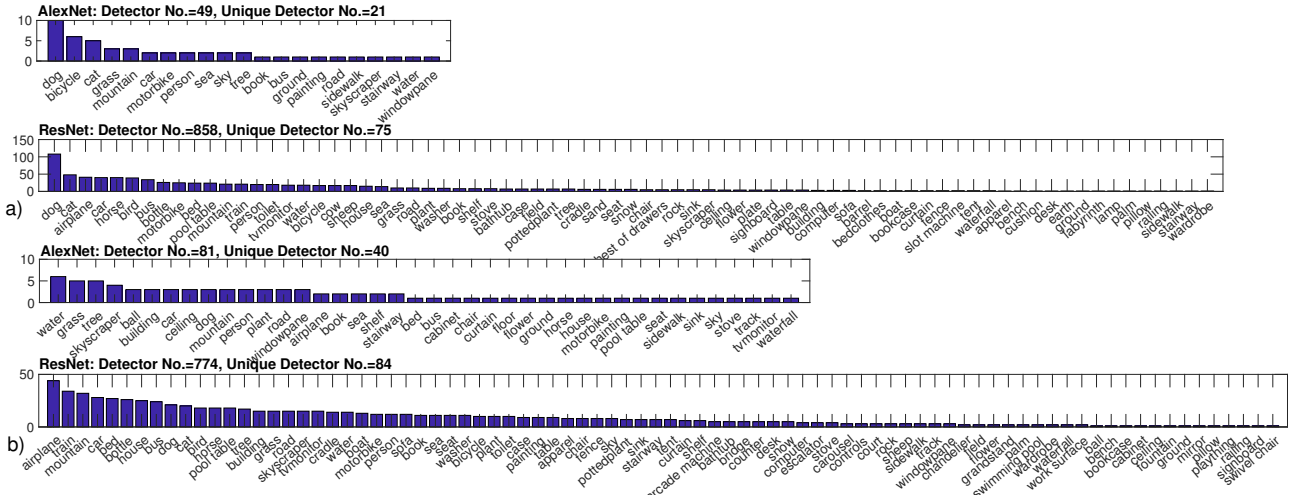

Figure 3. The object vocabularies of networks learned from two different data sources a) ImageNet and b) Places. The emergent object vocabularies used by the networks differ across the training datasets.

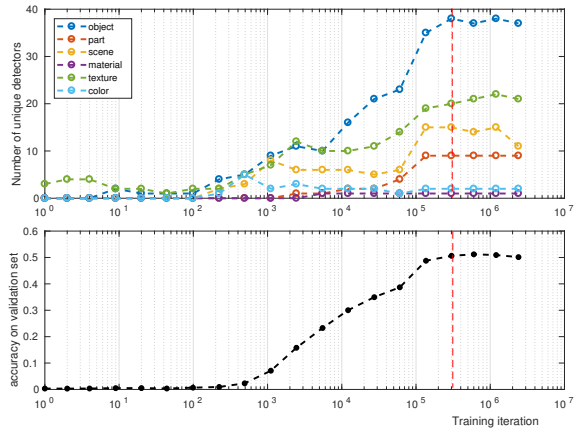

Figure 4. The interpretability of the units at conv5 layer of the baseline model over training iteration. The validation accuracy is also plotted. The baseline model is trained to 300,000 iterations (marked at the red line).

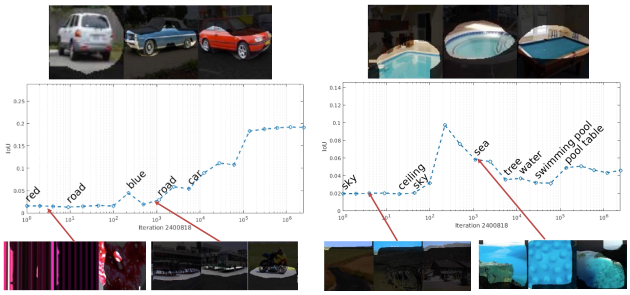

Figure 5. The plot of IoU for the top concept associated with two units over different training iteration. The final concept of each unit is shown above, while the intermediate concepts associated with the units are shown below.

