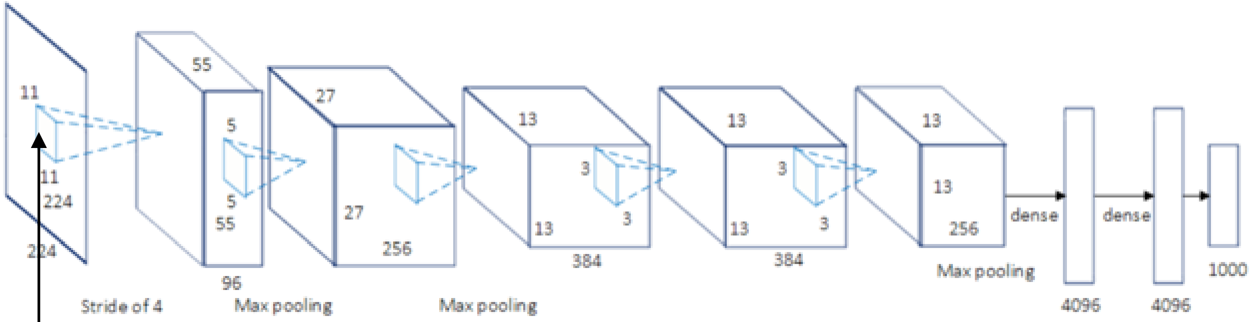
Visualizing and explaining neural networks

https://deepdreamgenerator.com/

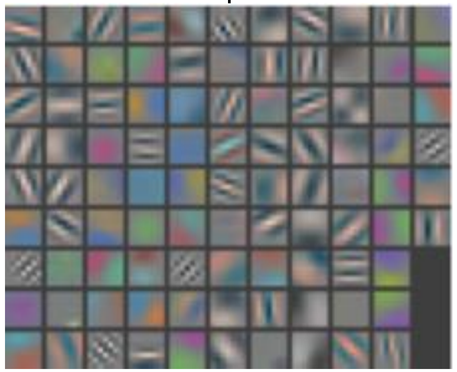
Outline

- Basic visualization techniques
- Mapping activations back to the image
- Synthesizing images to maximize activation
- Saliency maps
- Quantifying interpretability of units

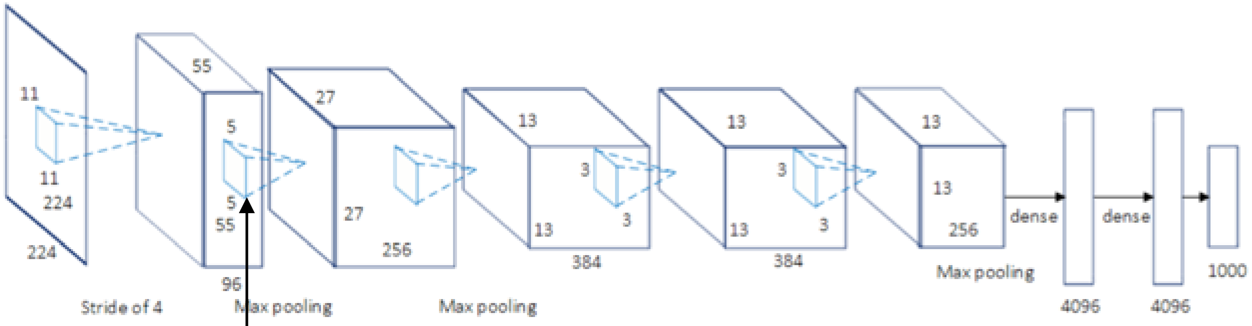
Overview and basic visualization techniques



Visualize first-layer
weights directly



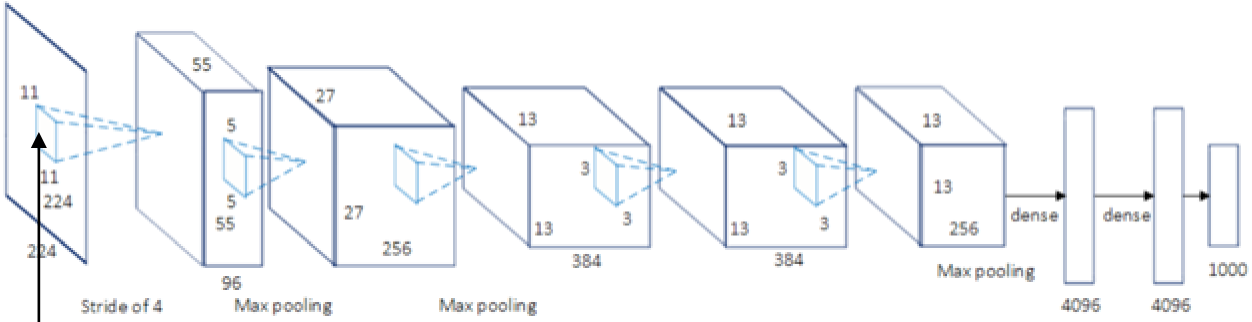
Overview and basic visualization techniques



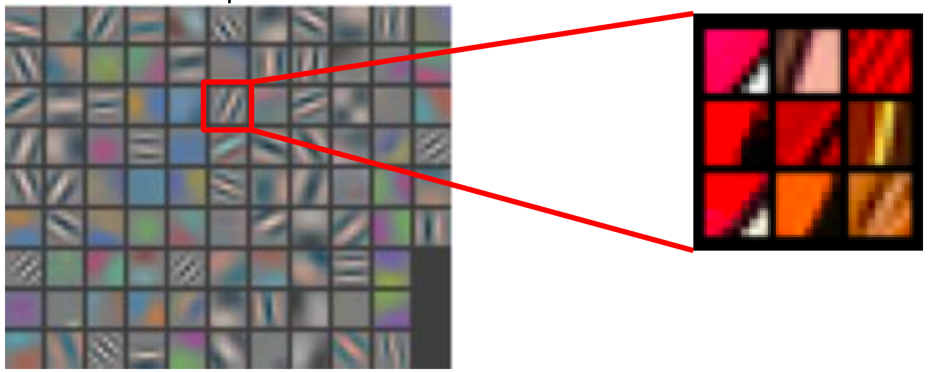
Not too helpful for
subsequent layers

Features from a CIFAR10 network, via [Stanford CS231n]

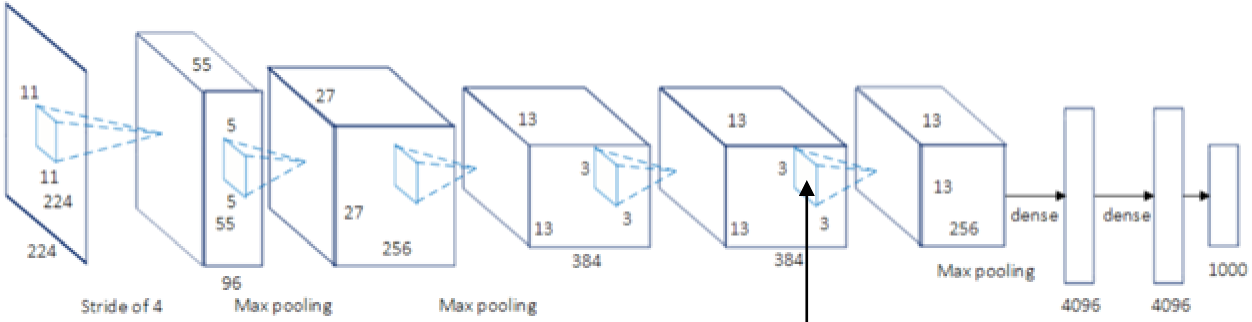
Overview and basic visualization techniques



Visualize maximally activating patches: pick a unit; run many images through the network; visualize patches that produce the highest output values

