On Visualizing and Understanding CNN

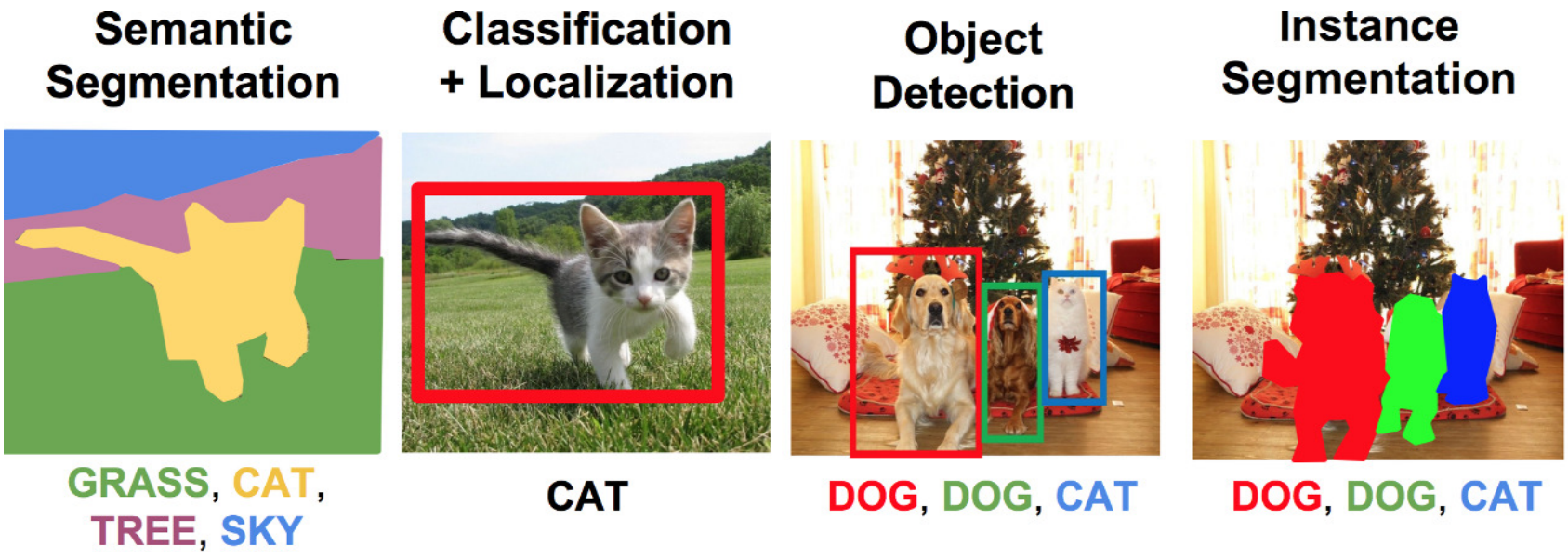
CS189/289A: Introduction to Machine Learning

Stella Yu

UC Berkeley

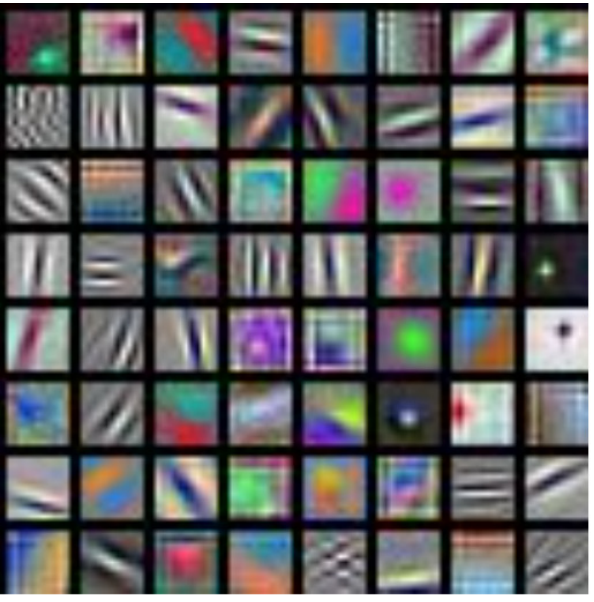
21 November 2017

What Has the Neural Net Learned?



1. Direct visualization of filters, activations, and optimal stimuli
2. Reconstruction by deconvolution
3. Activation maximization
4. Saliency maps
5. Code inversion
6. Semantic Interpretation

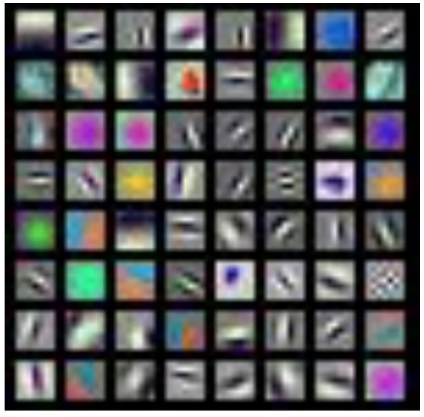
Direct Visualization of Filters at Layer 1



AlexNet:
64 x 3 x 11 x 11

ResNet-18:
64 x 3 x 7 x 7

ResNet-101:
64 x 3 x 7 x 7



DenseNet-121:
64 x 3 x 7 x 7

Direct Visualization of Filters at Layer 1+

Weights:

layer 1 weights

16 x 3 x 7 x 7

Weights:

layer 2 weights

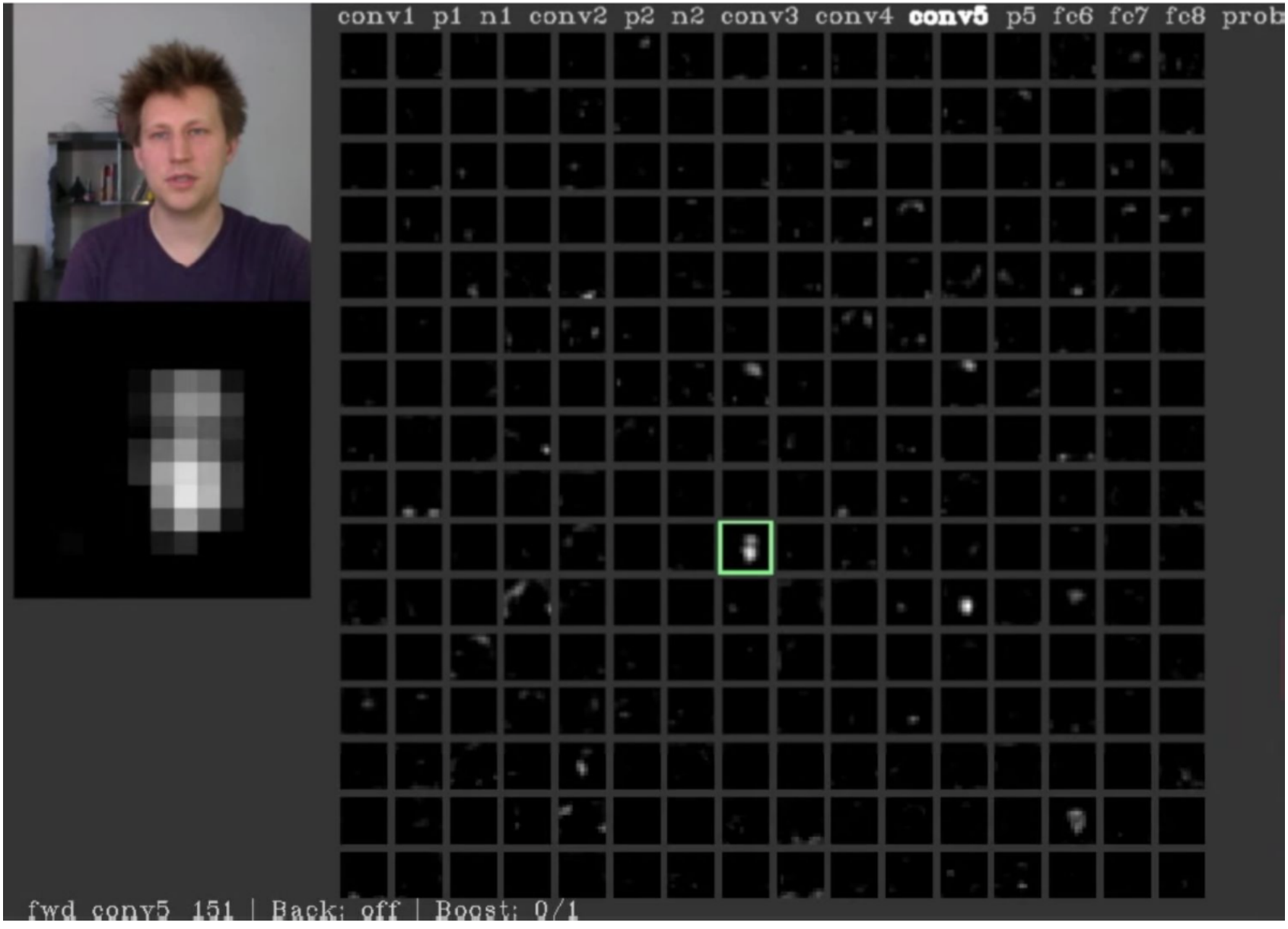
20 x 16 x 7 x 7

Weights:

layer 3 weights

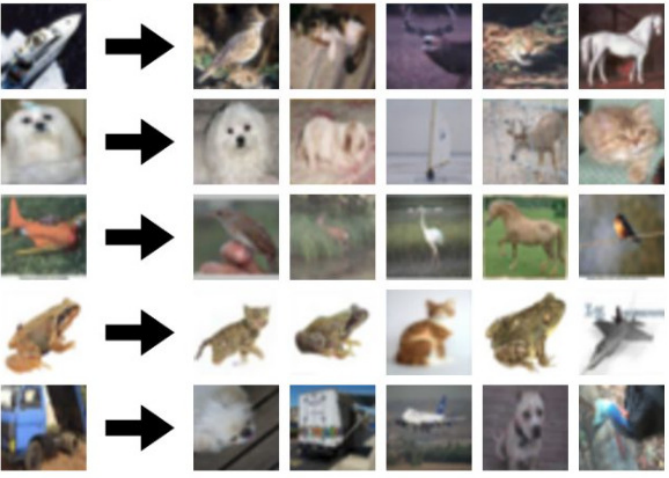
20 x 20 x 7 x 7

Direct Visualization of Activation Maps



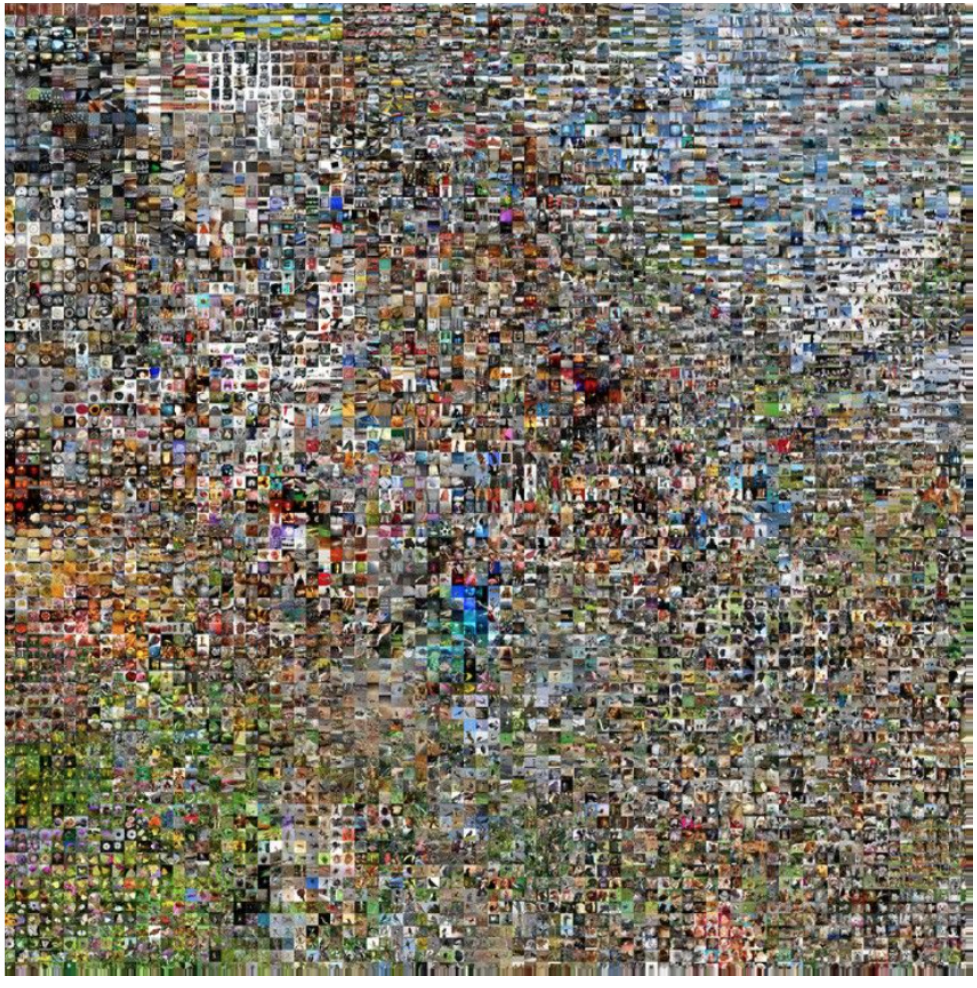
Visualizing The Last Feature Layer with Images

Recall: Nearest neighbors in pixel space



Test image L2 Nearest neighbors in feature space

Visualizing the Last Feature Layer with tSNE



A Single Neuron in the CNN

- Fully connected layer
  Each node has its own weights and responses.

- Convolutional layer
  All the nodes in the activation map share the same weights and have their own responses. The weights define a neuron / channel / feature / feature detector. Each node reveals this feature detector's response to an instance of the input image patch seen in its own sliding window.

- Pooling and ReLU layers
  Each node has its own response that depends on its inputs. No parameters.