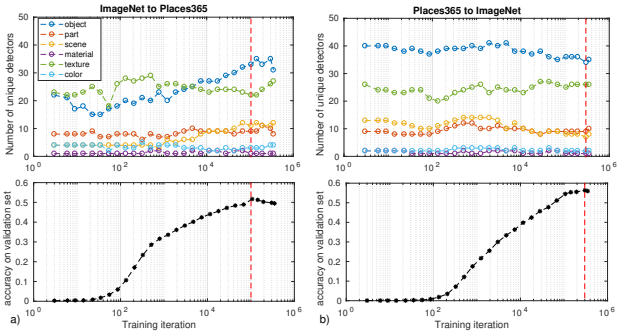

Figure 6. Fine-tune networks between ImageNet and Places. a) ImageNet to Places365 and b) Places365 to ImageNet. The validation accuracy is shown below. The network architecture is the same as AlexNet.

Figure 7 shows the concept evolution of the four units in the network fine-tuned from ImageNet to Places365 and reversely. The top associated concepts keep evolving during the fine-tuning process. For example, in the network fine-tuned from ImageNet to Places365, the first unit which detects the white dog before, evolves to detect the waterfall; the second unit which detects the green concept first, evolves to detect the baseball field. On the other hand, in the network fine-tuned from Places365 to ImageNet, units detecting different concepts converge to detect dog-relevant concepts such as ear and dog head. Interestingly though those units are mutated to detect different concepts, some of them remain to have similar colors or textures.

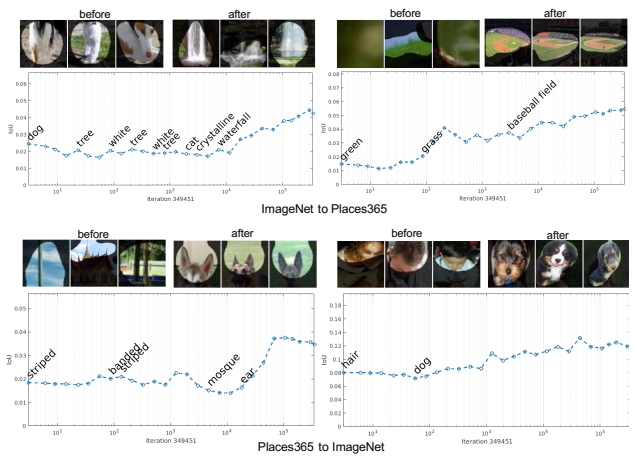

Figure 7. The top associated concepts evolve for four selected units in the networks being fine-tuned. The two units above are from the network fine-tuned from ImageNet to Places365, while the two units below are from the network fine-tuned from Places365 to ImageNet. Above each plot of the IoU we show the unit visualization before and after fine-tuning.

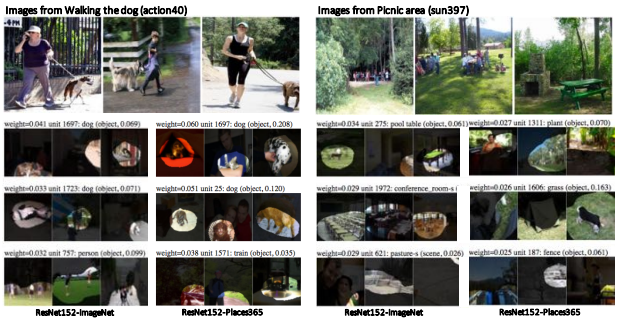

Figure 8. Class-specific units from ResNet152-ImageNet and ResNet152-Places365 on one class from action40 and sun397. For each class, we show three sample images, followed by the top 3 units from ResNet152-ImageNet and ResNet152-Places365 ranked by the class weight of linear SVM to predict that class. SVM weight, detected concept name and the IoU score are shown above each unit.