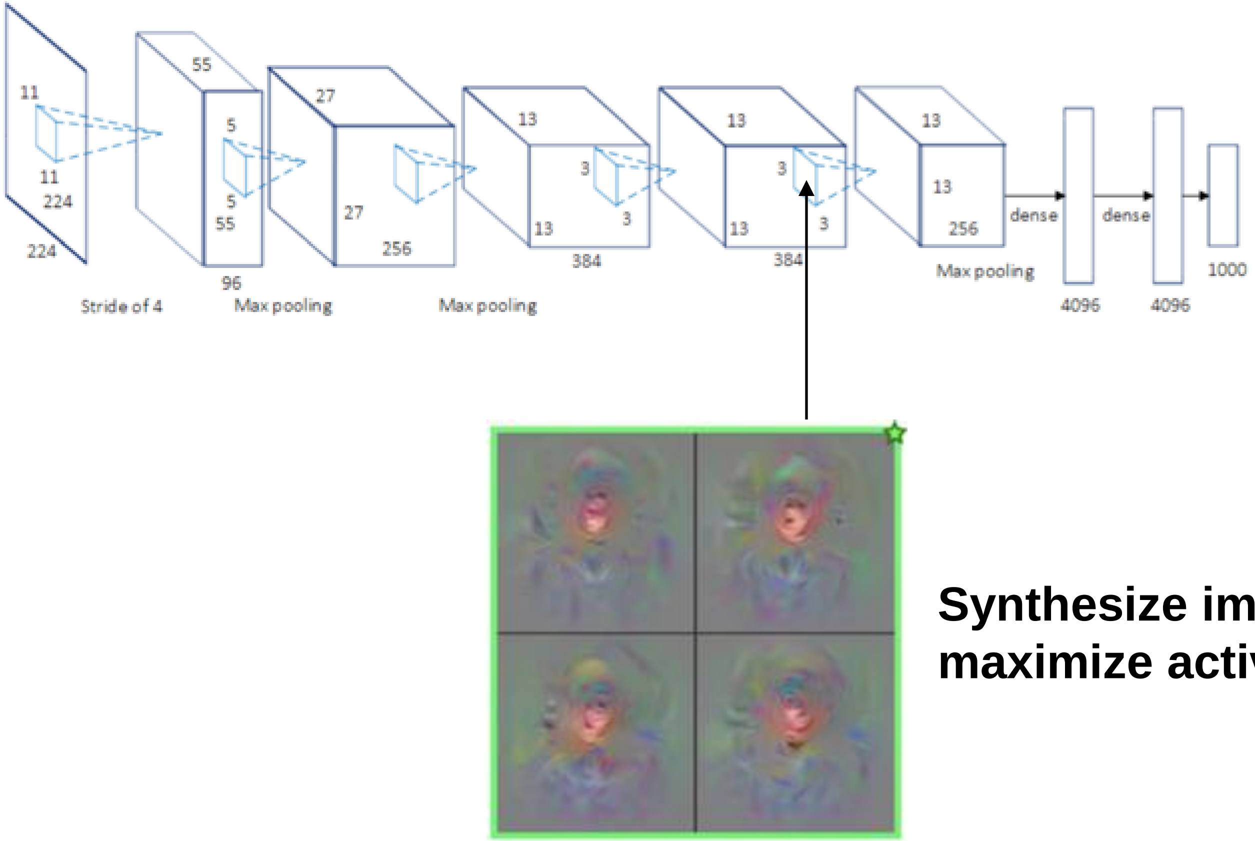


Overview and basic visualization techniques



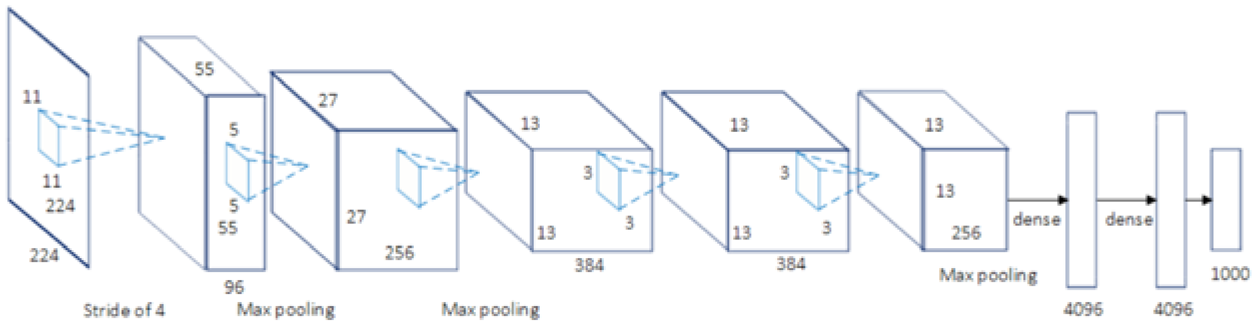
Visualize maximally activating patches

Overview and basic visualization techniques



Synthesize images to maximize activation

Overview and basic visualization techniques

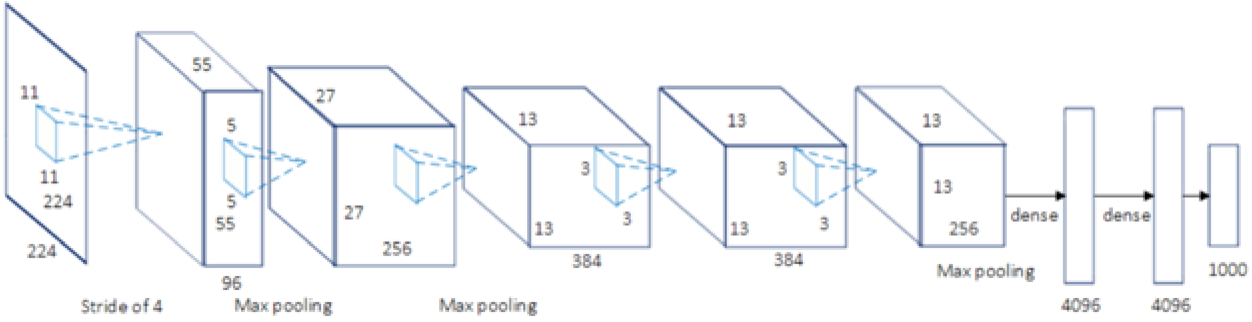


What about FC layers?

Visualize nearest neighbor images according to activation vectors

Source: [Stanford CS231n]

Overview and basic visualization techniques



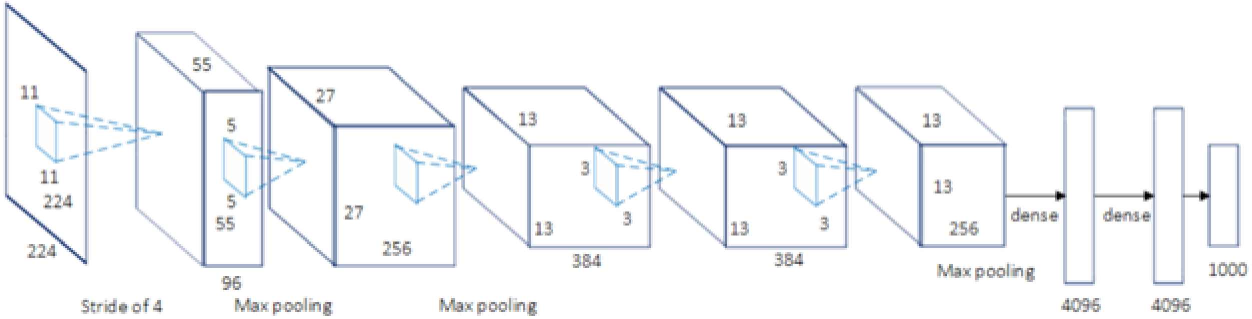
What about FC layers?

Fancy dimensionality reduction, e.g., [t-SNE]

Source: [Andrej Karpathy]

Overview and basic visualization techniques

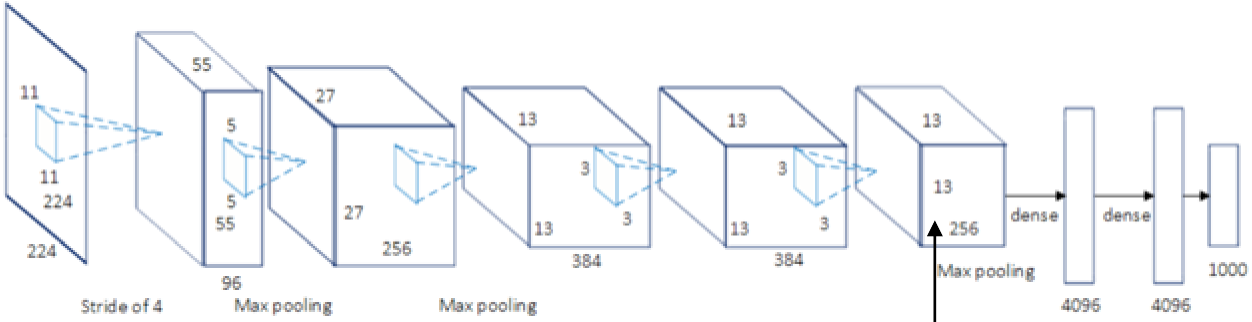
Given: a particular input image



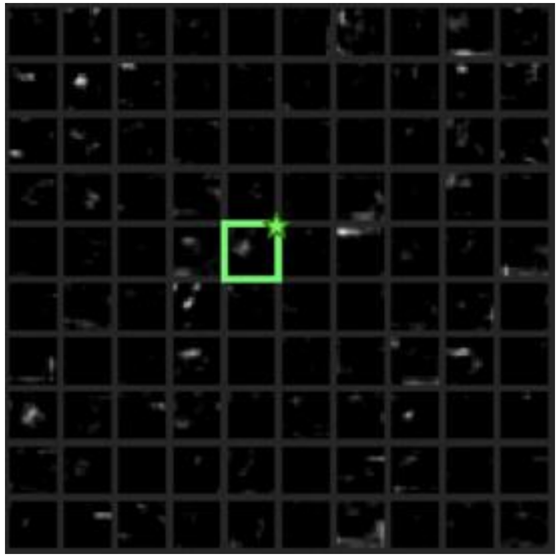
“cat”

Overview and basic visualization techniques

Given: a particular input image



“cat”

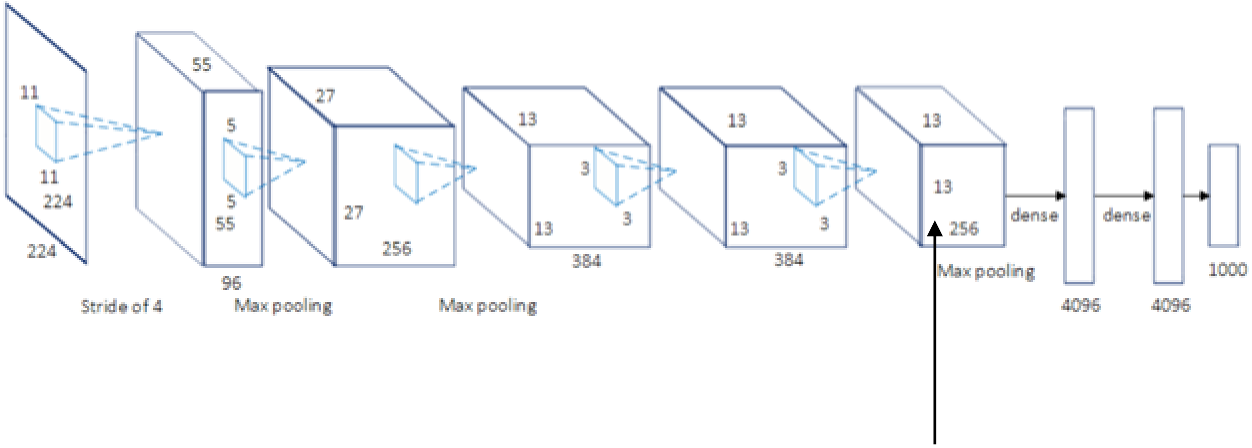


Source

Visualize activations for this image

Overview and basic visualization techniques

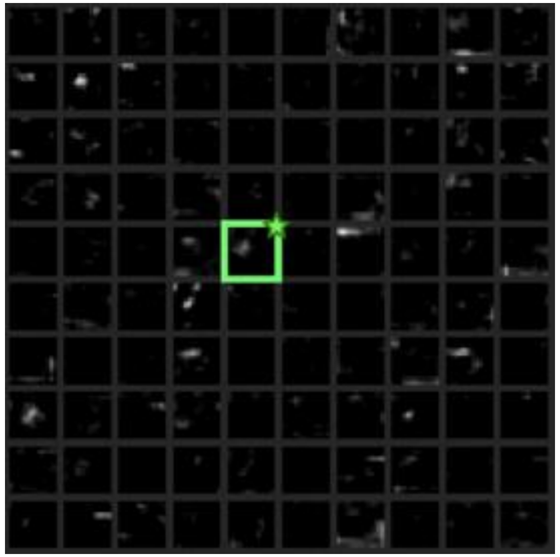
Given: a particular input image



"cat"