Backprop the Error: Convolutional Layer

- Matrix form and element-wise:

$$E = Loss(Y) \tag{1}$$

$$Y = WX \tag{2}$$

$$Y_i = \sum_{j=1}^{n} W_{ij} X_j \tag{3}$$

$$\frac{\partial E}{\partial X_i} = \sum_{j=1}^{m} \frac{\partial E}{\partial Y_j} \frac{\partial Y_j}{\partial X_i} = \sum_{j=1}^{m} \frac{\partial E}{\partial Y_j} W_{ji} \tag{4}$$

$$\frac{\partial E}{\partial X} = W' \frac{\partial E}{\partial Y} \tag{5}$$

- Forward and backward convolutions:

$$X \otimes W \rightarrow Y \tag{6}$$

$$\frac{\partial E}{\partial X} \leftarrow \frac{\partial E}{\partial Y} \otimes W' \tag{7}$$

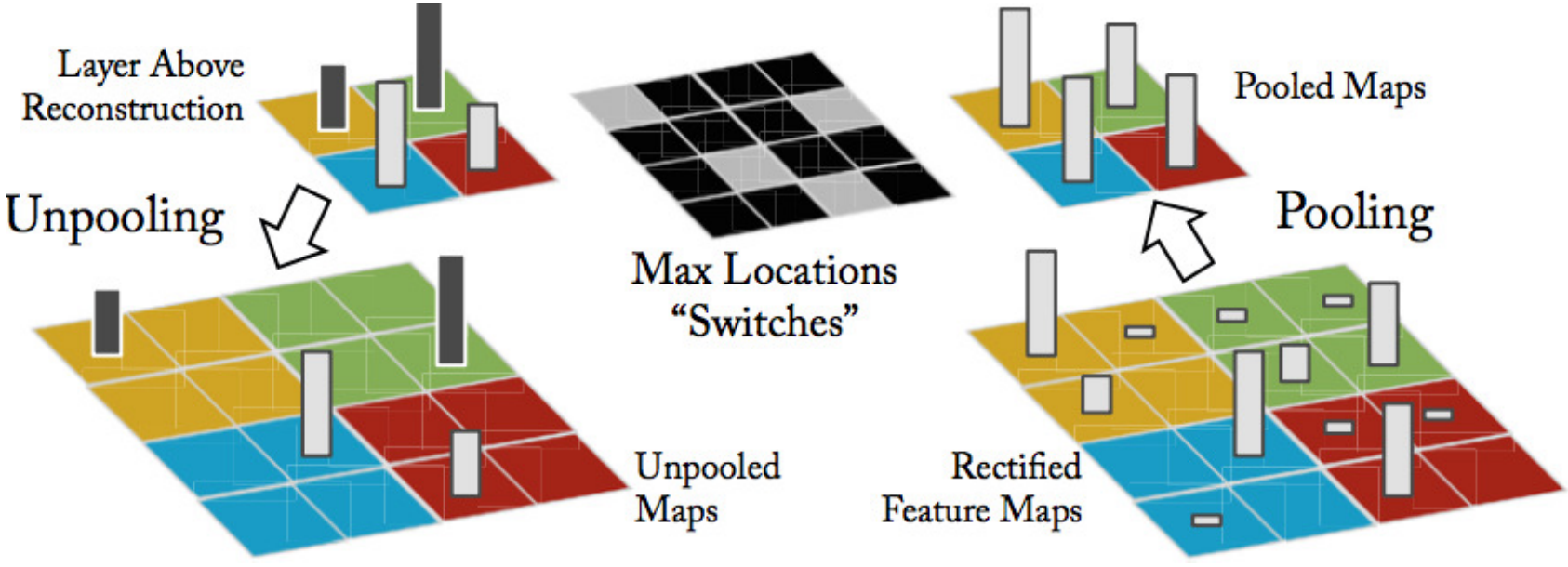
Backprop the Error: Maxpooling Layer

- Non-invertible, approximate

$$y = \max_i X_i \tag{8}$$

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial y} \cdot 1(X = y) \tag{9}$$

- Record the winner location: $\arg\max X$ in the forward path



- Not just the model parameters, but input-image dependent

Backprop the Error: ReLU Layer

- Standard: thresholding on the input value

$$y = \max(x, 0) \tag{10}$$

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} \cdot 1(x > 0) \tag{11}$$

- DeconvNet: thresholding on the output value

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} \cdot 1(\frac{\partial E}{\partial y} > 0) \tag{12}$$

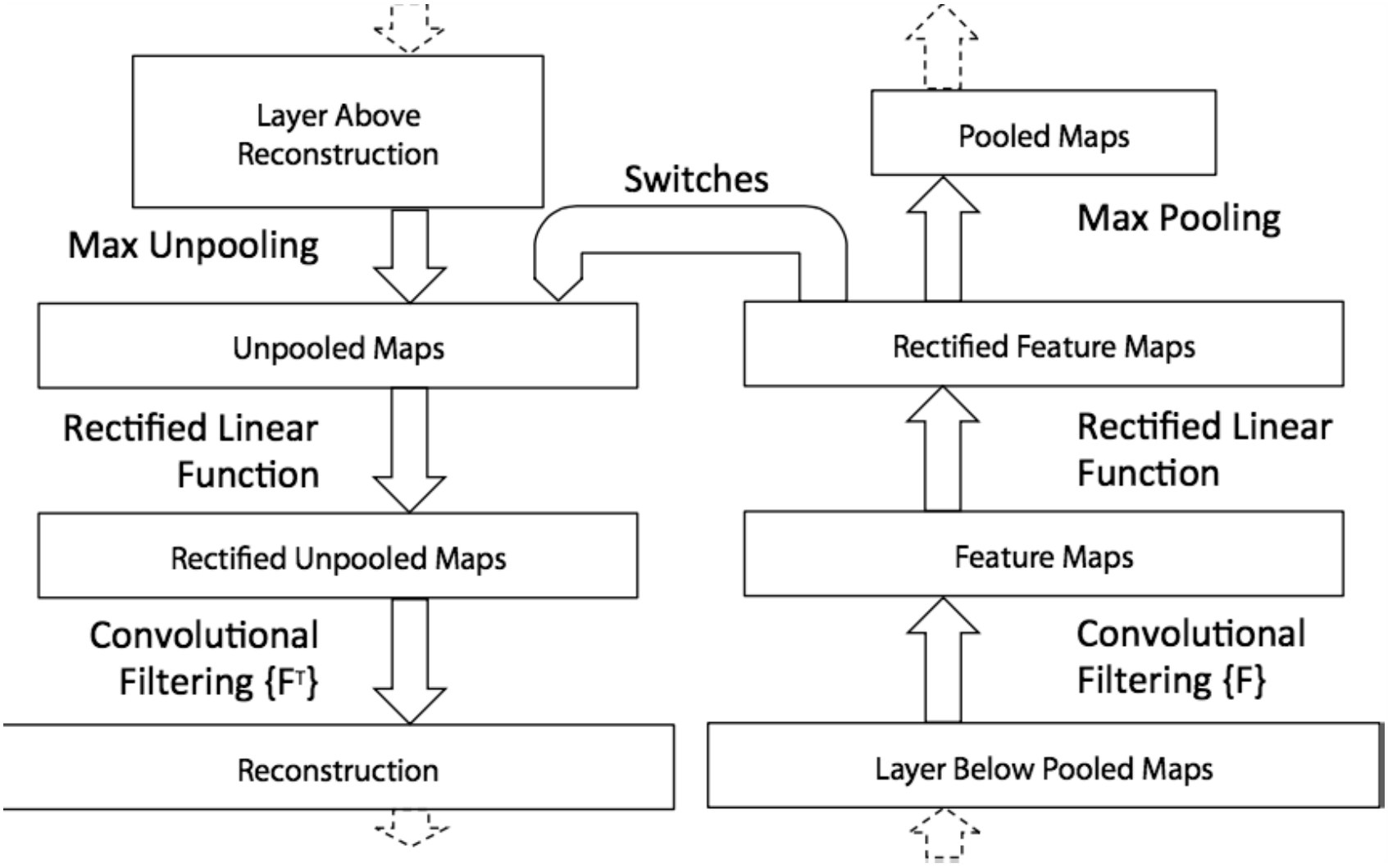
- Guided backprop: thresholding on both input and output

$$\frac{\partial E}{\partial x} = \frac{\partial E}{\partial y} \cdot 1(\frac{\partial E}{\partial y} > 0) \cdot 1(x > 0) \tag{13}$$

Deconvolutional Network for Visualization

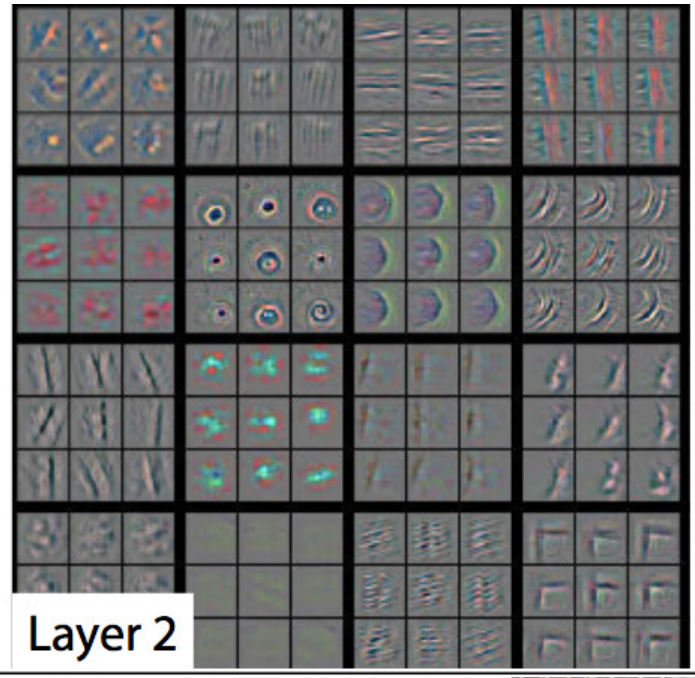
- Map activities at intermediate layers back to the input pixel space, showing what input pattern originally caused a given activation in the feature maps.
- A DeconvNet can be thought of as a ConvNet model that uses the same components (filtering, pooling) but in reverse, so instead of mapping pixels to features does the opposite.
- To examine a given activation, set all other activations in the layer to zero and pass the feature maps as input to the DeconvNet.
- *Visualizing and Understanding Convolutional Networks* Matthew D. Zeiler and Rob Fergus, ECCV, 2014.

Deconvolutional Network for Visualization

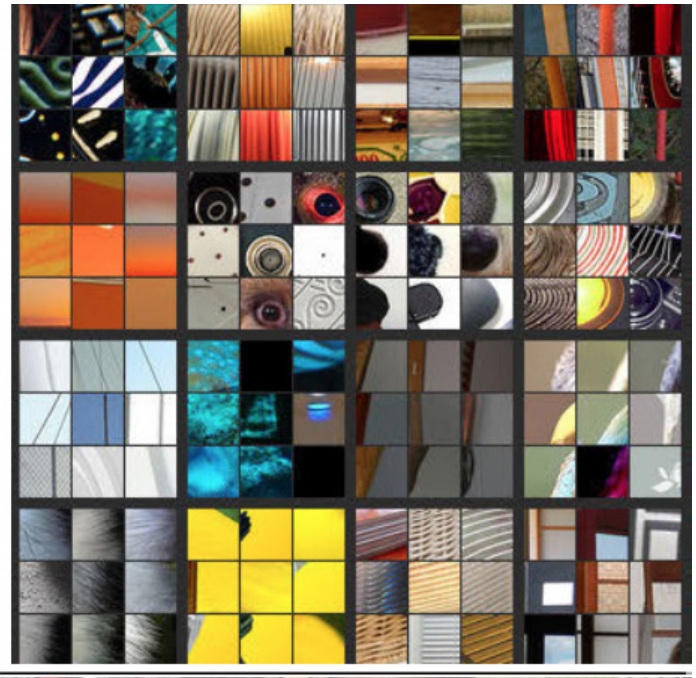


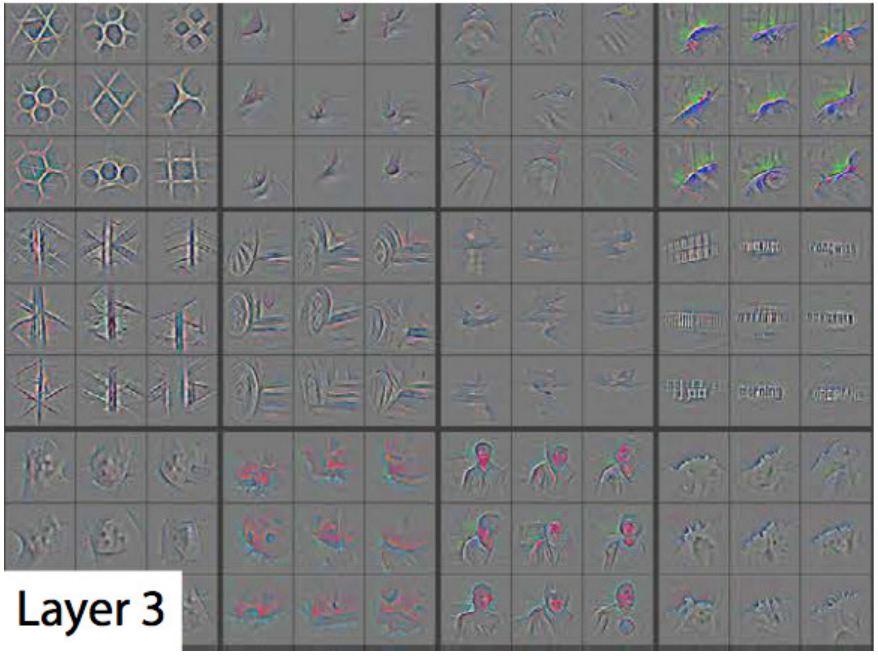
Convolutional Layers: Top 9 Activation Maps