3.3. Explanatory factors for the deep features

After we quantify the interpretations of the units inside the deep visual representation, the unit activation along with the interpreted label could be used as the explanatory factors for analyzing the prediction given by the deep features. Previous work (**?**) uses the weighted sum of the unit activation maps to highlight which image regions are most informative to the prediction, here we further decouple at individual unit level to segment the informative image regions.

We first plot the *Class-specific units*. After the linear SVM is trained, we can rank the elements of the feature according to their SVM weights to obtain the elements of the deep features which contribute most to that class. Those elements are units that act as explanatory factors, and we call those top ranked units associated with each output class *class-specific units*. Fig.8 shows the class-specific units of ResNet152-ImageNet and ResNet152-Places365 for one class from action40 and sun397 respectively. For example, for the *Walking the dog* class from action40, the top three class-specific units from ResNet152-ImageNet are two dog detection unit and one person detection unit; for the *Picnic area* class from sun397, the top three class-specific units from ResNet152-Places365 are plant detection unit, grass detection unit, and fence detection unit. The intuitive match between visual detectors and the classes they explain suggests that visual detectors from CNNs are behaving like the bag-of-semantic-words visual features.

We further use the individual units identified as concept detectors to build an explanation of the individual image prediction given by a classifier. The procedure is as follows: Given any image, let the unit activation of the deep feature (for ResNet the `GAP` activation) as $[x_1, x_2, ..., x_N]$, where each x_n represents the value summed up from the activation map of unit n. Let the top prediction's SVM response be $s = \sum_n w_n x_n$, where $[w_1, w_2, ..., w_N]$ is the SVM's learned weight. We get the top ranked units in Figure 9 by ranking $[w_1 x_1, w_2 x_2, ..., w_N x_N]$, which are the unit activations weighted by the SVM weight for the top predicted class. Then we simply upsample the activation map of the top ranked unit to segment the image.

The image segmentation using the individual unit activation are plotted in Fig. 9a. The unit segmentation explain the prediction explicitly. For example, the prediction for the first image is *Gardening*, the explanatory units detect plant, grass, person, flower, and pot. The prediction for the second image is *Riding a horse*, the explanatory units detect horse, fence and dog. We also plot some wrongly predicted samples in Figure 9b. The segmentation gives the intuition why the classifier made mistakes. For example, for the first image the classifier predicts it as *cutting vegetables* rather than the true label *gardening*, because the second unit wrongly consider the ground as table.

4. Conclusion

In this work we apply the newly proposed framework *Network Dissection* to analyze the evolution of the interpretations of units over the training iterations. We further show that the interpretations of units are explanatory factors for the deep features used in the visual recognition.