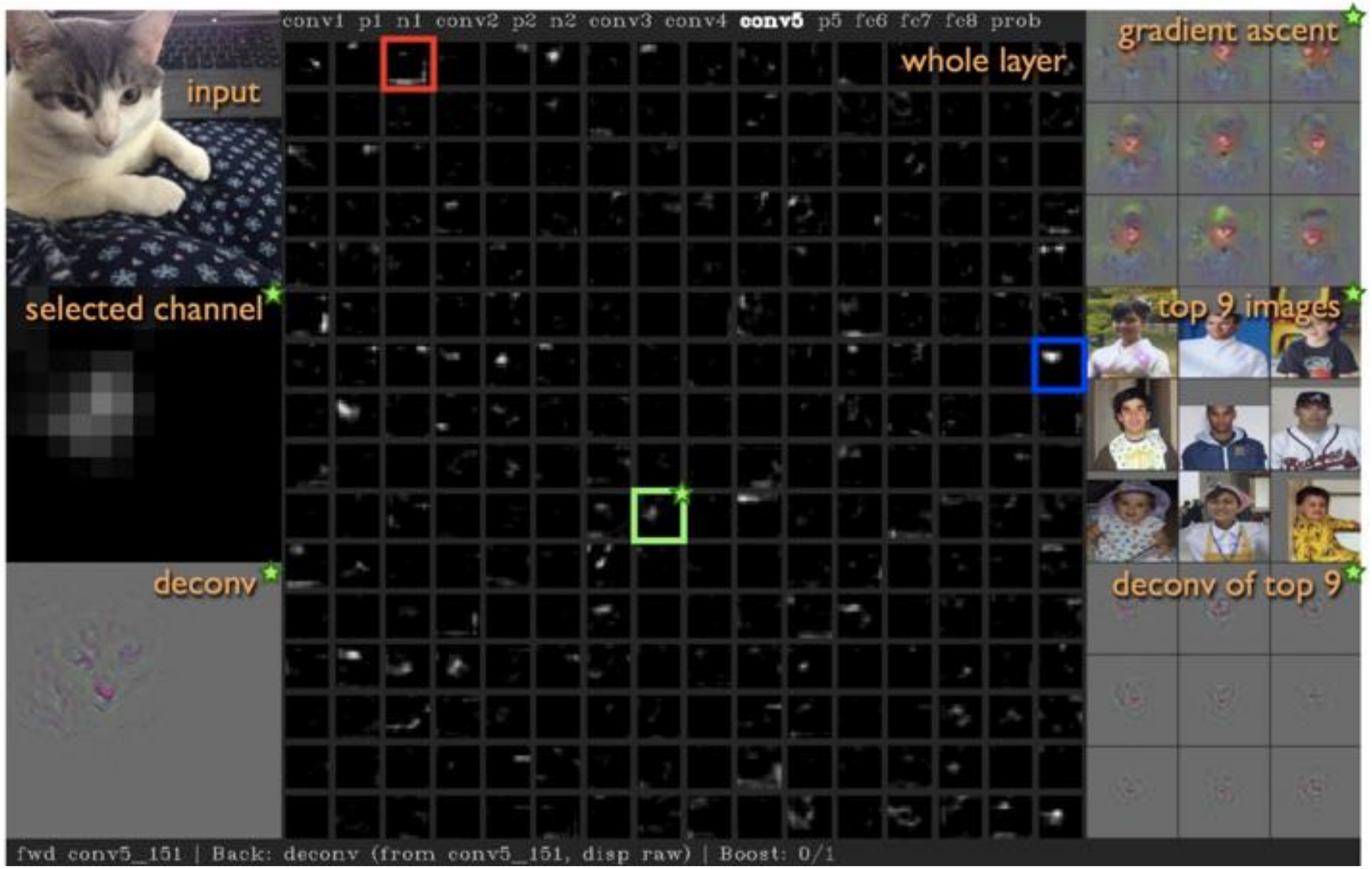
Visualize activations for this image

Visualize pixel values responsible for the activation



[Source]

Deep visualization toolbox



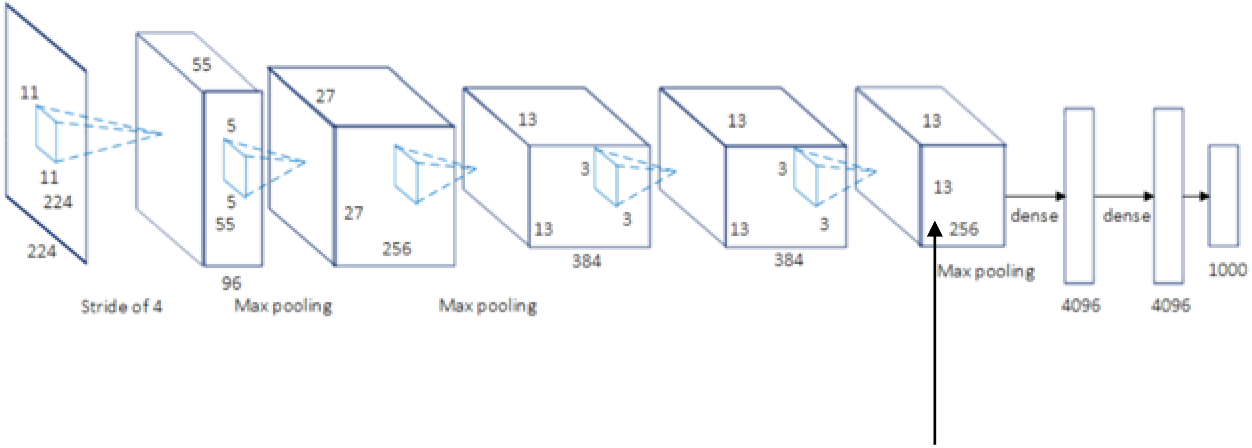
[YouTube video]

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, [Understanding neural networks through deep visualization], ICML DL workshop, 2015

Outline

- Basic visualization techniques
- Mapping activations back to the image
- Synthesizing images to maximize activation
- Saliency maps
- Quantifying interpretability of units

Mapping activations back to pixels



"cat"

- Let's take a single value in an intermediate feature map and propagate its gradient back to the original image pixels
- What does this tell us?

Mapping activations back to pixels

1. Forward an image through the network
2. Choose a feature map and an activation
3. Zero out all values except for the one of interest
4. Propagate that value back to the image

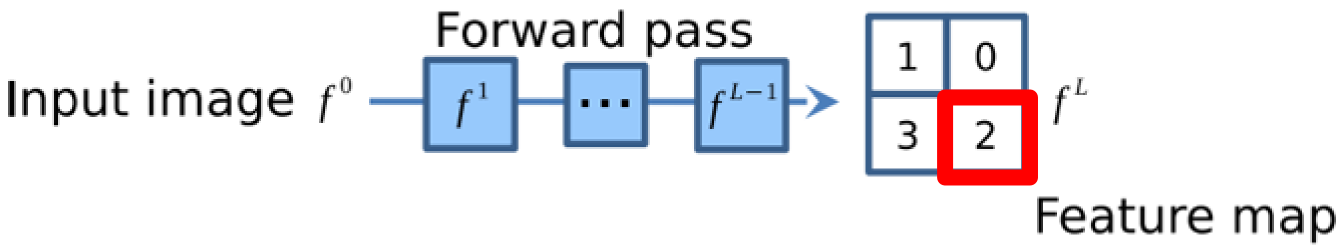
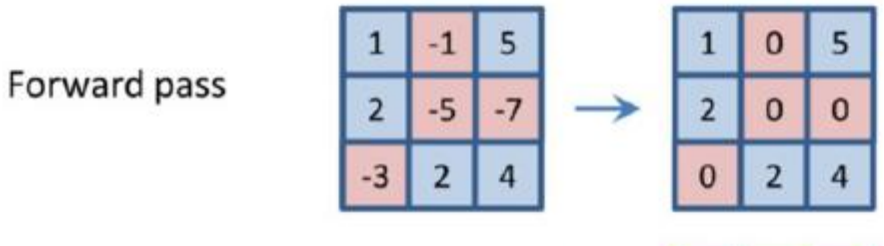


Figure source

Mapping activations back to pixels

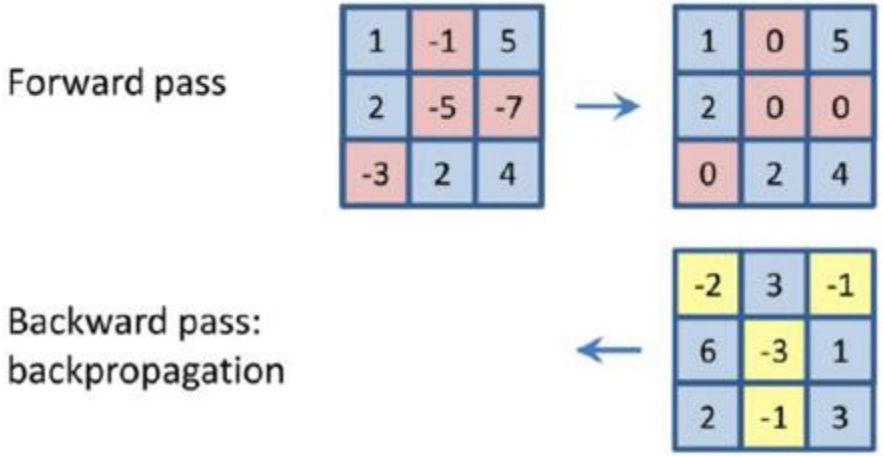
- Commonly used methods differ in how they treat the ReLU



J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, ICLR workshop, 2015

Mapping activations back to pixels

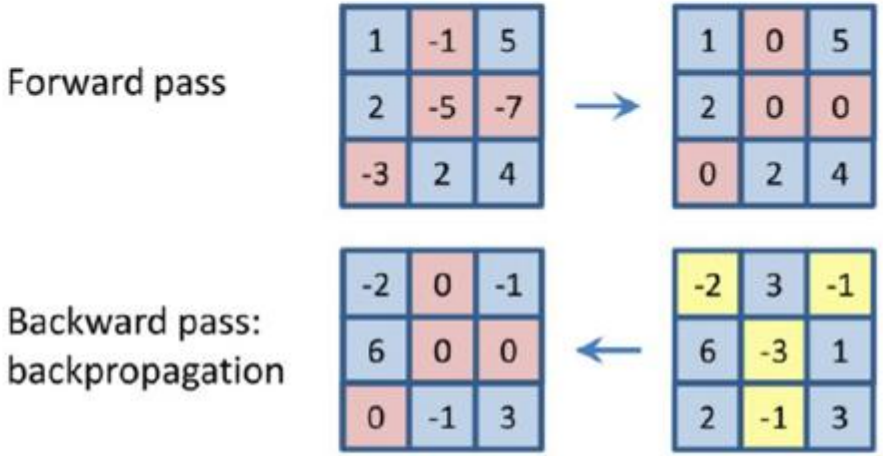
- Commonly used methods differ in how they treat the ReLU



J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, [Striving for simplicity: The all convolutional net](), ICLR workshop, 2015