Layer 1

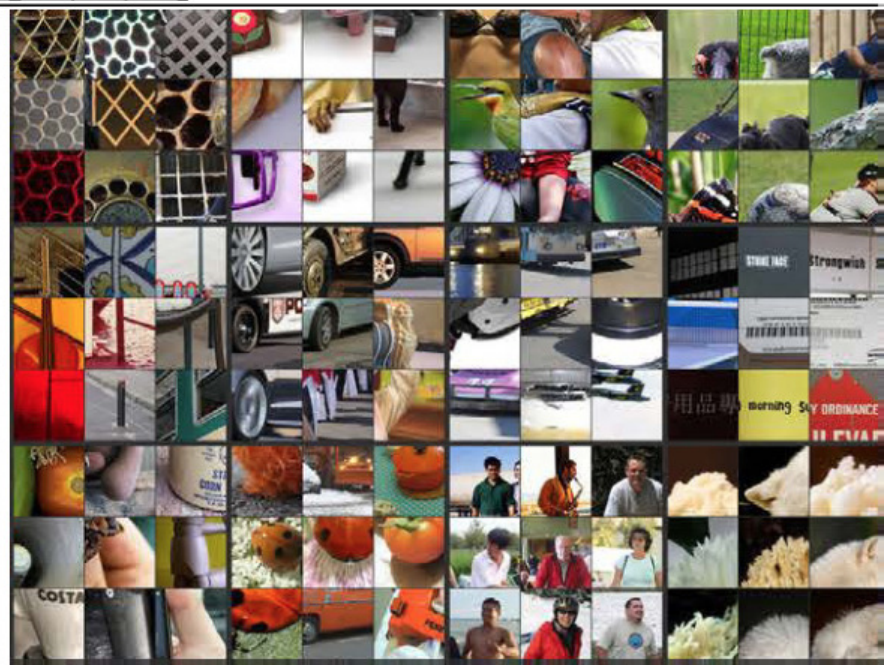


Layer 2

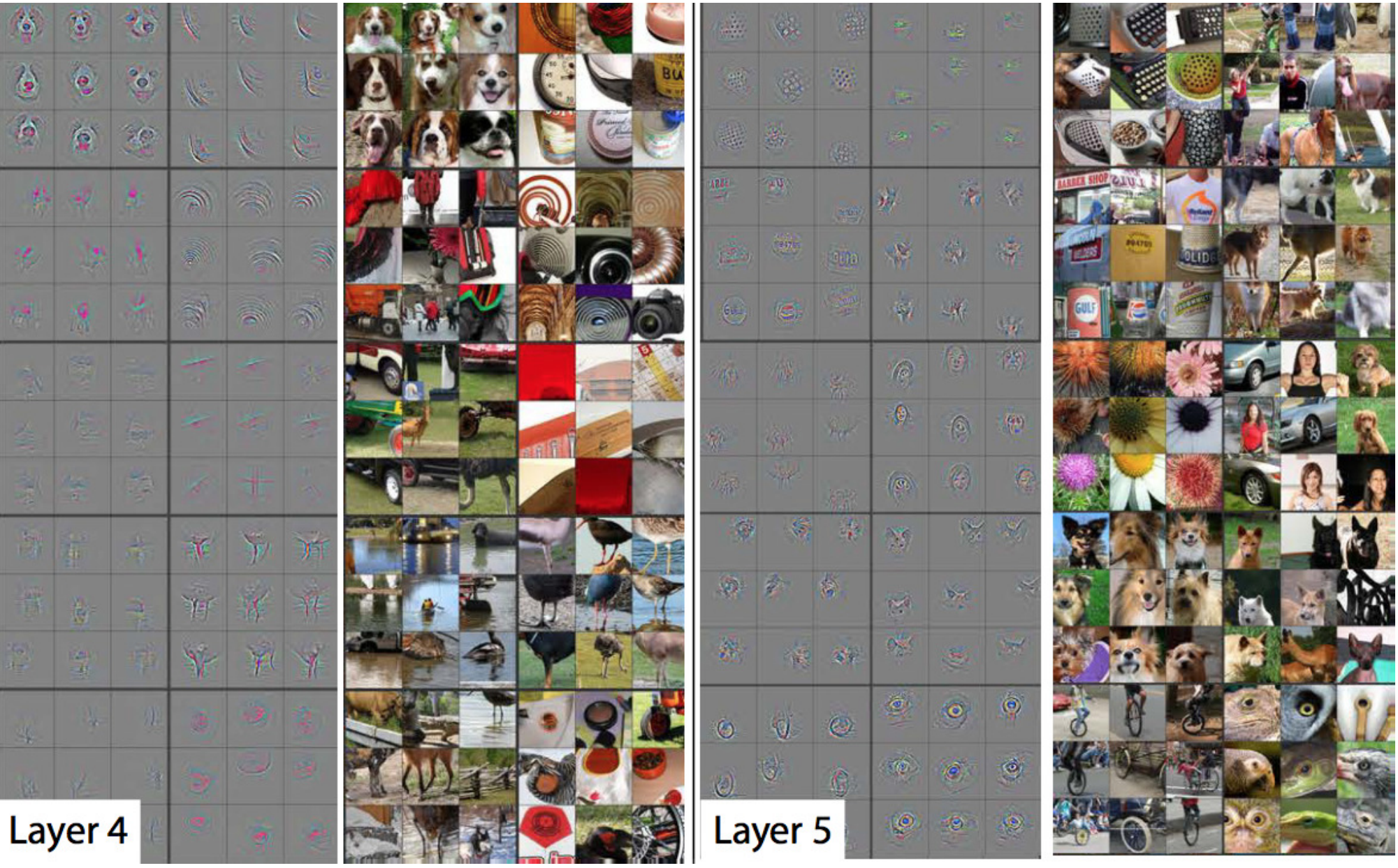




Layer 3



Convolutional Layers: Top 9 Activation Maps



Evolution of Conv Filters during Training

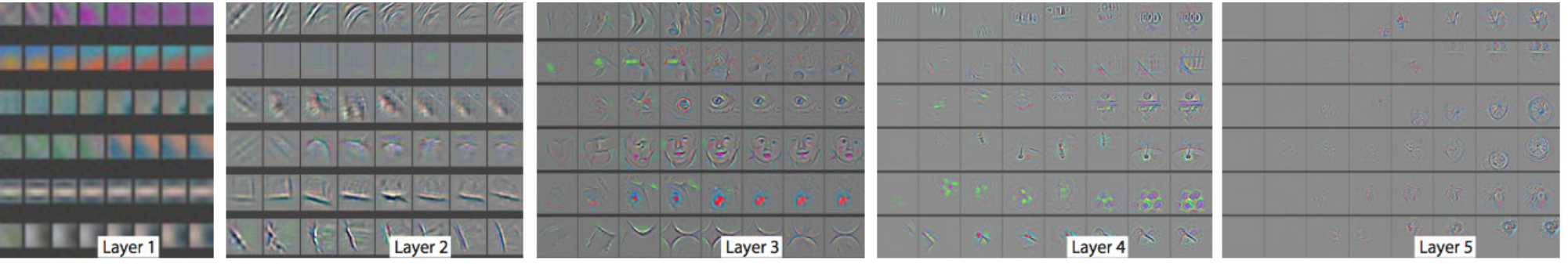


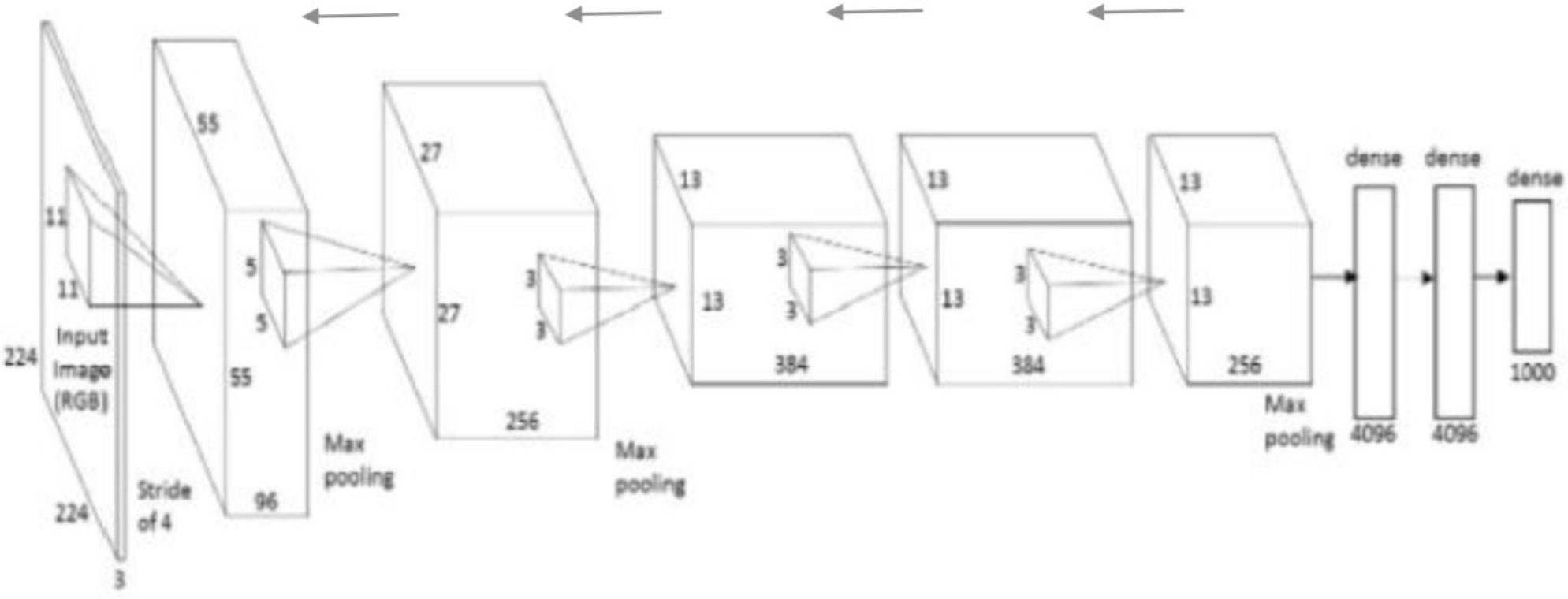
Fig. 4. Evolution of a randomly chosen subset of model features through training. Each layer's features are displayed in a different block. Within each block, we show a randomly chosen subset of features at epochs [1,2,5,10,20,30,40,64]. The visualization shows the strongest activation (across all training examples) for a given feature map, projected down to pixel space using our deconvnet approach. Color contrast is artificially enhanced and the figure is best viewed in electronic form.

Guided Backprop Process

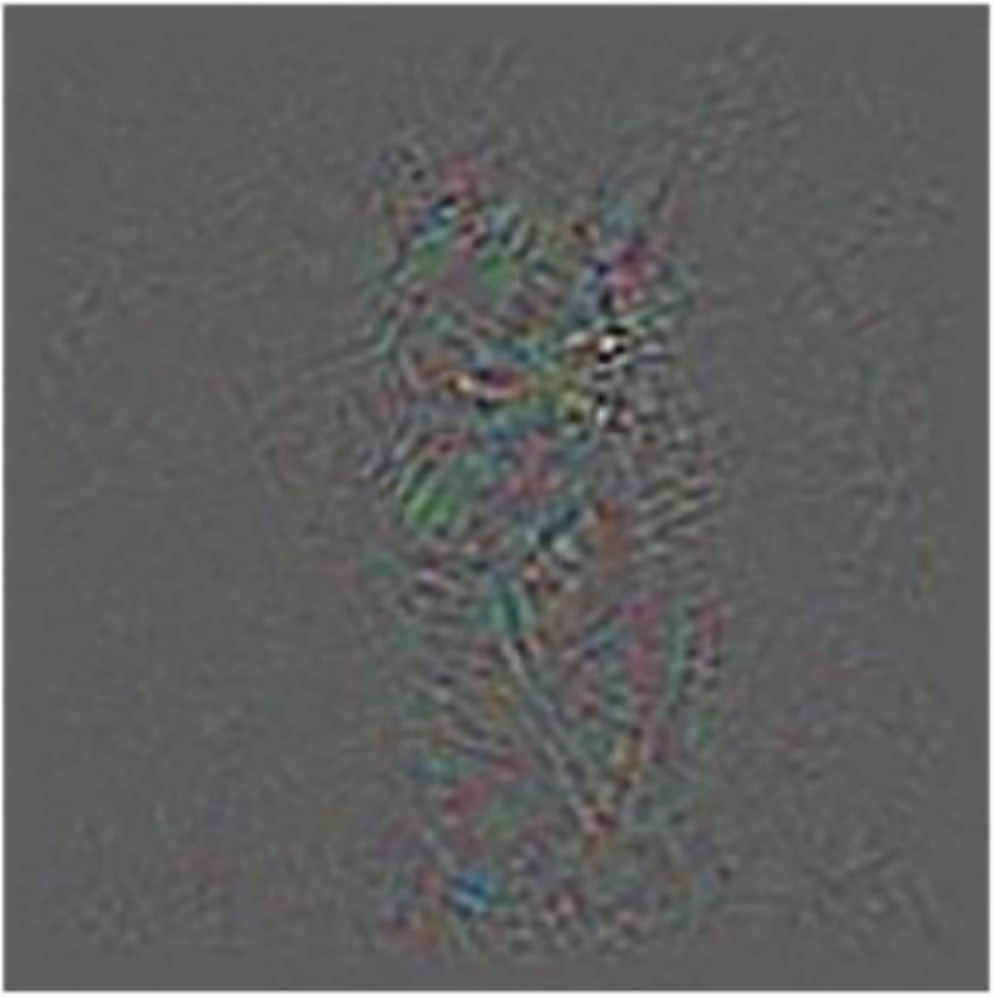
Compute gradient, zero out negatives, backpropagate

Compute gradient, zero out negatives, backpropagate

Compute gradient, zero out negatives, backpropagate



BP vs. Guided BP: Neurons As Detectors



Activation Maximization for Optimal Stimuli

- Given a neural net f parameterized by θ
- Given a neuron n in the network, with response function $f_n(\cdot; \theta)$
- Seek an input image I such that

$$\max_I \quad \underbrace{f_n(I;\theta)}_{\text{network activation}} \quad - \quad \underbrace{\text{prior}(I)}_{\text{admissible image set}} \tag{14}$$

- Optimizing the input with the network weights fixed

Visualizing the Class Model

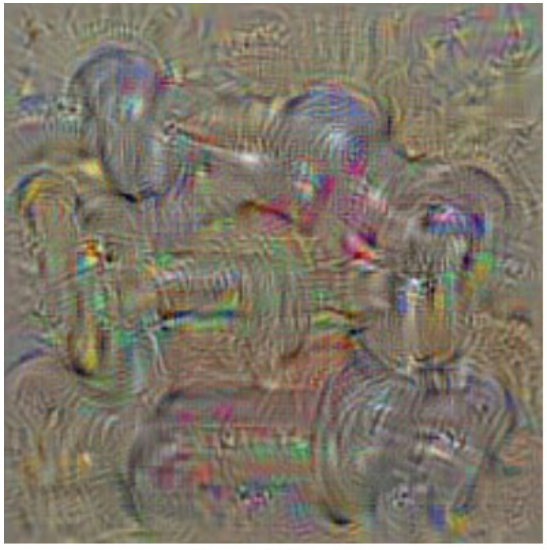
- Maximize the class score with L_2 regularization on the image

$$\arg\max_I \quad S_c(I) - \lambda \|I\|_2^2 \tag{15}$$

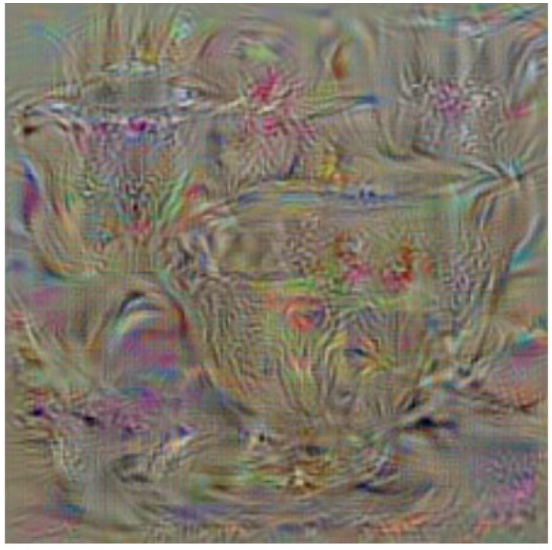
- Initialize with the zero image

- *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*, Karen Simonyan, Andrea Vedaldi, Andrew Zisserman, NIPS, 2014.