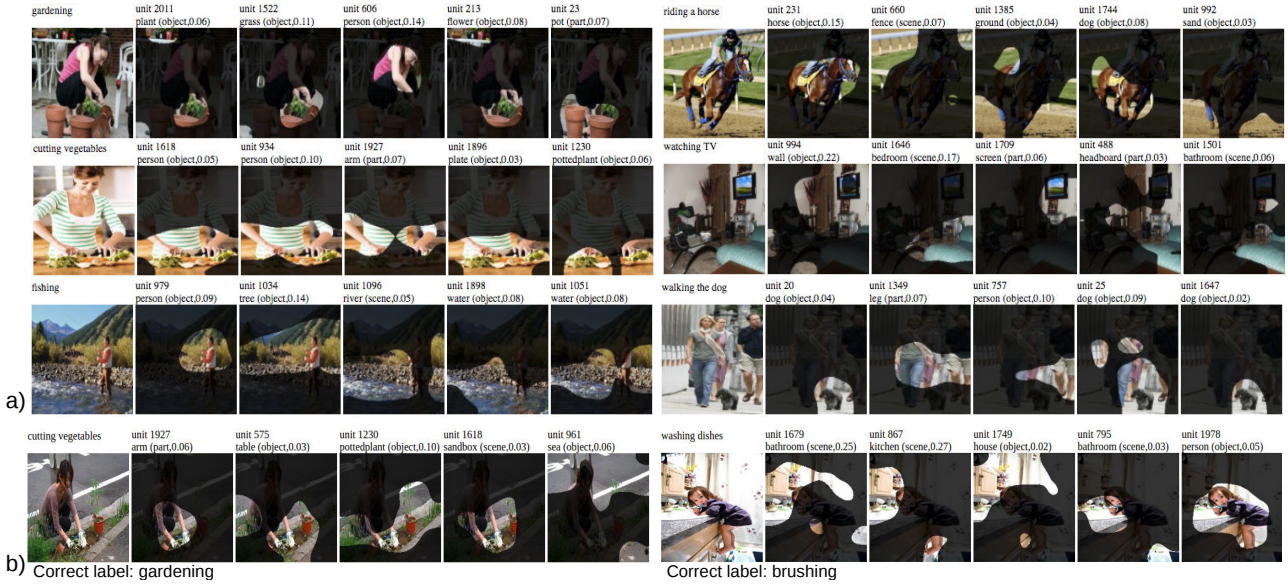

Figure 9. Segmenting images using the top activated units weighted by the class label from ResNet152-Places365 deep feature. a) the correctly predicted samples. b) the wrongly predicted samples.

References

Bau, David, Zhou, Bolei, Khosla, Aditya, Oliva, Aude, and Torralba, Antonio. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

Bell, Sean, Bala, Kavita, and Snavely, Noah. Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH)*, 2014.

Chen, Xianjie, Mottaghi, Roozbeh, Liu, Xiaobai, Fidler, Sanja, Urtasun, Raquel, and Yuille, Alan. Detect what you can: Detecting and representing objects using holistic models and body parts. In *Proc. CVPR*, 2014.

Cimpoi, Mircea, Maji, Subhransu, Kokkinos, Iasonas, Mohamed, Sammy, and Vedaldi, Andrea. Describing textures in the wild. In *Proc. CVPR*, 2014.

Jolliffe, Ian. *Principal component analysis*. Wiley Online Library, 2002.

Maaten, Laurens van der and Hinton, Geoffrey. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Mahendran, Aravindh and Vedaldi, Andrea. Understanding deep image representations by inverting them. *Proc. CVPR*, 2015.

Mottaghi, Roozbeh, Chen, Xianjie, Liu, Xiaobai, Cho, Nam-Gyu, Lee, Seong-Whan, Fidler, Sanja, Urtasun, Raquel, and Yuille, Alan. The role of context for object detection and semantic segmentation in the wild. In *Proc. CVPR*, 2014.

Nguyen, Anh, Dosovitskiy, Alexey, Yosinski, Jason, Brox, Thomas, and Clune, Jeff. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *In Advances in Neural Information Processing Systems*, 2016.

Simonyan, Karen, Vedaldi, Andrea, and Zisserman, Andrew. Deep inside convolutional networks: Visualising image classification models and saliency maps. *International Conference on Learning Representations Workshop*, 2014.

Zeiler, Matthew D and Fergus, Rob. Visualizing and understanding convolutional networks. *Proc. ECCV*, 2014.

Zhou, Bolei, Khosla, Aditya, Lapedriza, Agata, Oliva, Aude, and Torralba, Antonio. Object detectors emerge in deep scene cnns. *International Conference on Learning Representations*, 2015.

Zhou, Bolei, Zhao, Hang, Puig, Xavier, Fidler, Sanja, Barriuso, Adela, and Torralba, Antonio. Scene parsing through ade20k dataset. *Proc. CVPR*, 2017.