Mapping activations back to pixels

- Commonly used methods differ in how they treat the ReLU

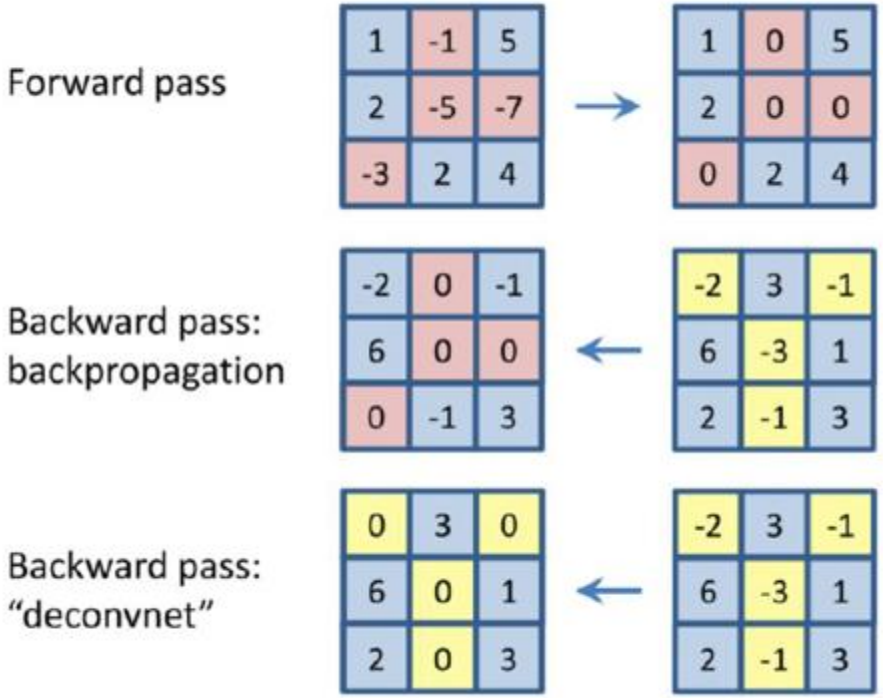


Propagating back negative gradients bad for visualization

J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, ICLR workshop, 2015

Mapping activations back to pixels

- Commonly used methods differ in how they treat the ReLU



Propagating back negative gradients bad for visualization

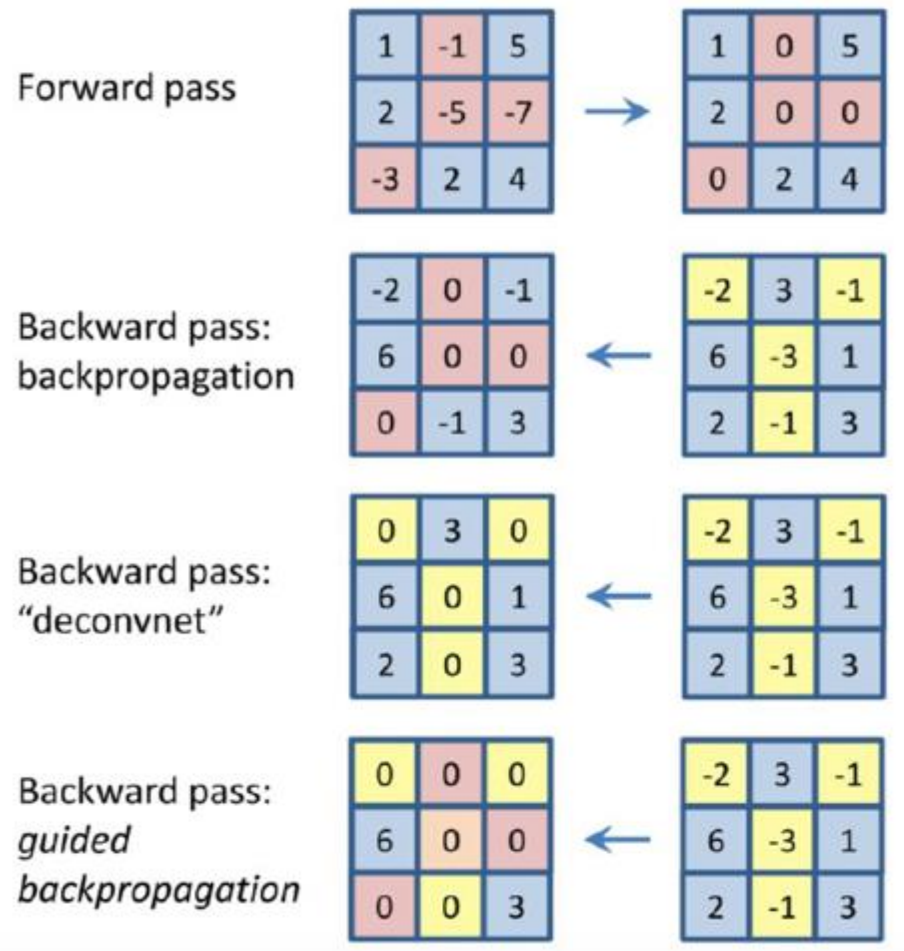
Zeiler and Fergus (2014)

Does not work well for architectures w/o max pooling, for high layers

J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, ICLR workshop, 2015

Mapping activations back to pixels

- Commonly used methods differ in how they treat the ReLU



Propagating back negative gradients bad for visualization

Zeiler and Fergus (2014)

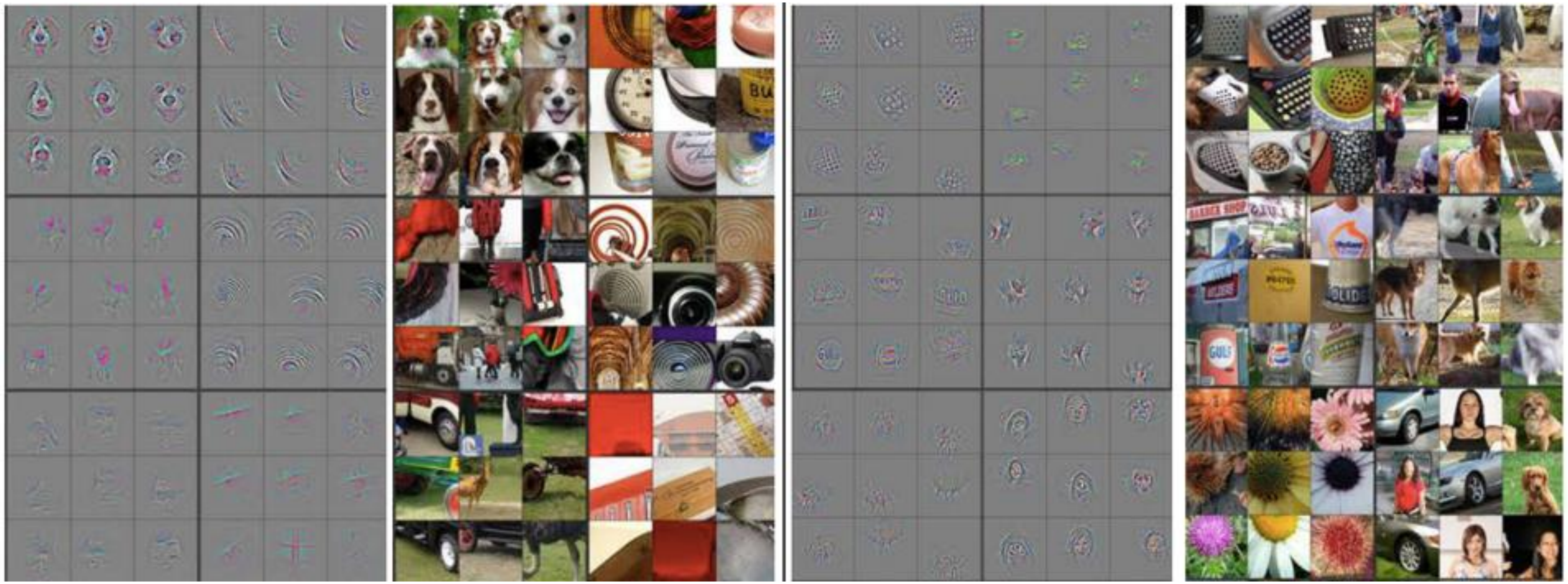
Does not work well for architectures w/o max pooling, for high layers

Springenberg et al. (2015)

J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, Striving for simplicity: The all convolutional net, ICLR workshop, 2015

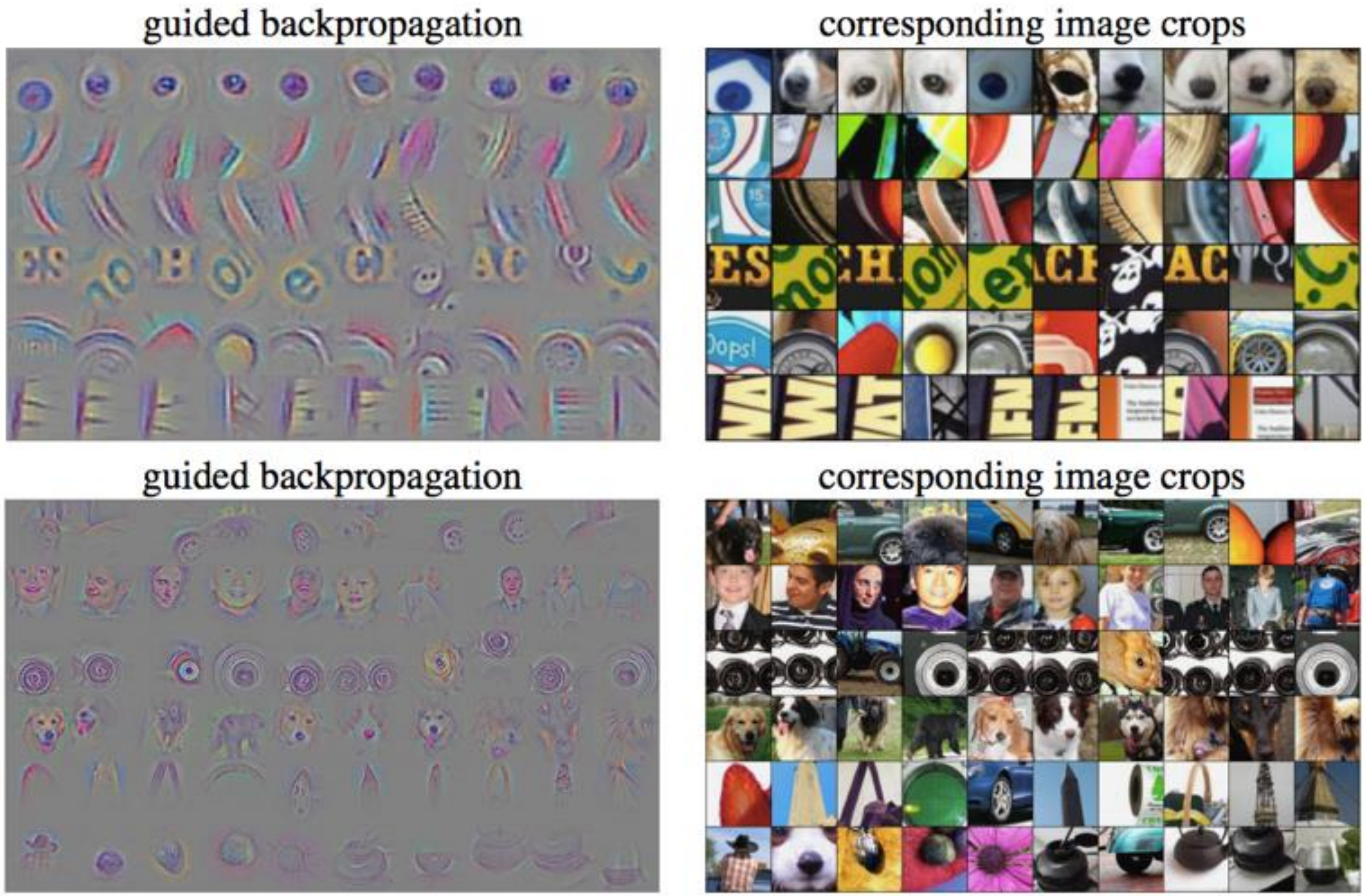
Deconvnet visualization

AlexNet Layer 4 AlexNet Layer 5



M. Zeiler and R. Fergus, [Visualizing and Understanding Convolutional Networks](), ECCV 2014

Guided backpropagation visualization



J. Springenberg, A. Dosovitskiy, T. Brox, M. Riedmiller, [Striving for simplicity: The all convolutional net](), ICLR workshop, 2015