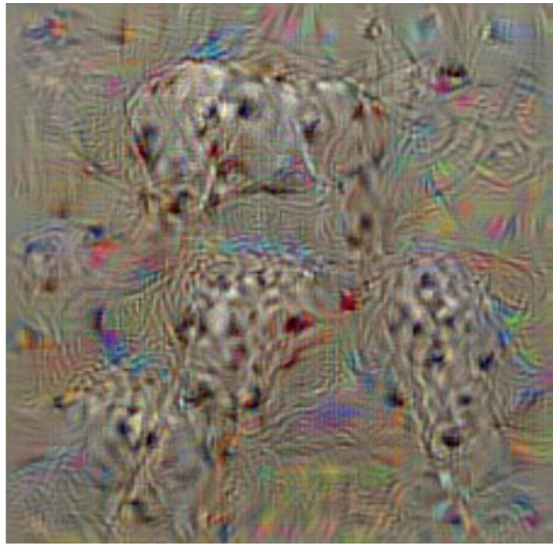
Visualizing the Class Model



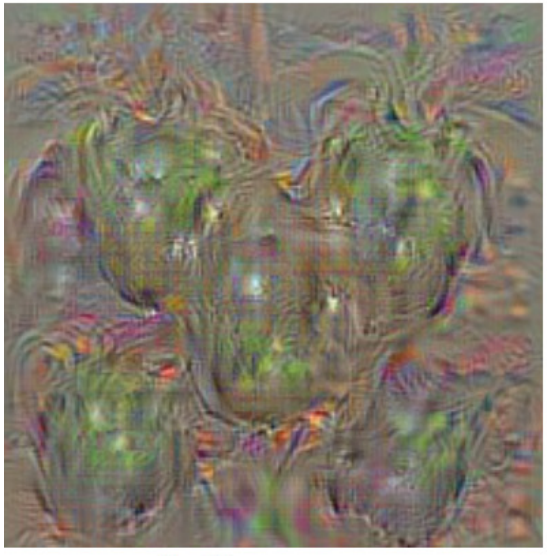
dumbbell



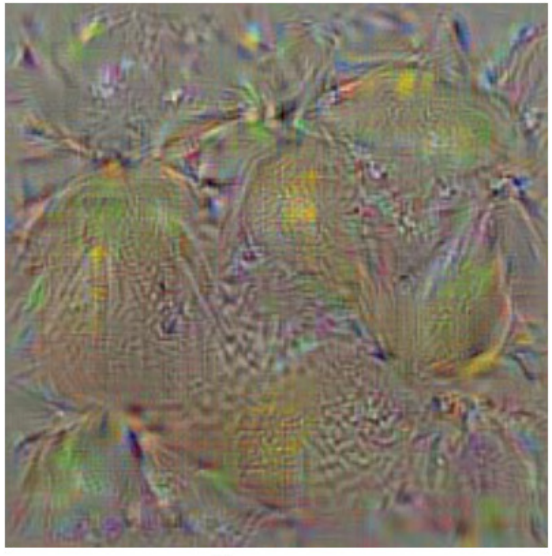
cup



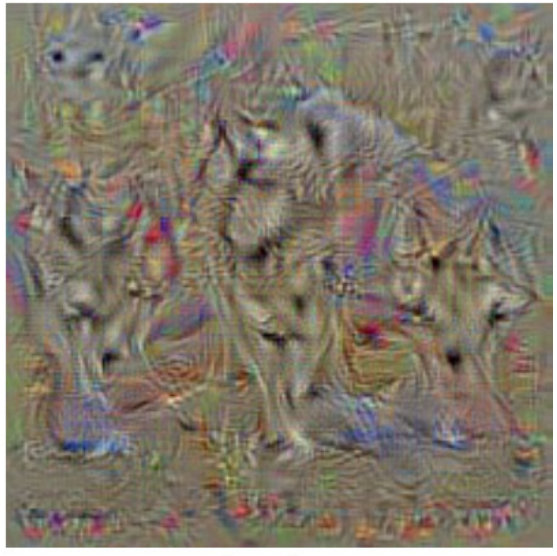
dalmatian



bell pepper

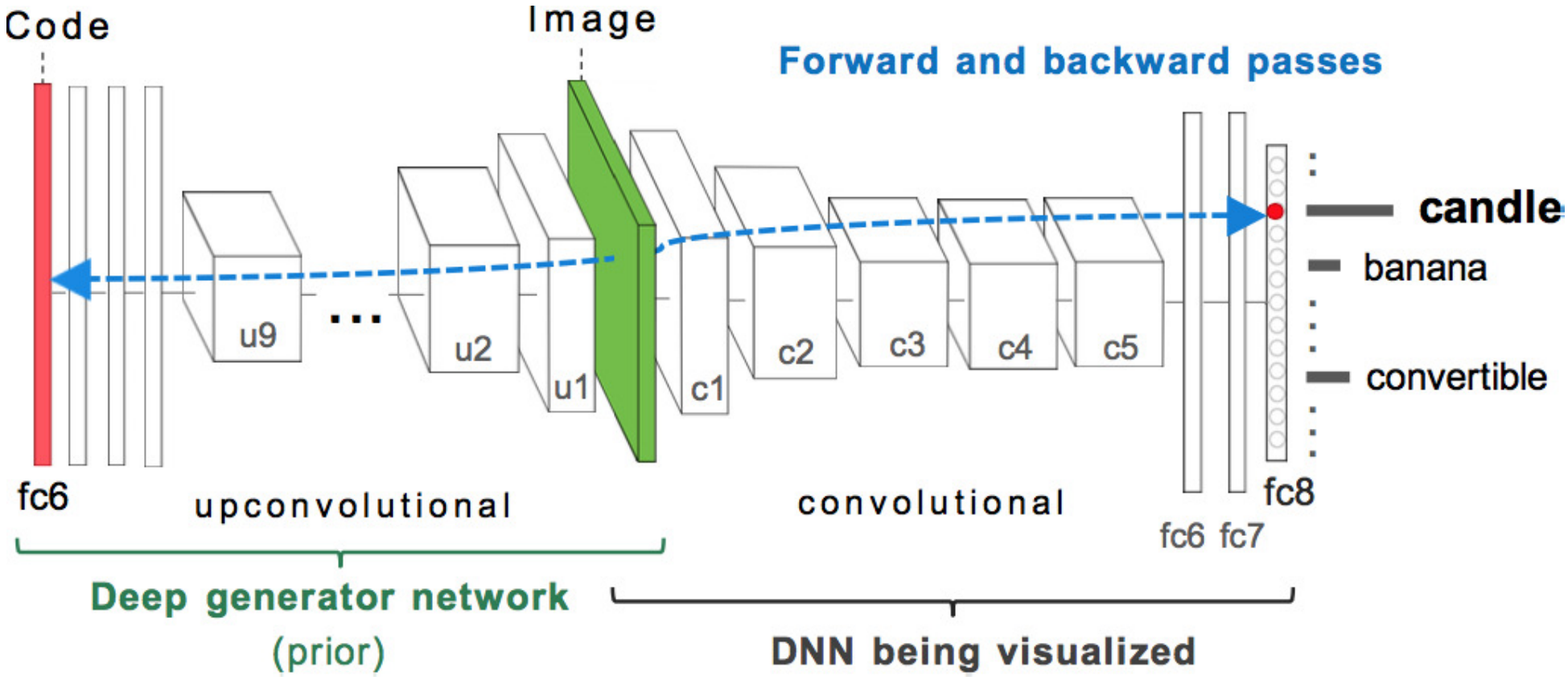


lemon



husky

Synthetize RF with Natural Image Priors



$$\arg\max_{\text{code}} \quad \text{Net}(\text{Generator}(\text{code})) - \lambda \|\text{code}\|^2 \qquad (16)$$

- *Synthesizing the preferred inputs for neurons in neural networks via deep generator networks*, Anh Nguyen, Jason Yosinski, Alexey Dosovitskiy, Thomas Brox, Jeff Clune, NIPS 2016.

ImageNet CNN RF Mapping with Natural Stimuli

Figure 1: Images synthesized from scratch to highly activate output neurons in the CaffeNet deep neural network, which has learned to classify different types of ImageNet images.

Places-CNN RF Mapping with Natural Stimuli

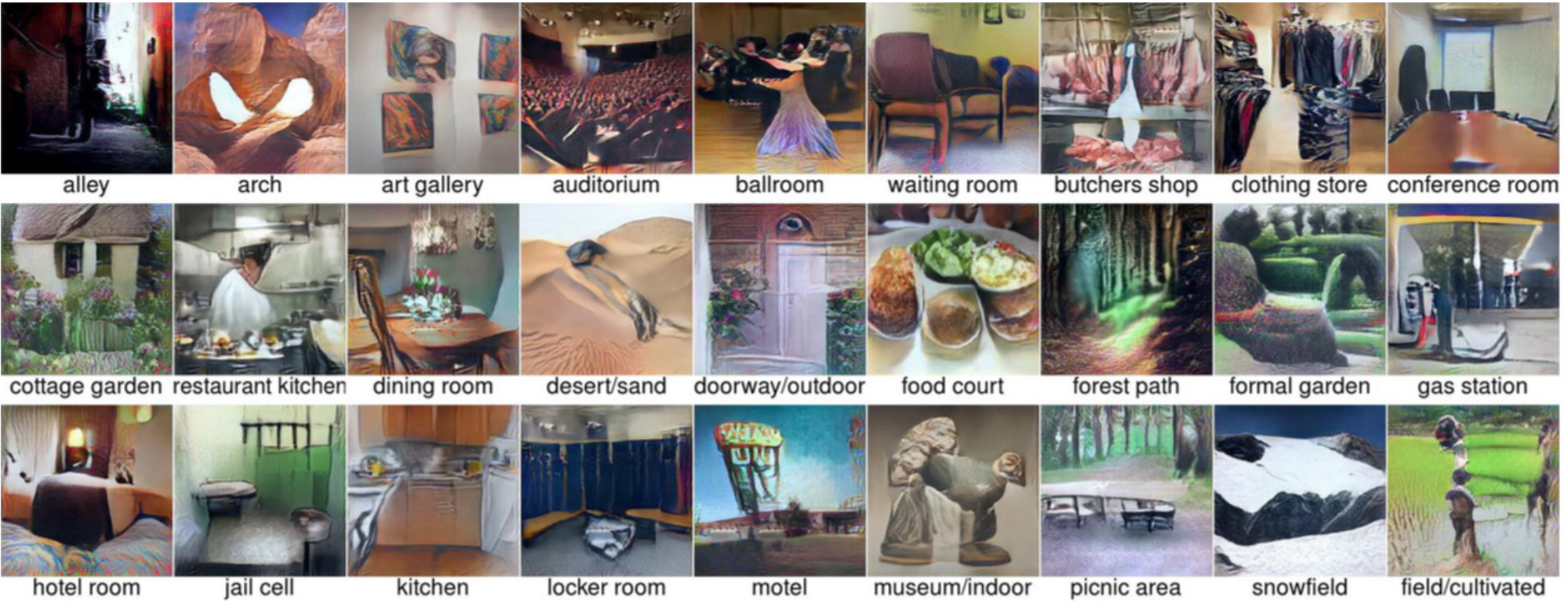


Figure 3: Preferred stimuli for output units of an AlexNet DNN trained on the MIT Places dataset [26], showing that the ImageNet-trained prior generalizes well to a dataset comprised of images of scenes.

Multifaceted Neuronal Responses

Figure 1. **Top:** Visualizations of 8 types of images (feature facets) that activate the same “grocery store” class neuron. **Bottom:** Example training set images that activate the same neuron, and resemble the corresponding synthetic image in the top panel.

Initialize RFs with Response Cluster Means

- *Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks*, Anh Nguyen, Jason Yosinski, Jeff Clune, ICML 2016.

Saliency Map: Which Pixels and How Much?

- Linear score model or first-order Taylor expansion

$$S(I) = w'I + b \tag{17}$$

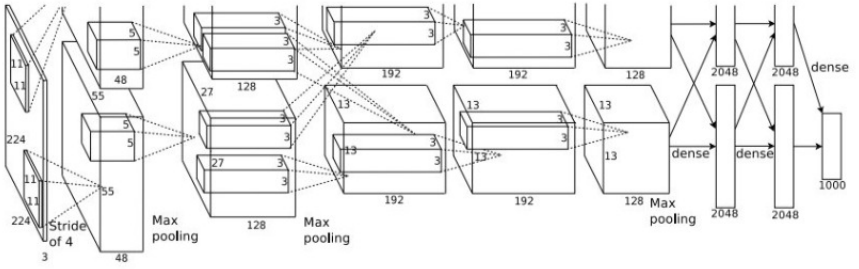
- Gradient: How I should change to maximize $S(I)$

$$\frac{\partial S}{\partial I} = w \tag{18}$$

- Magnitude: Which pixels need to be changed the least to affect the class score the most.