Outline

- Basic visualization techniques
- Mapping activations back to the image
- Synthesizing images to maximize activation
- Saliency maps
- Quantifying interpretability of units

Visualization by optimization (model inversion)

- How can we synthesize images that maximize activation of a given neuron?
- Basic approach: find image x maximizing target activation $f(x)$ subject to *natural image regularization penalty* $R(x)$:

$$x^* = \arg\max_x f(x) - \lambda R(x)$$

Visualization by optimization (model inversion)

- Maximize $f(x) - \lambda R(x)$
  - $f(x)$ is score for a category *before softmax*
  - $R(x)$ is L2 regularization
  - Perform *gradient ascent* starting with zero image, add dataset mean to result

dumbbell **cup** **dalmatian**

K. Simonyan, A. Vedaldi, and A. Zisserman, [Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps](), ICLR 2014

Visualization by optimization (model inversion)

- Alternative approach to regularization:
  at each step of gradient ascent, apply operator r that regularizes the image:

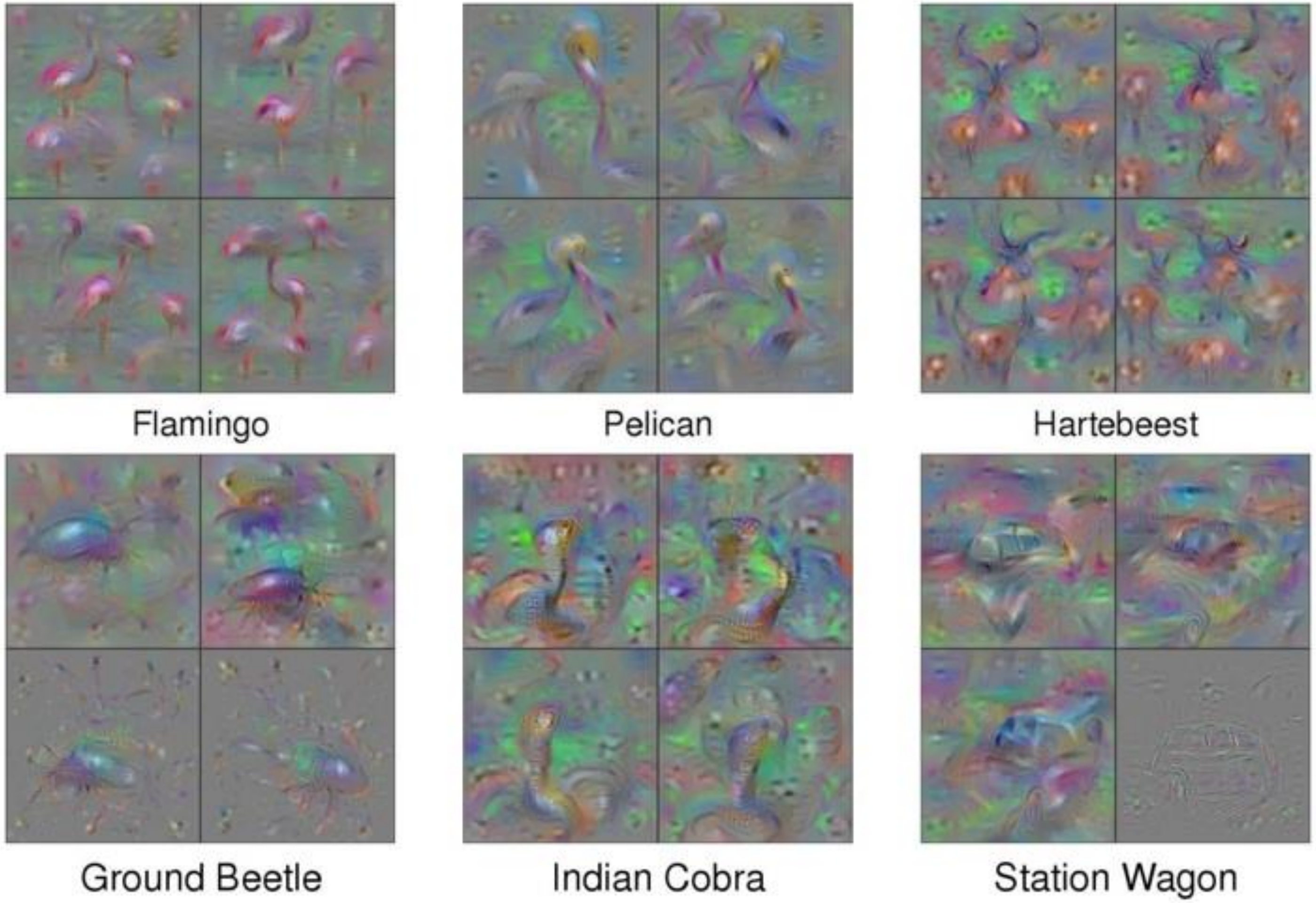
$$x \leftarrow r\left(x + \eta \frac{\partial f}{\partial x}\right)$$

- Combination that gives good-looking results:
  - L2 decay
  - Gaussian blur (every few iterations)
  - Clip pixel values with small magnitude
  - Clip pixel values with small contribution to the activation (estimated by product of pixel value and gradient)

J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, [Understanding neural networks through deep visualization](), ICML DL workshop, 2015

Visualization by optimization (model inversion)

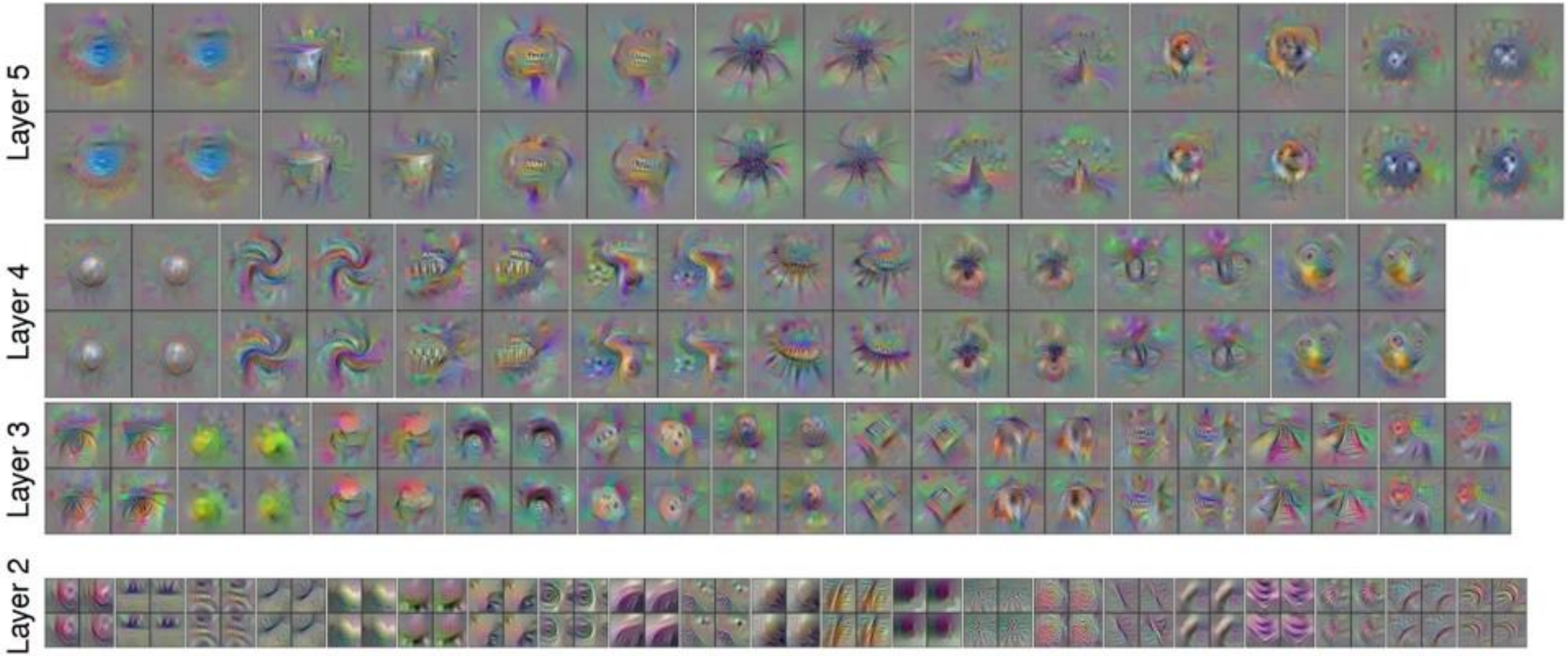
- Example visualizations:



J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, [Understanding neural networks through deep visualization](), ICML DL workshop, 2015

Visualization by optimization (model inversion)

- Example visualizations of intermediate features:



J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, [Understanding neural networks through deep visualization](), ICML DL workshop, 2015

Multifaceted feature visualization

- Key idea: most neurons in high layers respond to a mix of different patterns or “facets”
- For coherent visualizations, zero in on individual facets

A. Nguyen, J. Yosinski, J. Clune, [Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks](), ICML workshop, 2016

Multifaceted feature visualization

- Key idea: most neurons in high layers respond to a mix of different patterns or "facets"
- For coherent visualizations, zero in on individual facets

- Algorithm:
    - Cluster FC activations of training images to identify facets
    - For each facet, initialize optimization with mean image of that facet
    - To attempt to produce image of a single object, use *center-biased regularization* (start with blurry image, gradually increase resolution and update center pixels more than edge pixels)

A. Nguyen, J. Yosinski, J. Clune, Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, ICML workshop, 2016

Multifaceted feature visualization

A. Nguyen, J. Yosinski, J. Clune, Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks, ICML workshop, 2016

Google DeepDream

Choose an image and a layer in a CNN; repeat:

1. Forward: compute activations at chosen layer
2. Set gradient of chosen layer *equal to its activation*
   - Equivalent to maximizing $\sum_i f_i^2(x)$
3. Backward: Compute gradient w.r.t. image
4. Update image (with some tricks)

Source: [Stanford CS231n]

https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

https://deepdreamgenerator.com/

Outline

- Basic visualization techniques
- Mapping activations back to the image
- Synthesizing images to maximize activation
- Saliency maps
- Quantifying interpretability of units

Saliency maps

- Which parts of the image played the most important role in the network's decision?

Prediction: "car" 64%

Source: K. Saenko

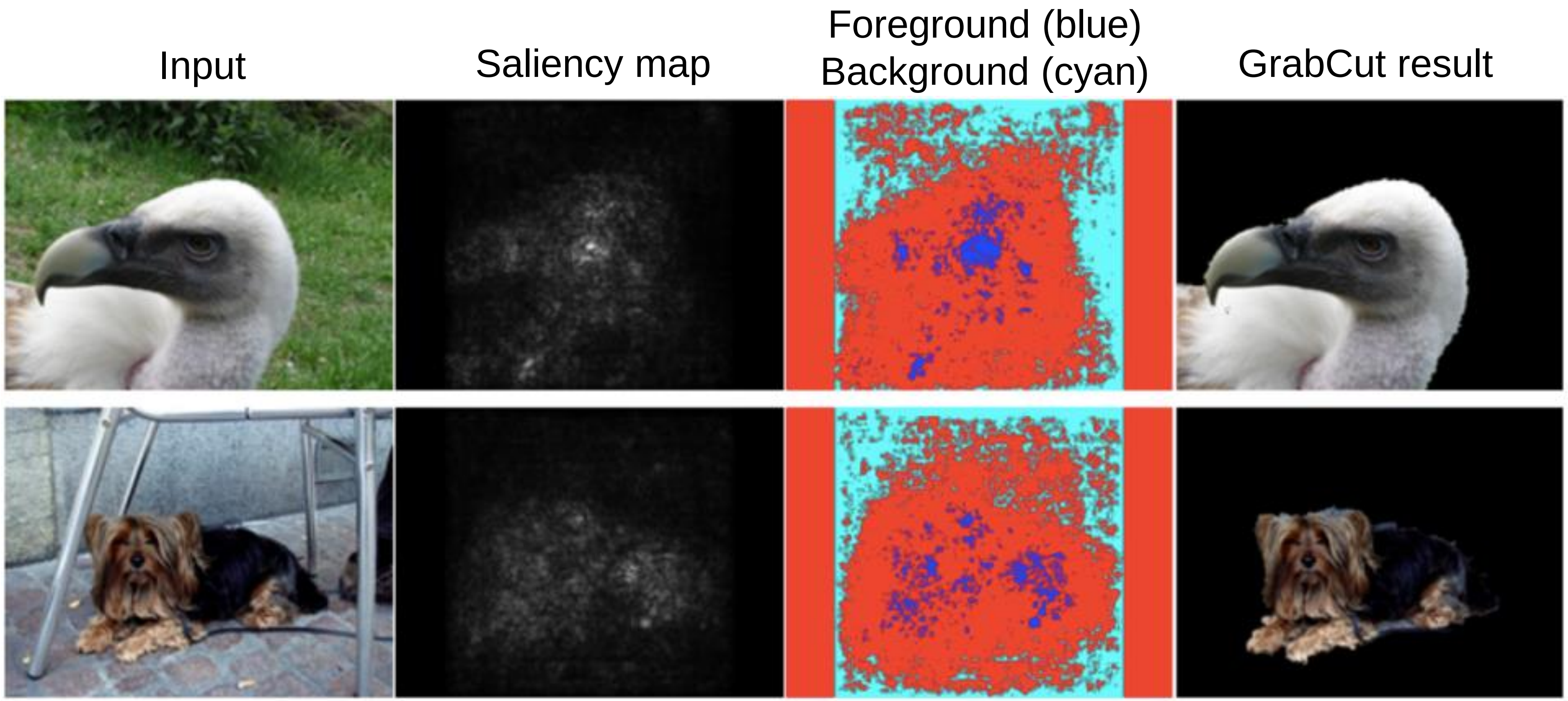
“White box” saliency via gradients

- Backpropagate gradient of class score (before softmax) to the image, display max of absolute values across color channels

K. Simonyan, A. Vedaldi, and A. Zisserman, [Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps](), ICLR 2014

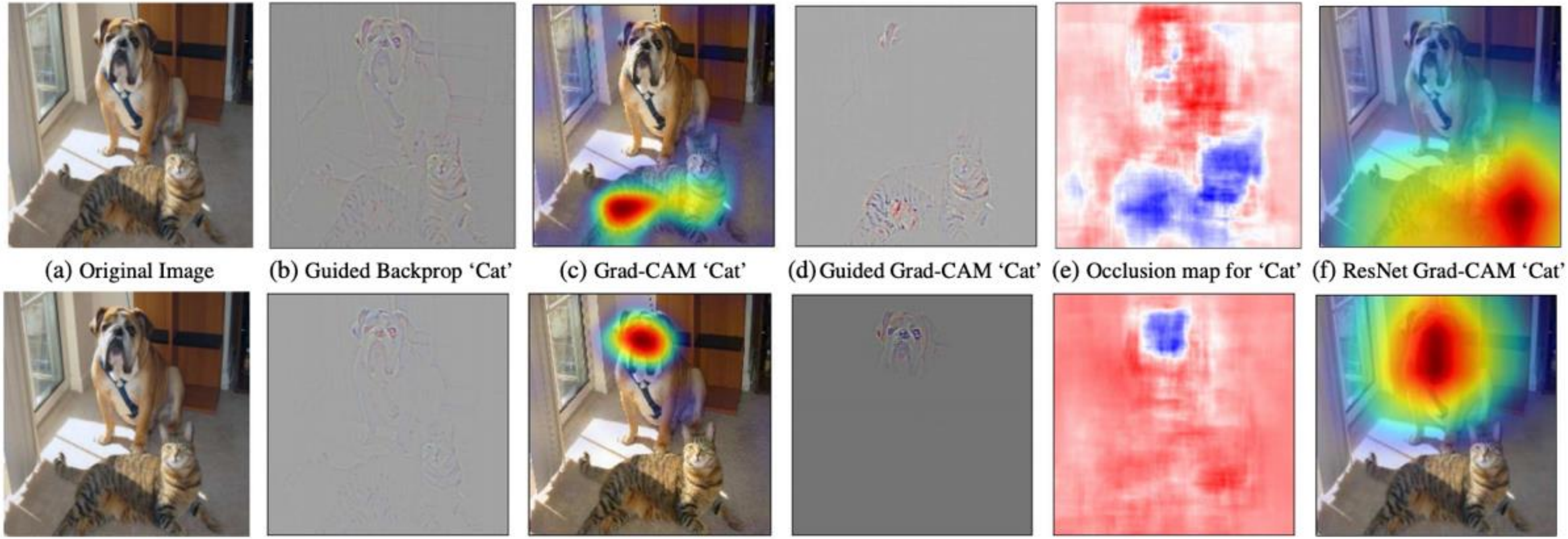
“White box” saliency via gradients

- Can be used for *weakly supervised* segmentation:



K. Simonyan, A. Vedaldi, and A. Zisserman, [Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps](), ICLR 2014

Gradient-weighted class activation mapping (Grad-CAM)



(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d) Guided Grad-CAM 'Cat' (e) Occlusion map for 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j) Guided Grad-CAM 'Dog' (k) Occlusion map for 'Dog' (l) ResNet Grad-CAM 'Dog'