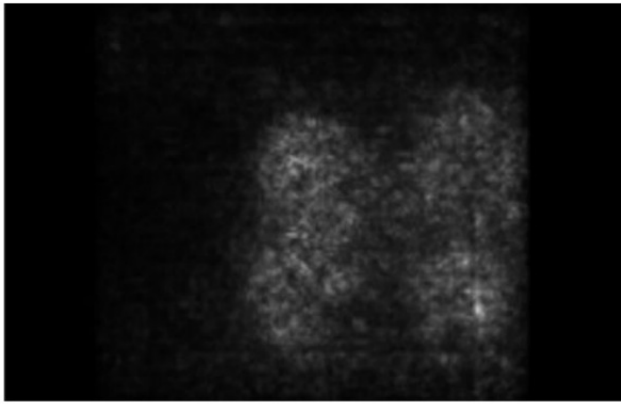
Localization: Visualing Pixel Saliency Map

How to tell which pixels matter for classification?



Dog

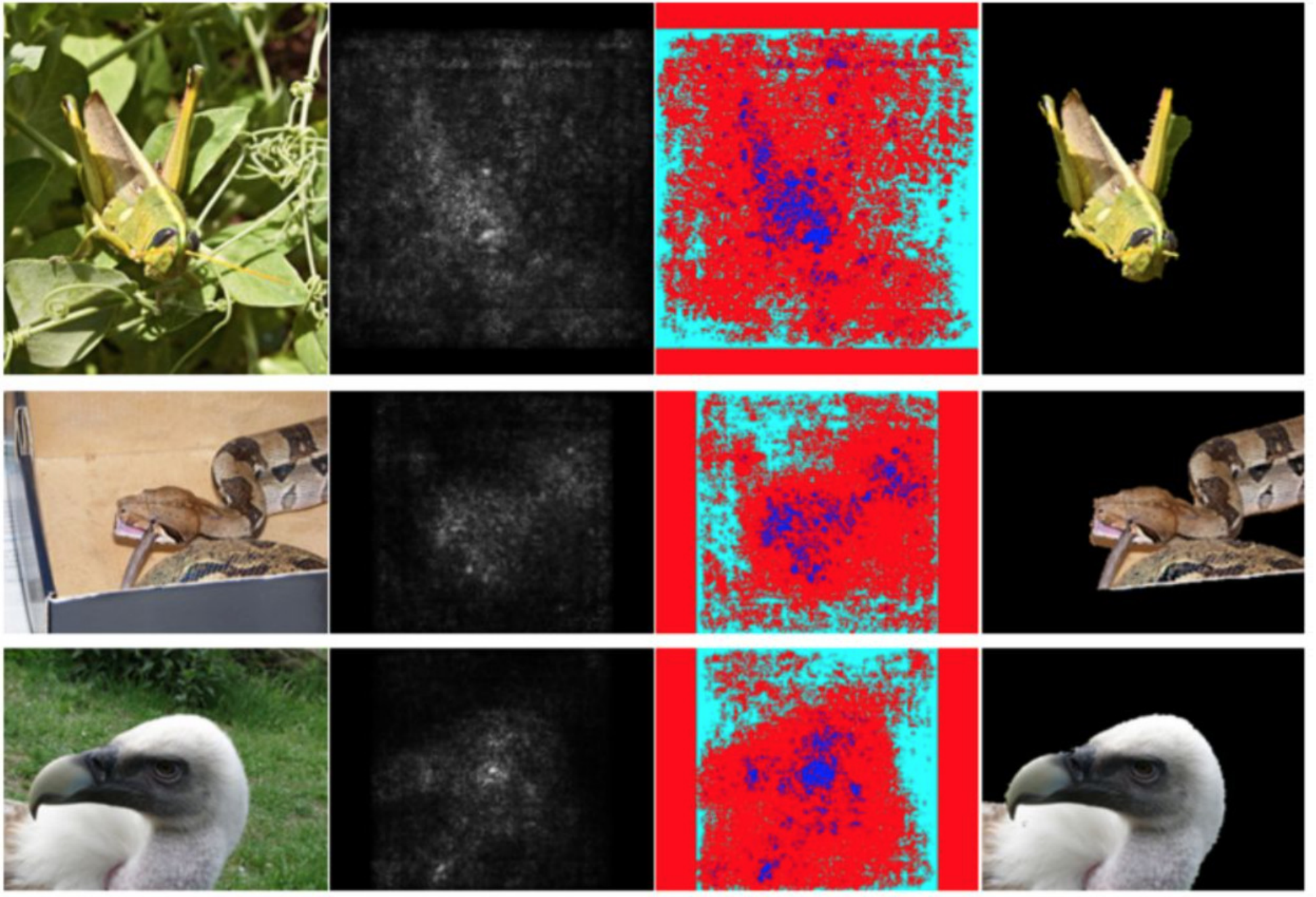
Compute gradient of (unnormalized) class score with respect to image pixels, take absolute value and max over RGB channels



Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Figures copyright Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 2014; reproduced with permission.

Unsupervised Segmentation with Pixel Saliency



RF Mapping by Global Average Pooling

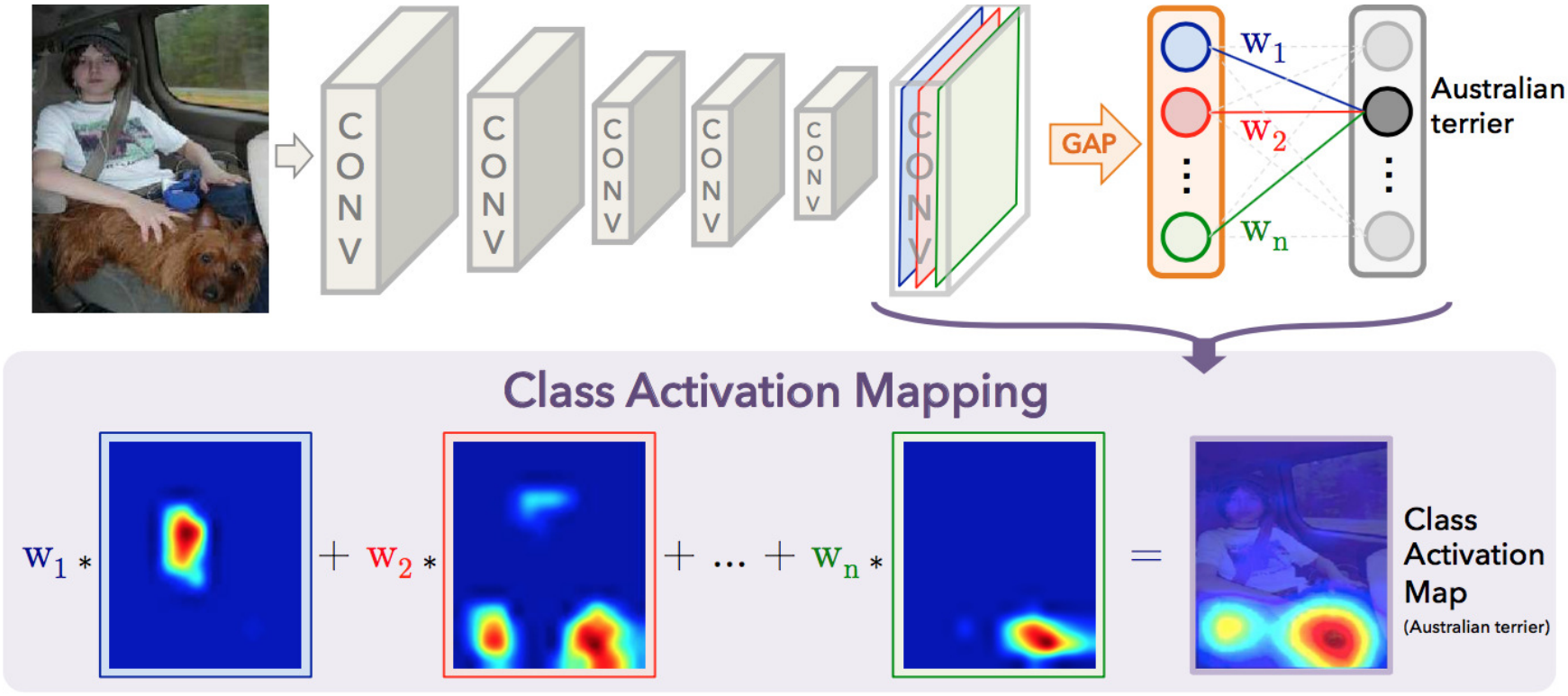


Figure 2. Class Activation Mapping: the predicted class score is mapped back to the previous convolutional layer to generate the class activation maps (CAMs). The CAM highlights the class-specific discriminative regions.

- *Learning Deep Features for Discriminative Localization*, Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba, CVPR 2016.

RF Mapping by Global Average Pooling

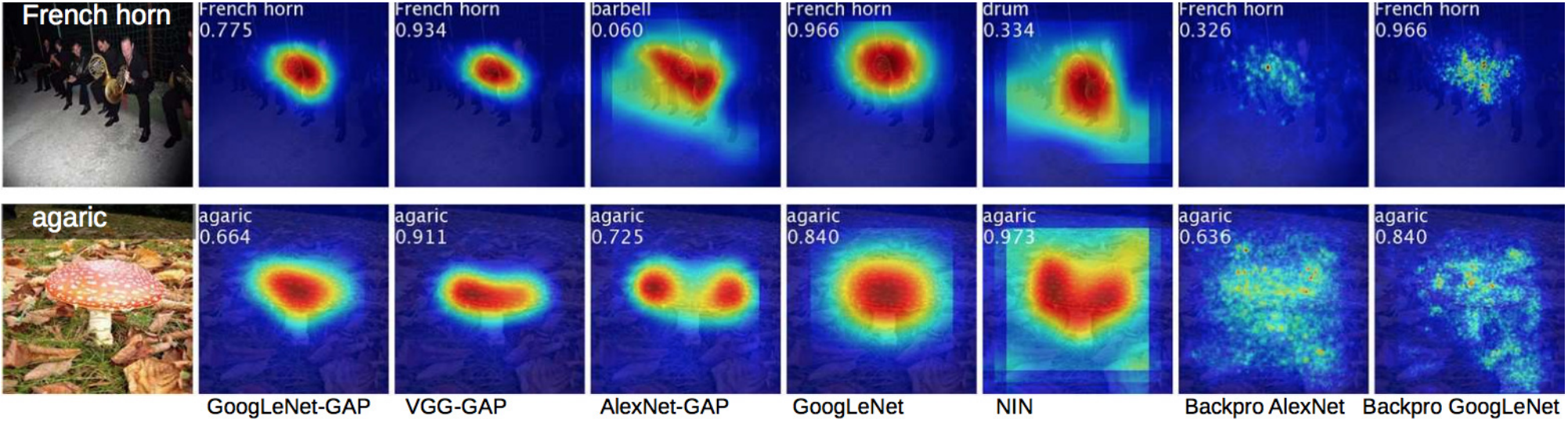


Figure 5. Class activation maps from CNN-GAPs and the class-specific saliency map from the backpropagation methods.

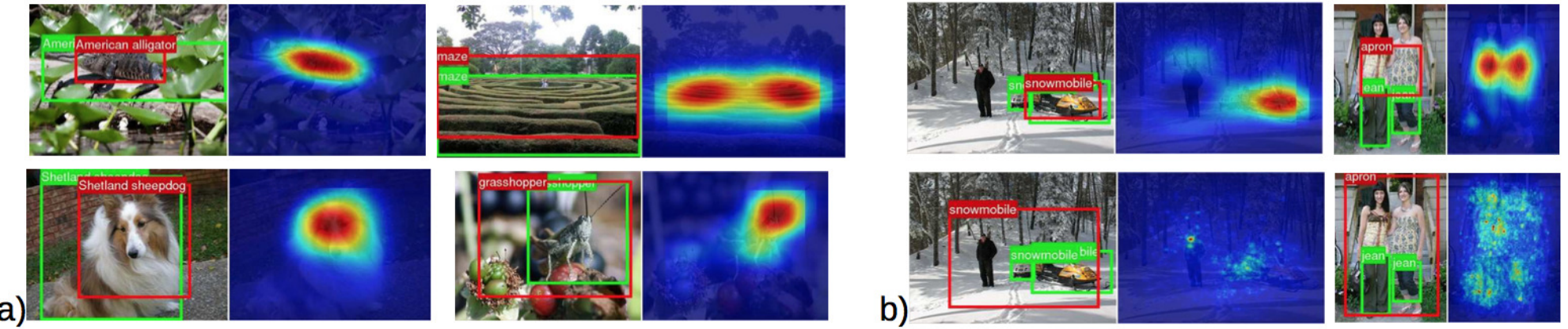
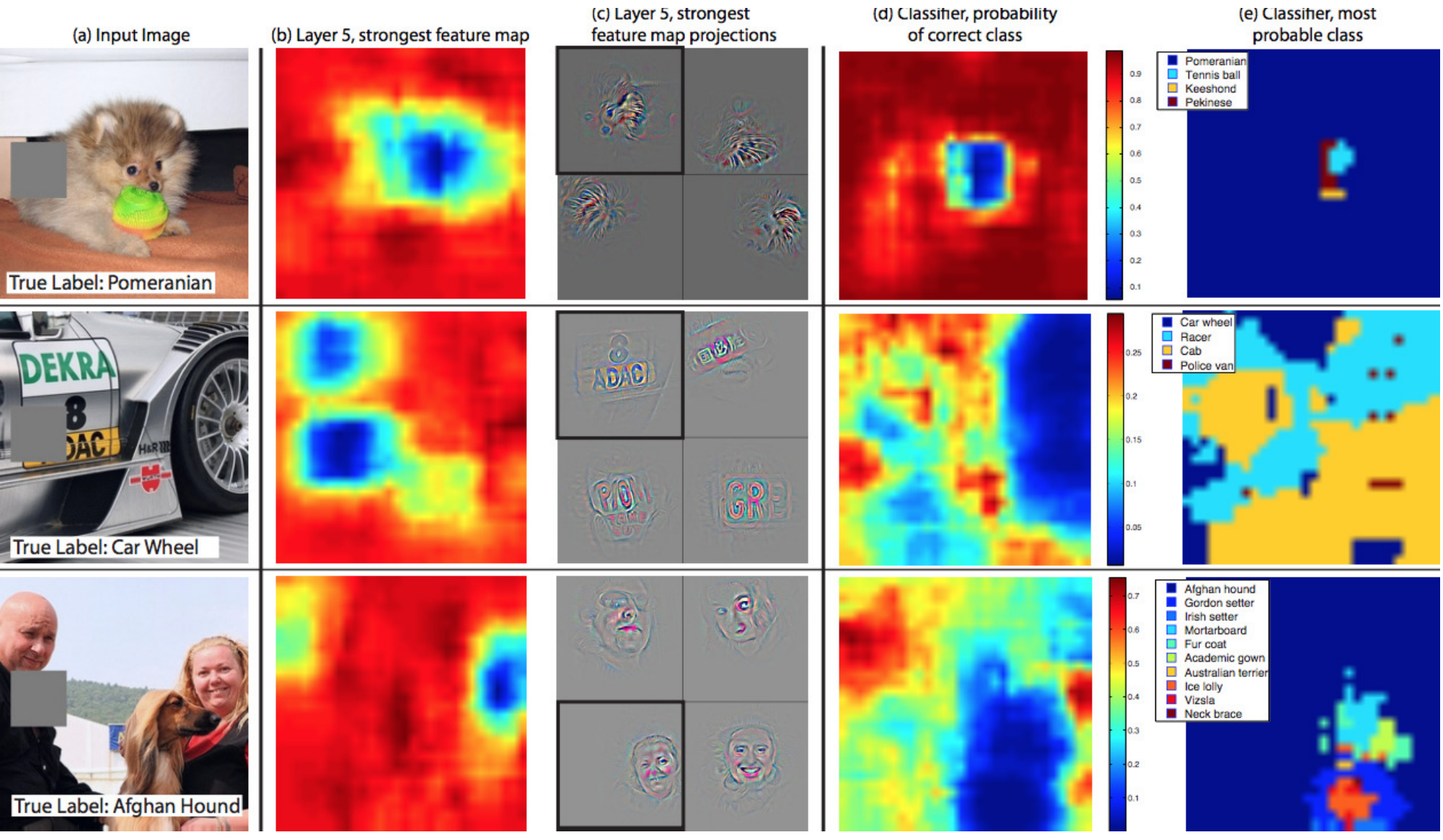


Figure 6. a) Examples of localization from GoogleNet-GAP. b) Comparison of the localization from GooleNet-GAP (upper two) and the backpropagation using AlexNet (lower two). The ground-truth boxes are in green and the predicted bounding boxes from the class activation map are in red.