R. Selvaraju et al. [Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization](). ICCV 2017

“Black box” saliency via masking

- Slide square occluder across image, see how class score changes

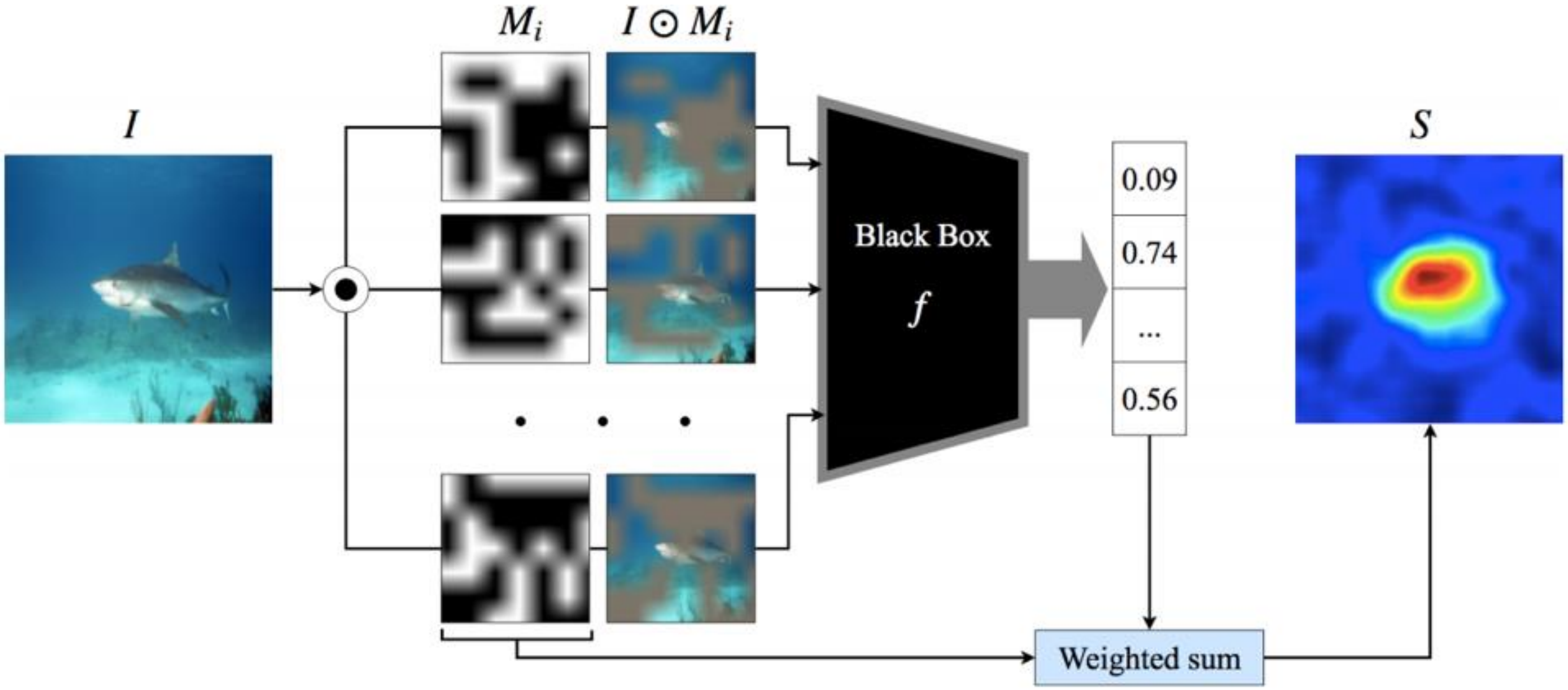
Input image

Correct class probability as function of occluder position

M. Zeiler and R. Fergus, [Visualizing and Understanding Convolutional Networks](), ECCV 2014

“Black box” saliency via masking

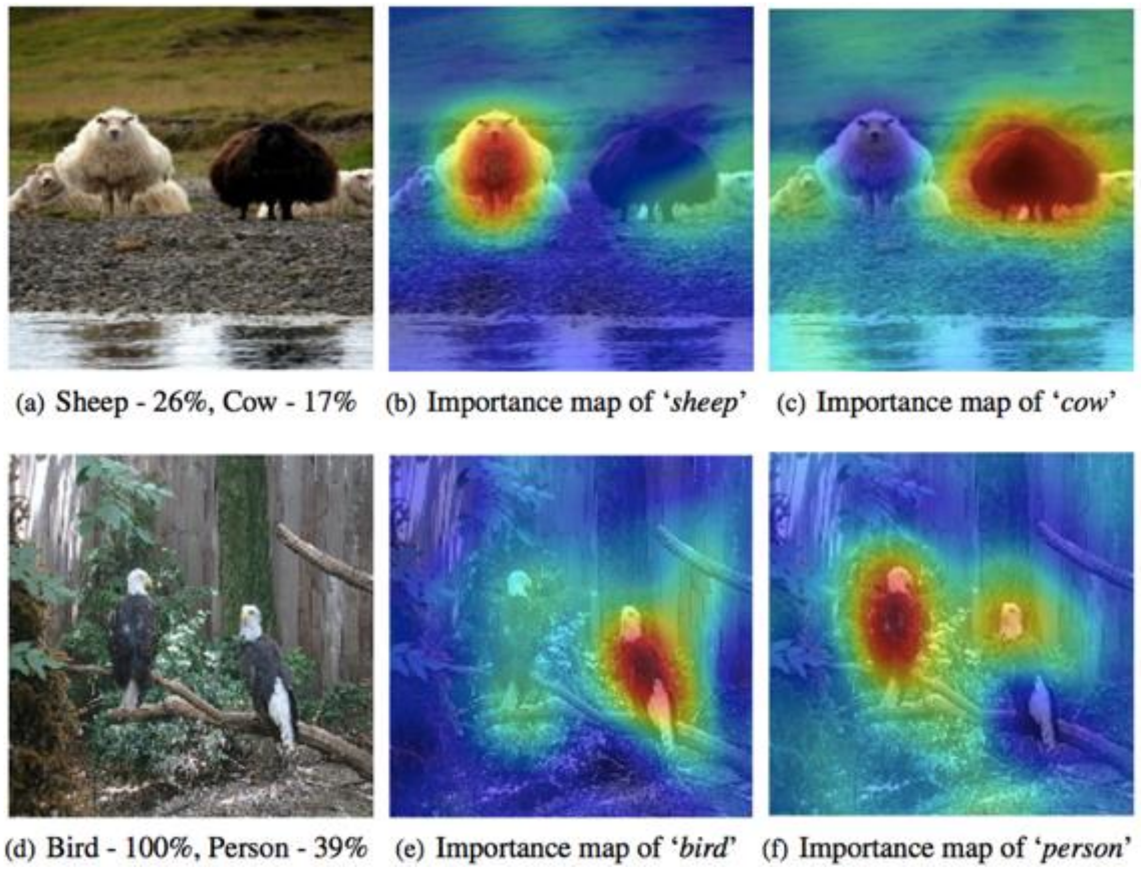
- Saliency of a class at a pixel is expected score for that class over all masks where the pixel is visible



V. Petsiuk, A. Das, K. Saenko, [RISE: Randomized Input Sampling for Explanation of Black-box Models](), BMVC 2018

“Black box” saliency via masking

- Saliency of a class at a pixel is expected score for that class over all masks where the pixel is visible



V. Petsiuk, A. Das, K. Saenko, [RISE: Randomized Input Sampling for Explanation of Black-box Models](), BMVC 2018

“Black box” saliency via masking

- Application: detecting model/dataset bias

Prediction: “cow” 76%

Source: K. Saenko

“Black box” saliency via masking

- Application: detecting model/dataset bias

Baseline: *A **man** sitting at a desk with a laptop computer.*

Improved model: *A **woman** sitting in front of a laptop computer.*

L. Hendricks, K. Burns, K. Saenko, T. Darrell, A. Rohrbach, Women Also Snowboard: Overcoming Bias in Captioning Models, ECCV 2018

Outline

- Basic visualization techniques
- Mapping activations back to the image
- Synthesizing images to maximize activation
- Saliency maps
- Quantifying interpretability of units

Quantifying interpretability of units

- From the beginning, people have observed that many units in higher layers seem to fire on meaningful concepts
- But how can we quantify this?

AlexNet Layer 4 AlexNet Layer 5

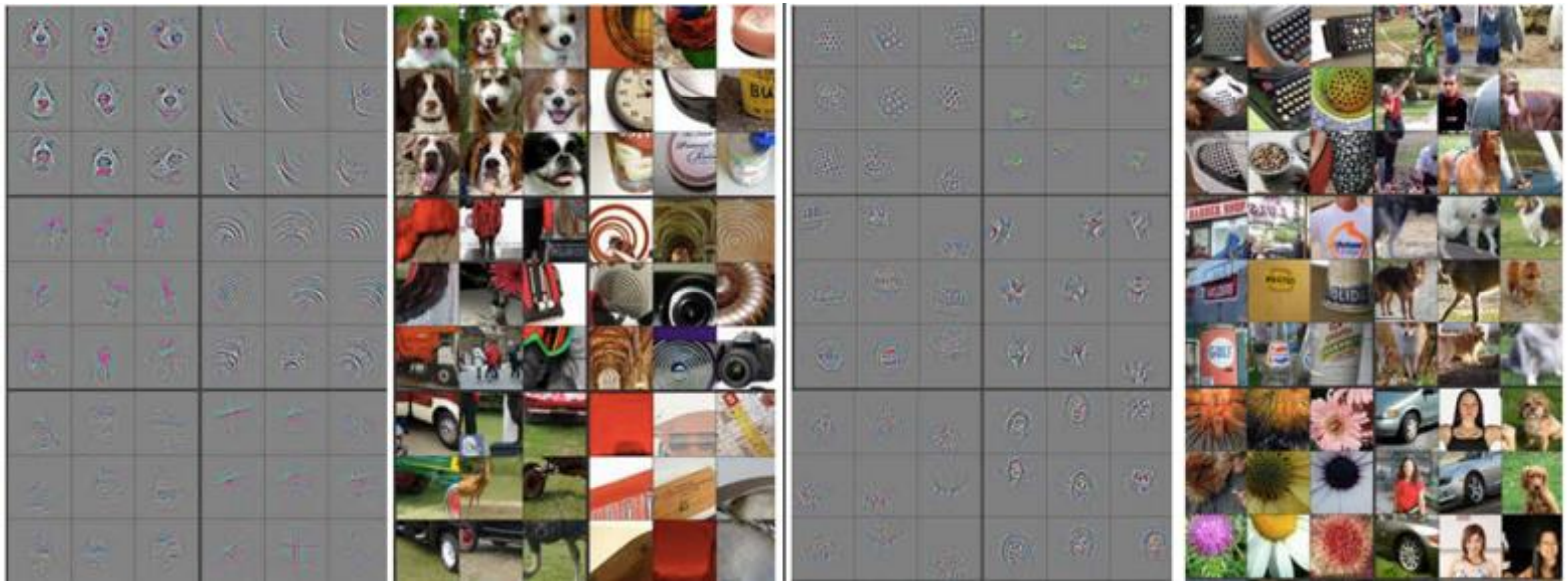
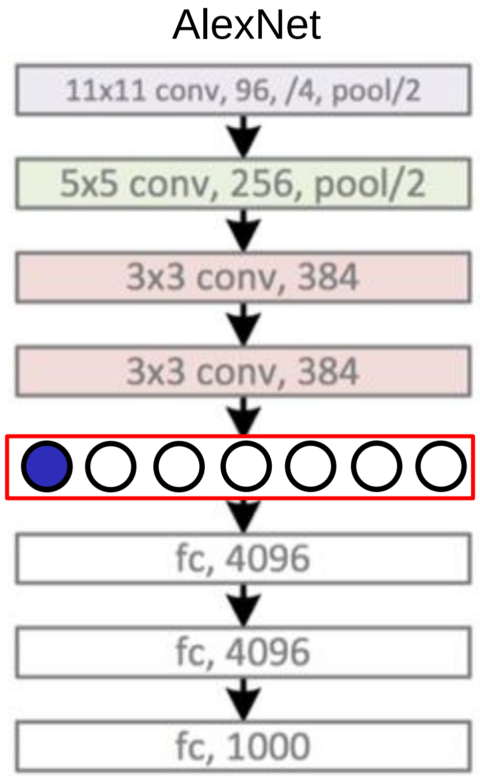
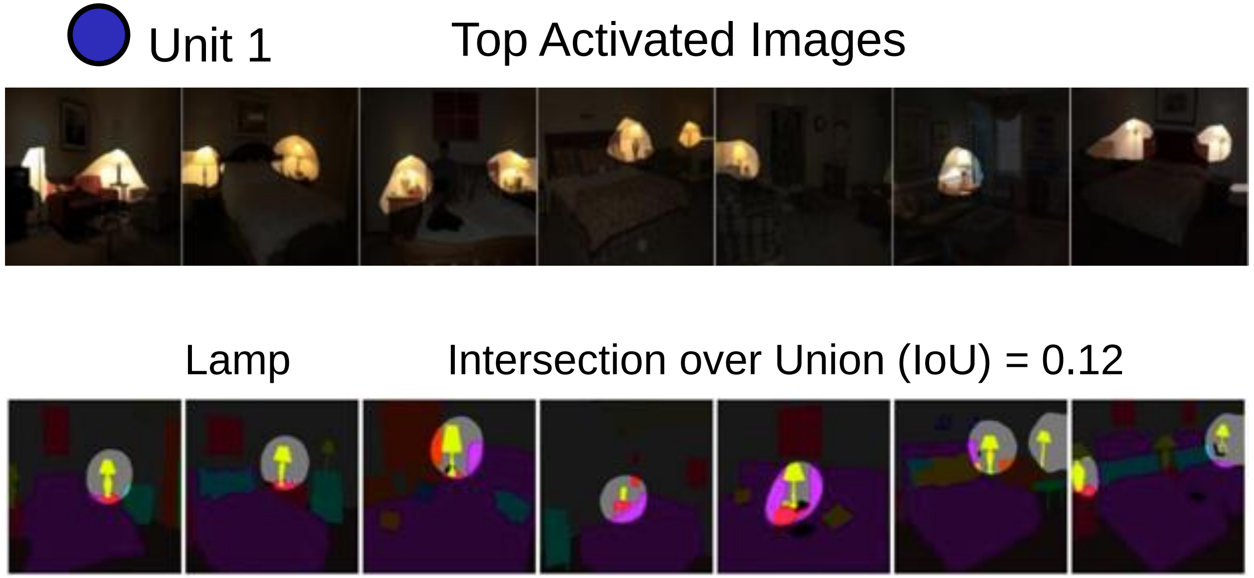