Activation w/o Contribution: Occlusion Heat Map



Tanaka: Critical Feature for Neurons, 1993

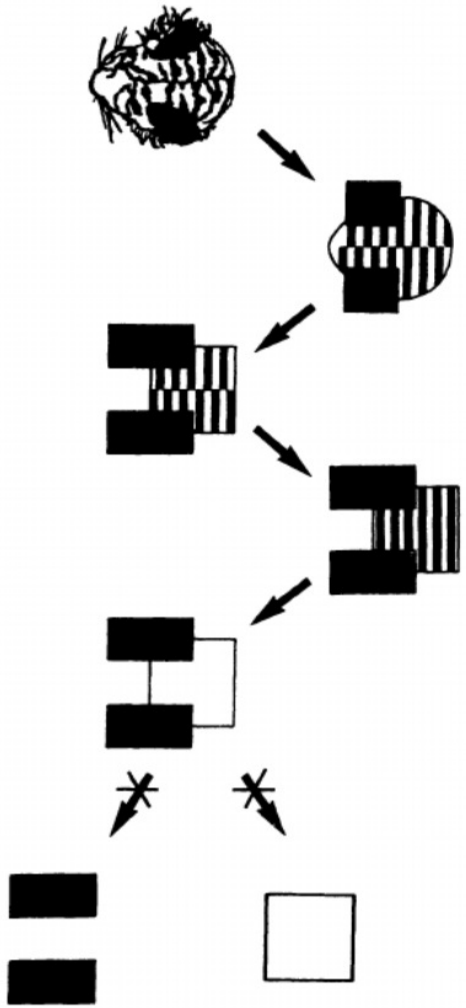


Fig. 1. An example of the procedure to determine the critical feature for activation of single cells: the gradual reduction of the complexity of the image of effective object stimuli. The substitution of intermediate features for the image of a tiger head, down to a combination of a white square with a pair of black rectangles, did not reduce the magnitude of the response. Further decomposition eliminated the response.

mode of the decrease in response was different when the stimulus was changed from the optimal stimulus.

We then made vertical and oblique penetrations through TE (*4*). The critical feature of a cell located at the middle of the penetration was first determined. A set of stimuli, including the optimal, suboptimal, and ineffective stimuli for the first tested cell, was made and then used to test the responsiveness of other cells recorded at different positions along the same penetration. Cells that responded to related stimuli in the stimulus set, that is, stimuli that were identical or similar to the optimal stimulus

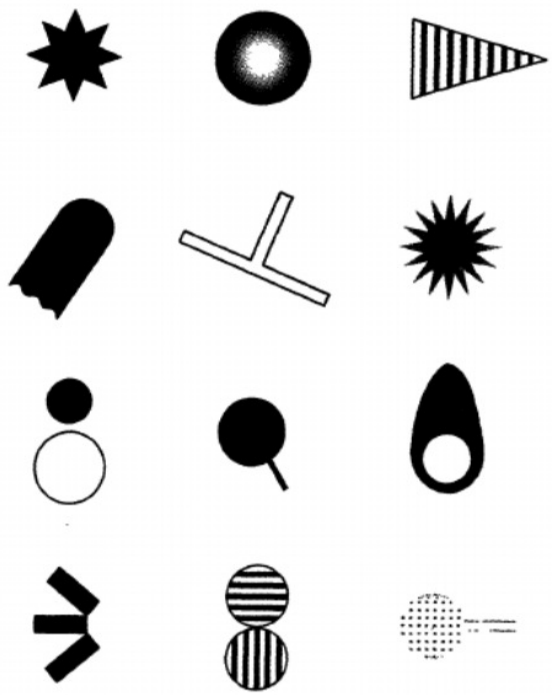


Fig. 2. Twelve examples of the critical features for the activation of single cells in area TE.

tle. Representation by multiple cells in a columnar module in which the selectivity varies from cell to cell and effective stimuli largely overlap can satisfy two apparently conflicting requirements in visual recognition: robustness to subtle changes in input images and precision of representation. Whereas the image of an object projected to the retina changes in response to variations in illumination, viewing angle, and articulation of the object, the global organization of outputs from TE changes little. The clustering of cells with overlapping and slightly different selectivity works as a buffer to absorb the changes. General advantages of the distributed representation have been extensively discussed elsewhere (*6*).

The representation by multiple cells with overlapping selectivity can be more precise than a mere summation of representation by individual cells. A similar argument has been made for hyperacuity (*7*). The position of the receptive fields changes gradually in the retina with a large overlap

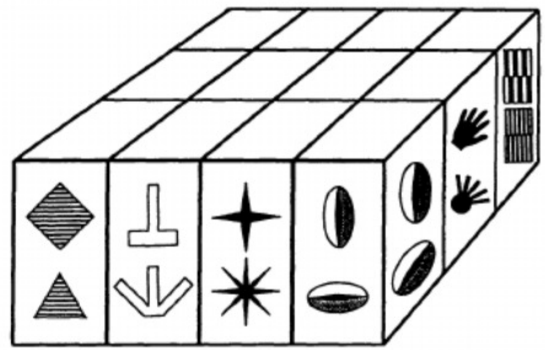


Fig. 3. Schematic diagram of the columnar organization in TE. Cells with similar but slightly different selectivity cluster in elongated vertical columns, perpendicular to the cortical surface.

RF of CNN Units: Minimal Image Representation

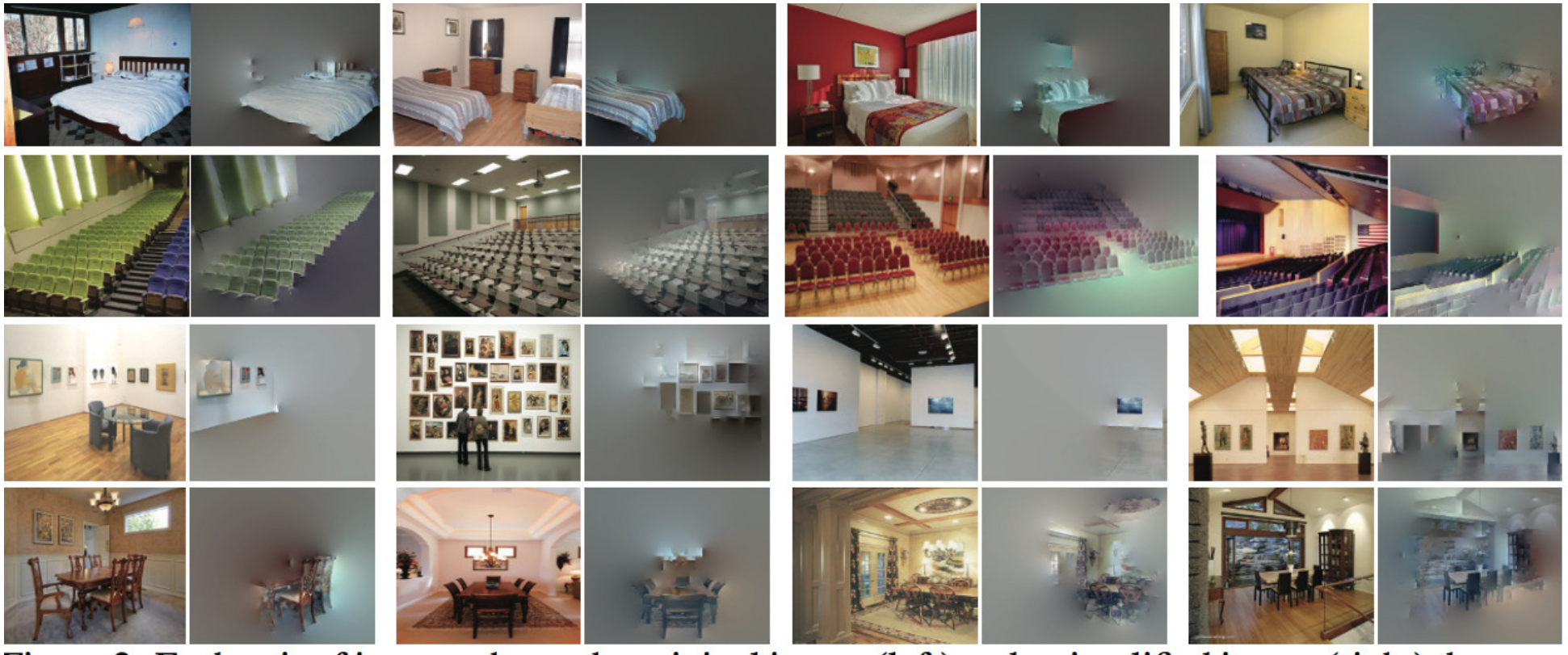


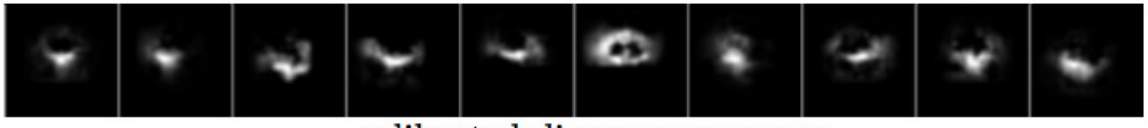
Figure 2: Each pair of images shows the original image (left) and a simplified image (right) that gets classified by the Places-CNN as the same scene category as the original image. From top to bottom, the four rows show different scene categories: bedroom, auditorium, art gallery, and dining room.

- *Object Detectors Emerge in Deep Scene CNNs*,
  Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva,
  Antonio Torralba, ICLR 2015.

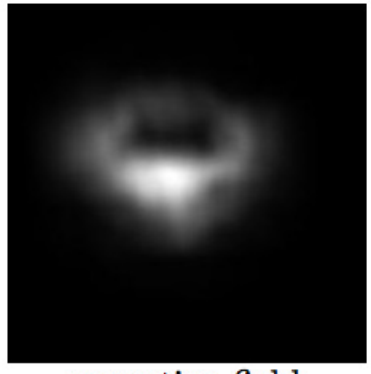
Mapping RF with Sliding-Window Discrepancy

sliding-window stimuli

discrepancy maps for top 10 images



calibrated discrepancy maps



receptive field

Figure 3: The pipeline for estimating the RF of each unit. Each sliding-window stimuli contains a small randomized patch (example indicated by red arrow) at different spatial locations. By comparing the activation response of the sliding-window stimuli with the activation response of the original image, we obtain a discrepancy map for each image (middle top). By summing up the calibrated discrepancy maps (middle bottom) for the top ranked images, we obtain the actual RF of that unit (right).

RFs for ImageNet-CNN and Places-CNN

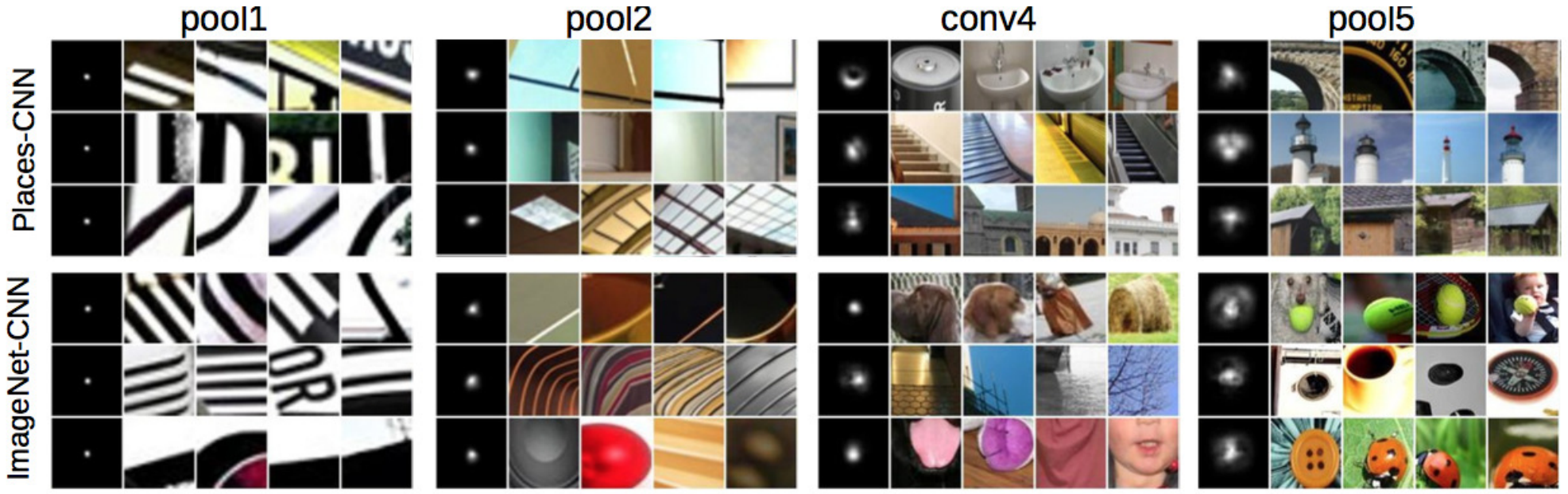


Figure 4: The RFs of 3 units of pool1, pool2, conv4, and pool5 layers respectively for ImageNet- and Places-CNNs, along with the image patches corresponding to the top activation regions inside the RFs.

Segmentation based on CNN RFs

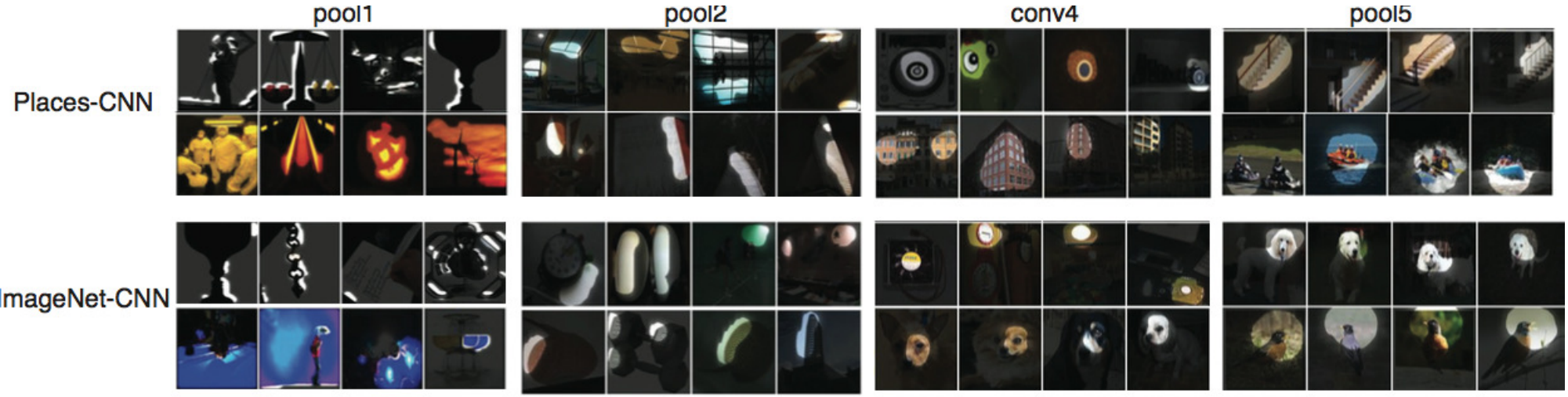


Figure 5: Segmentation based on RFs. Each row shows the 4 most confident images for some unit.

Object Detectors Emerge at Places-CNN

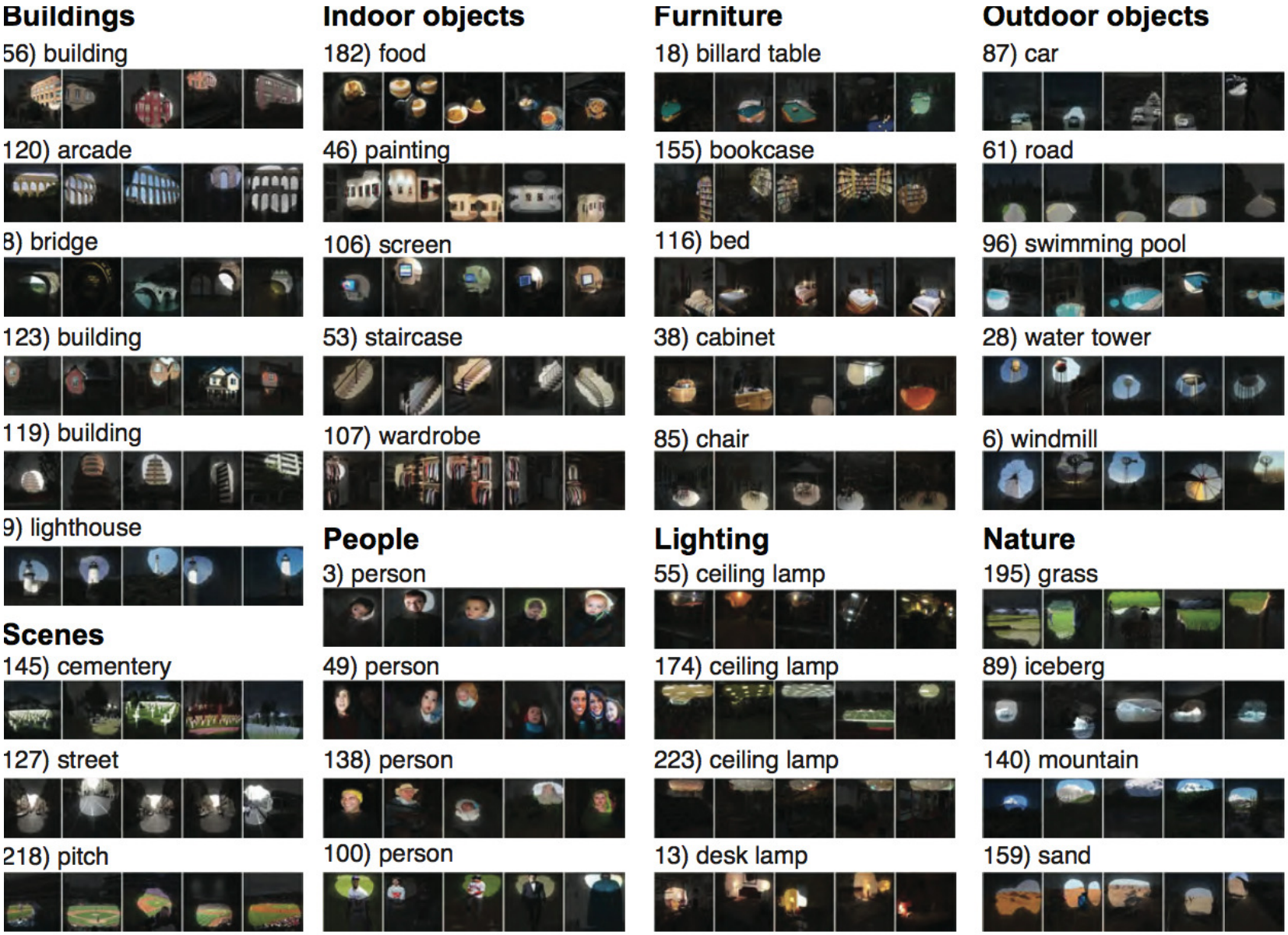


Figure 11: Segmentations using pool5 units from Places-CNN. Many classes are encoded by several units covering different object appearances. Each row shows the 5 most confident images for each unit. The number represents the unit number in pool5.

Inverting a CNN Representation: Code → Image

- Given a neural net f parameterized by θ
- Given the feature representation F_0 at a certain layer l,
- Seek an input image I such that

$$\min_I \underbrace{L(f_l(I;\theta), F_0)}_{\text{feature matching loss}} - \underbrace{\text{prior}(I)}_{\text{admissible image set}} \tag{19}$$

- Optimizing the input with the network weights fixed

Invert Features at Different Depth Layers

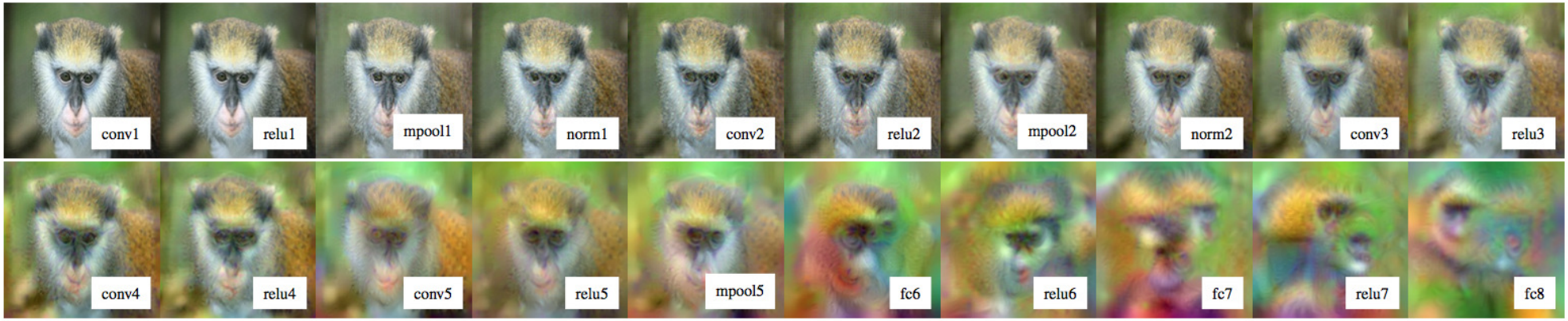


Figure 6. **CNN reconstruction.** Reconstruction of the image of Fig. 5.a from each layer of CNN-A. To generate these results, the regularization coefficient for each layer is chosen to match the highlighted rows in table 3. This figure is best viewed in color/screen.

- *Understanding Deep Image Representations by Inverting Them*, Aravindh Mahendran, Andrea Vedaldi, CVPR 2015.

Invert Local Features at Different Depth Layers

Figure 9. **CNN receptive field.** Reconstructions of the image of Fig. 5.a from the central 5×5 neuron fields at different depths of CNN-A. The white box marks the field of view of the 5×5 neuron field. The field of view is the entire image for conv5 and relu5.

Invert Feature Subsets at Different Depth Layers

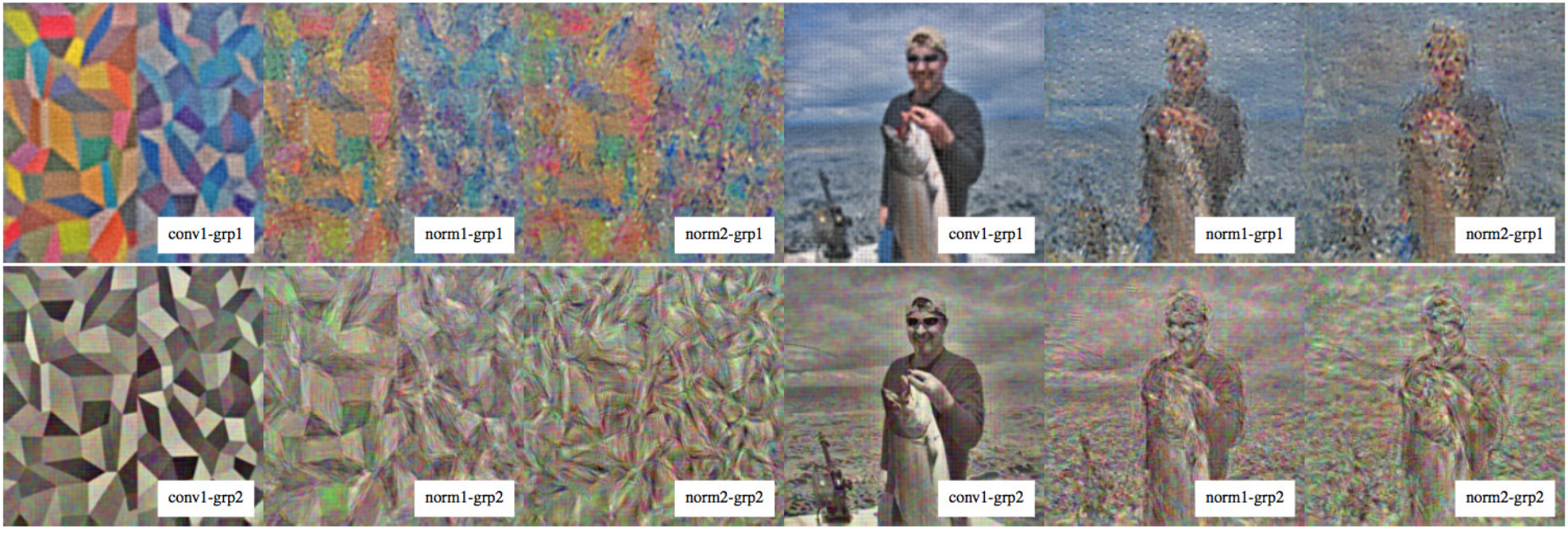


Figure 10. **CNN neural streams.** Reconstructions of the images of Fig. 5.c-b from either of the two neural streams of CNN-A. This figure is best seen in colour/screen.

Learn to Invert Features with A CNN

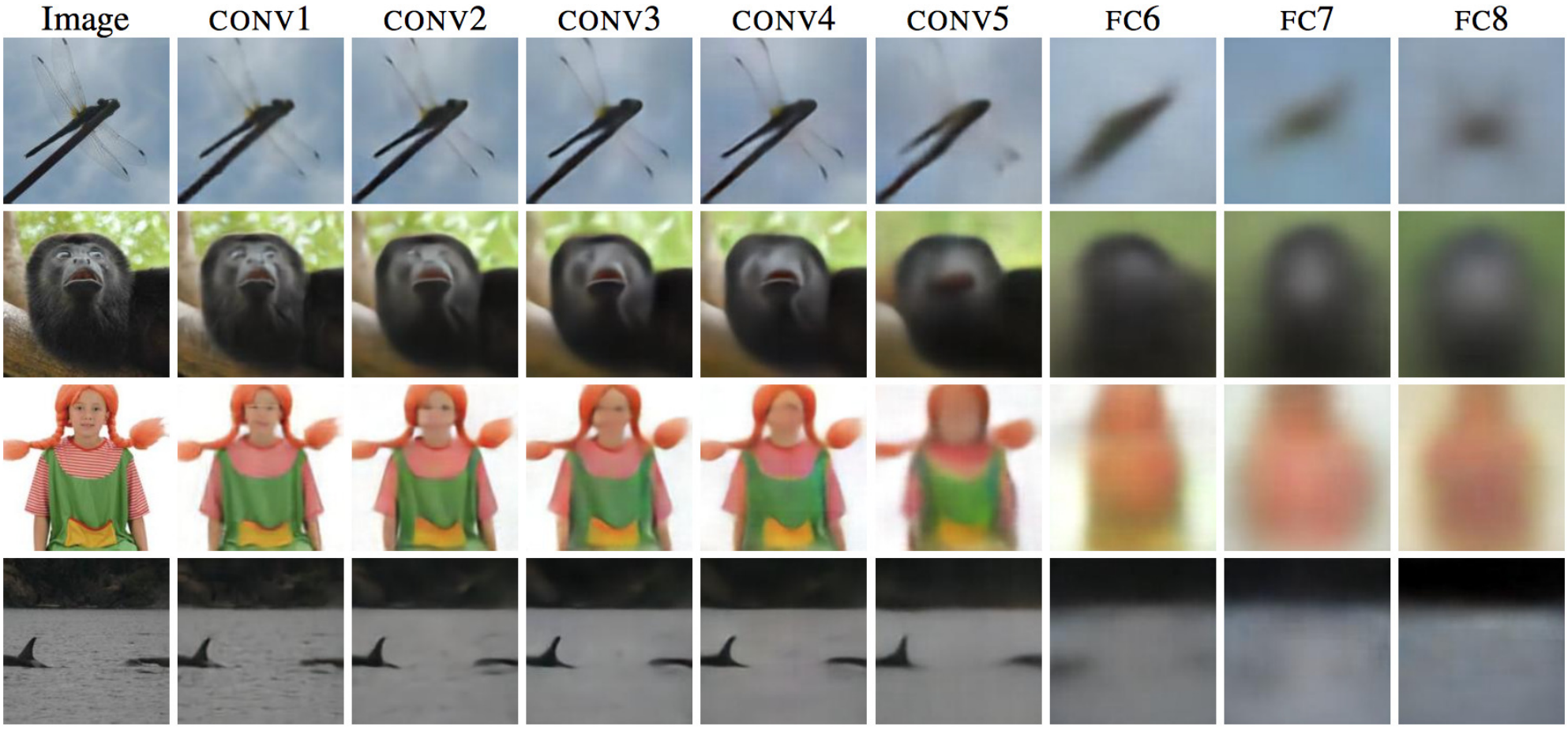


Figure 5: Reconstructions from different layers of AlexNet.