Figure: Zeiler & Fergus

Quantifying interpretability of units

- For a given unit, measure the overlap between areas of high activation and semantic segmentations for a large set of visual concepts

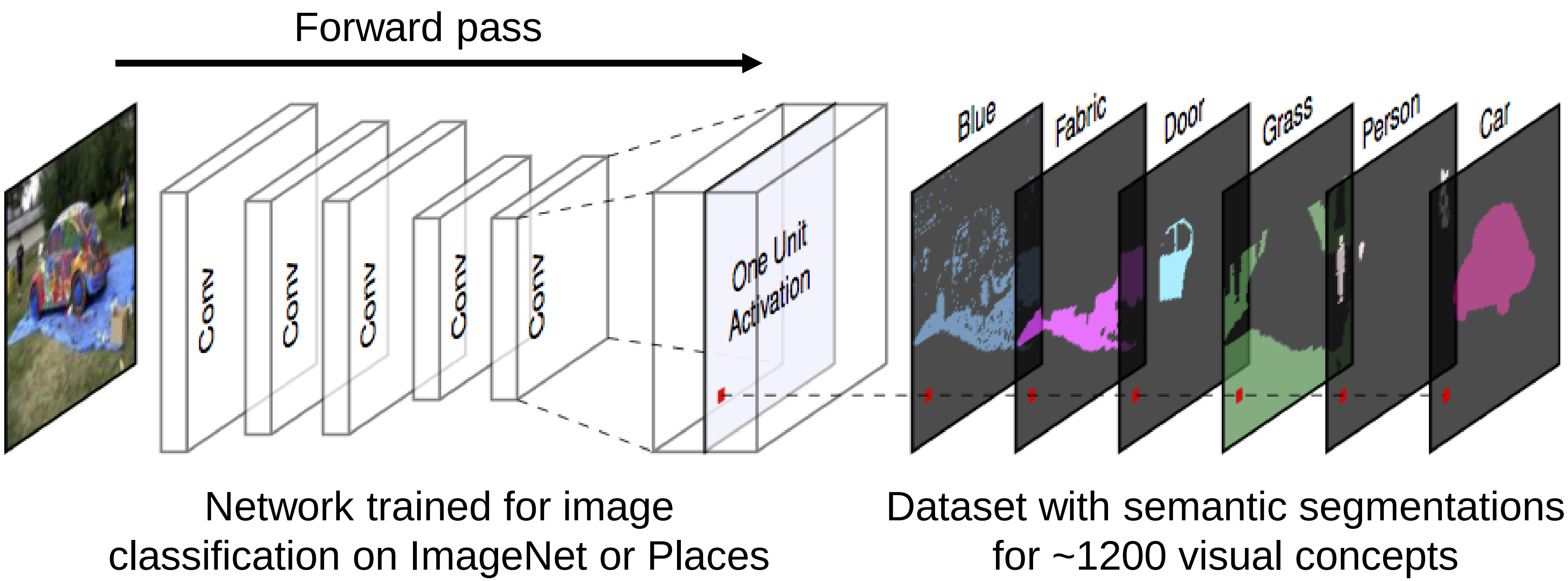




D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, [Network Dissection: Quantifying Interpretability of Deep Visual Representations](), CVPR 2017

Quantifying interpretability of units

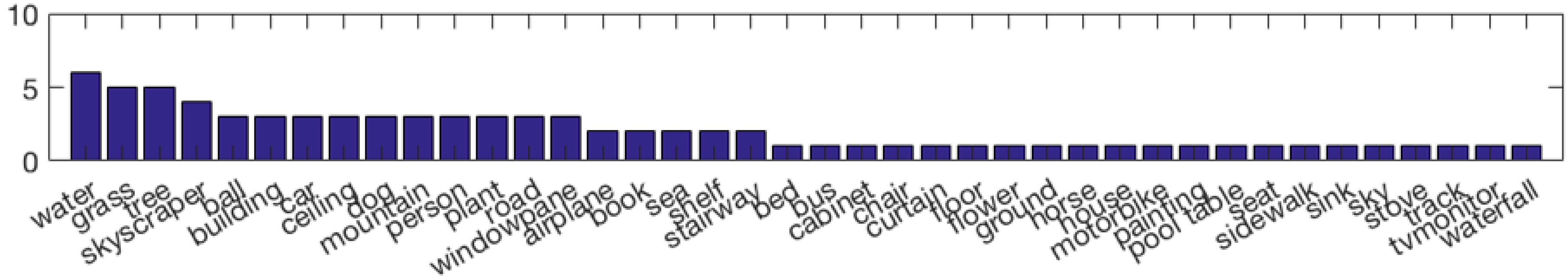
- For a given unit, measure the overlap between areas of high activation and semantic segmentations for a large set of visual concepts



D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, [Network Dissection: Quantifying Interpretability of Deep Visual Representations](), CVPR 2017

Quantifying interpretability of units

Histogram of object detectors for Places AlexNet conv5 units
81/256 units with IoU > 0.04



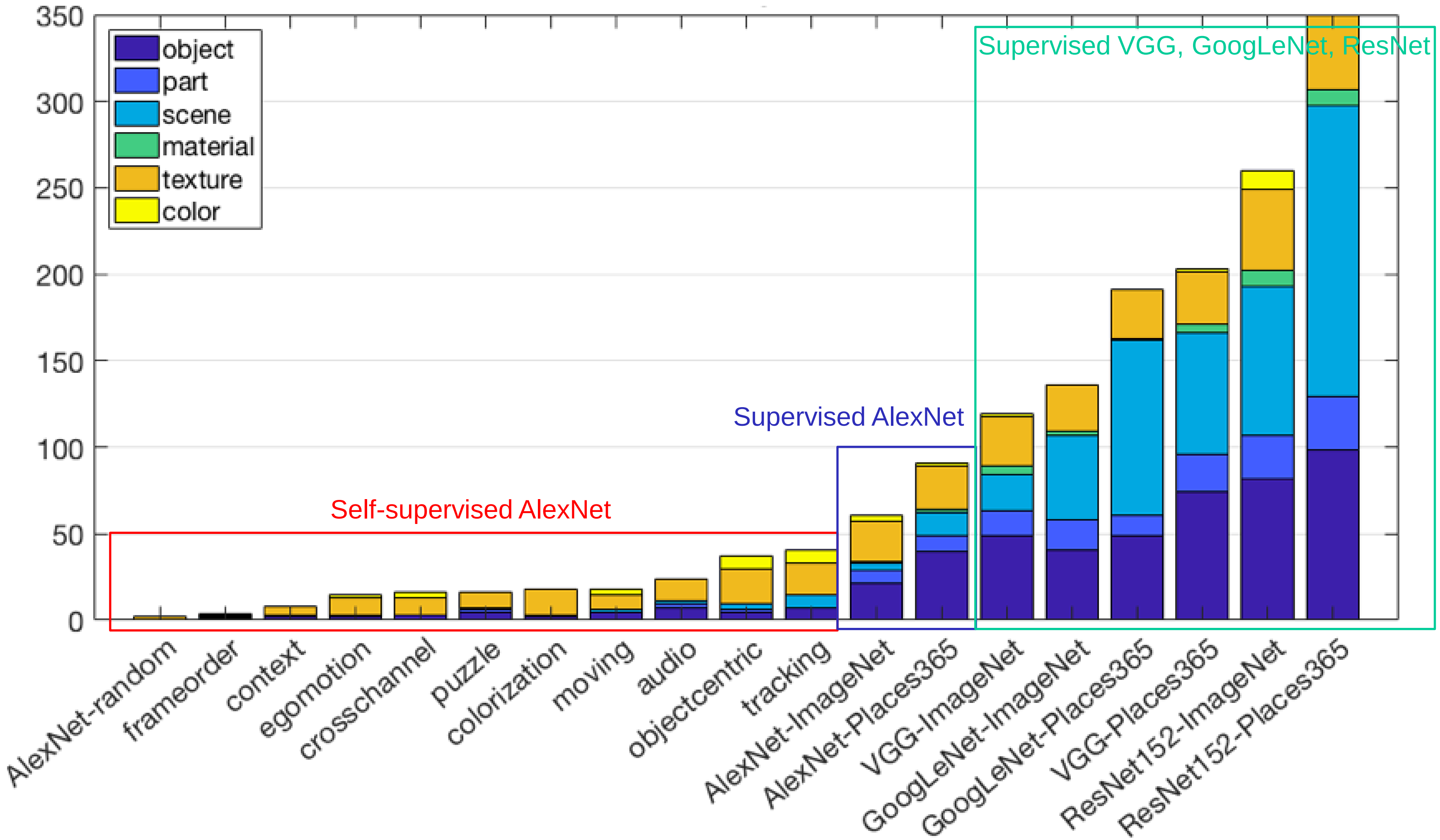
conv5 unit 79 car (object) IoU=0.13

conv5 unit 107 road (object) IoU=0.15

D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, Network Dissection: Quantifying Interpretability of Deep Visual Representations, CVPR 2017

Quantifying interpretability of units

Comparison of number of unique detectors across architectures



D. Bau, B. Zhou, A. Khosla, A. Oliva, A. Torralba, [Network Dissection: Quantifying Interpretability of Deep Visual Representations](), CVPR 2017

Summary

- Basic visualization techniques
  - Showing weights, top activated patches, nearest neighbors
- Mapping activations back to the image
  - Deconvolution
  - Guided back-propagation
- Synthesizing images to maximize activation
  - Gradient ascent with natural image regularization
- Saliency maps
  - “White box” vs. “black box”
- Explainability/interpretability
  - Explaining network decisions, detecting bias
  - Quantifying interpretability of intermediate units

Acknowledgement

Thanks to the following courses and corresponding researchers for making their teaching/research material online

- Deep Learning, Stanford University
- Introduction to Deep Learning, University of Illinois at Urbana-Champaign
- Introduction to Deep Learning, Carnegie Mellon University
- Convolutional Neural Networks for Visual Recognition, Stanford University
- Natural Language Processing with Deep Learning, Stanford University
- And Many More ......