- *Inverting Visual Representations with Convolutional Networks*, Alexey Dosovitskiy, Thomas Brox, CVPR 2016.

Learn to Invert Features with An Image Generator

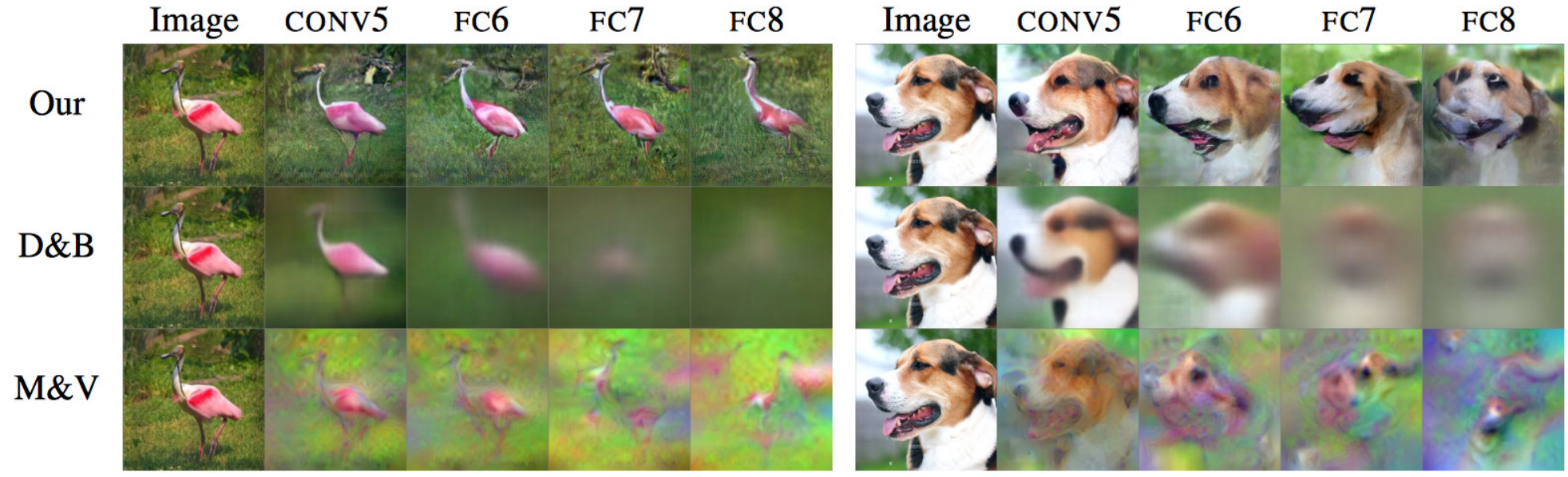


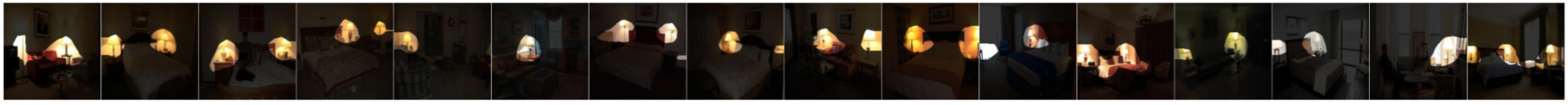
Figure 4: AlexNet inversion: comparison with Dosovitskiy and Brox [26] and Mahendran and Vedaldi [21] . Our results are significantly better, even our failure cases (second image).

- *Generating Images with Perceptual Similarity Metrics based on Deep Networks*, Alexey Dosovitskiy, Thomas Brox, NIPS 2016.

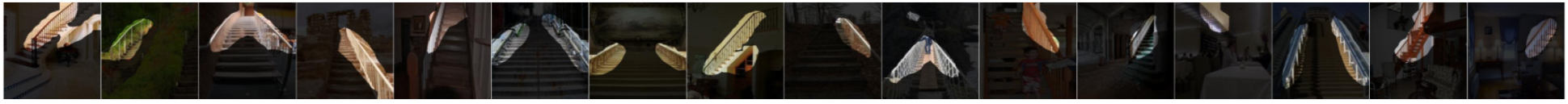
Interpretable RF: Align RF with Semantic Concepts

AlexNet-Places205 conv5 unit 138: heads

AlexNet-Places205 conv5 unit 215: castles



AlexNet-Places205 conv5 unit 13: lamps

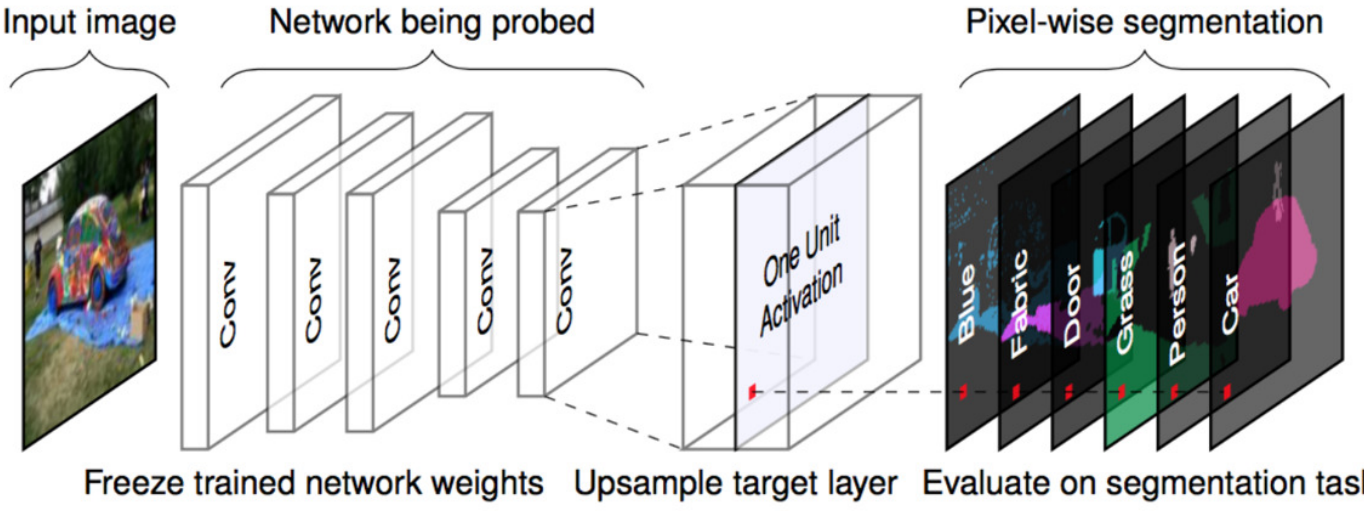


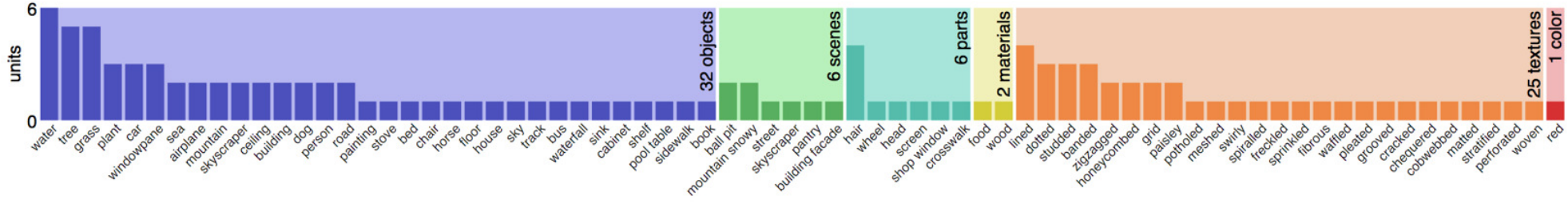
AlexNet-Places205 conv5 unit 53: stairways

- *Network Dissection: Quantifying Interpretability of Deep Visual Representations*, David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, Antonio Torralba, CVPR 2017.

Network Dissection

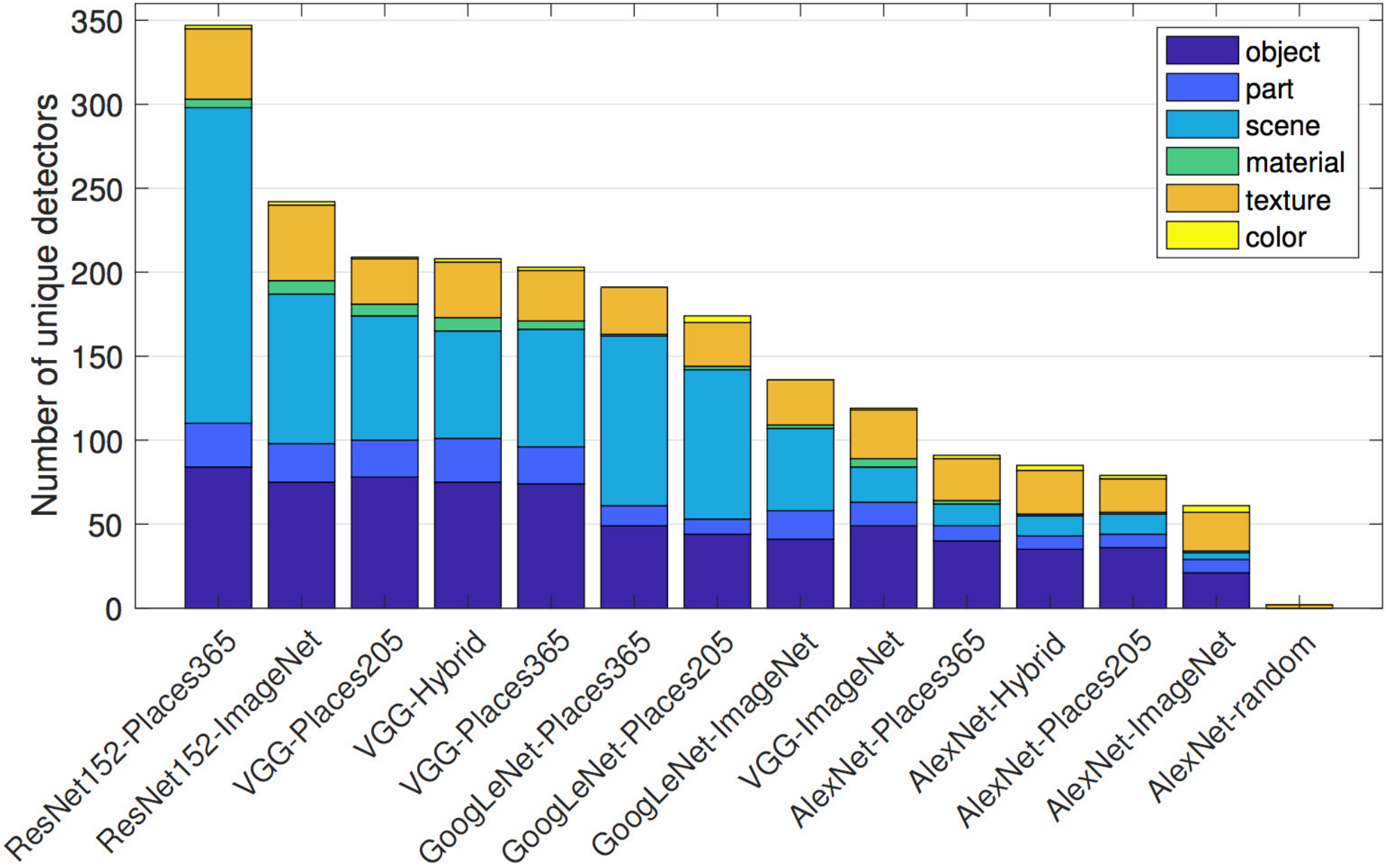
Network Dissection is our method for quantifying interpretability of individual units in a deep CNN (i.e., our answer to question #1). It works by measuring the alignment between unit response and a set of concepts drawn from a broad and dense segmentation data set called Broden.





By measuring the concept that best matches each unit, Net Dissection can break down the types of concepts represented in a layer: here the 256 units of AlexNet conv5 trained on Places represent many objects and textures, as well as some scenes, parts, materials, and a color.

Number of Unique Detectors w/ CNN Architectures



Model Fine-tuning: Dog → Waterfall

Summary

1. Direct visualization is only useful for earlier layers.

2. Reconstruction by deconvolution with guided backprop shows RF of positive contributions.

3. Activation maximization synthesizes prototypical images for a neuron, artificial or natural, single or multifaceted, depending on image priors and initializations for optimization.

4. Saliency maps and occlusion heatmaps provide RF localization and object segmentation.

5. Code inversion with generators shows optimal natural stimuli for a particular feature response pattern.

6. Aligning CNN RFs to segmentation annotated datasets provides network dissection in terms of semantic interpretations.