Visualization and Understanding CNNs

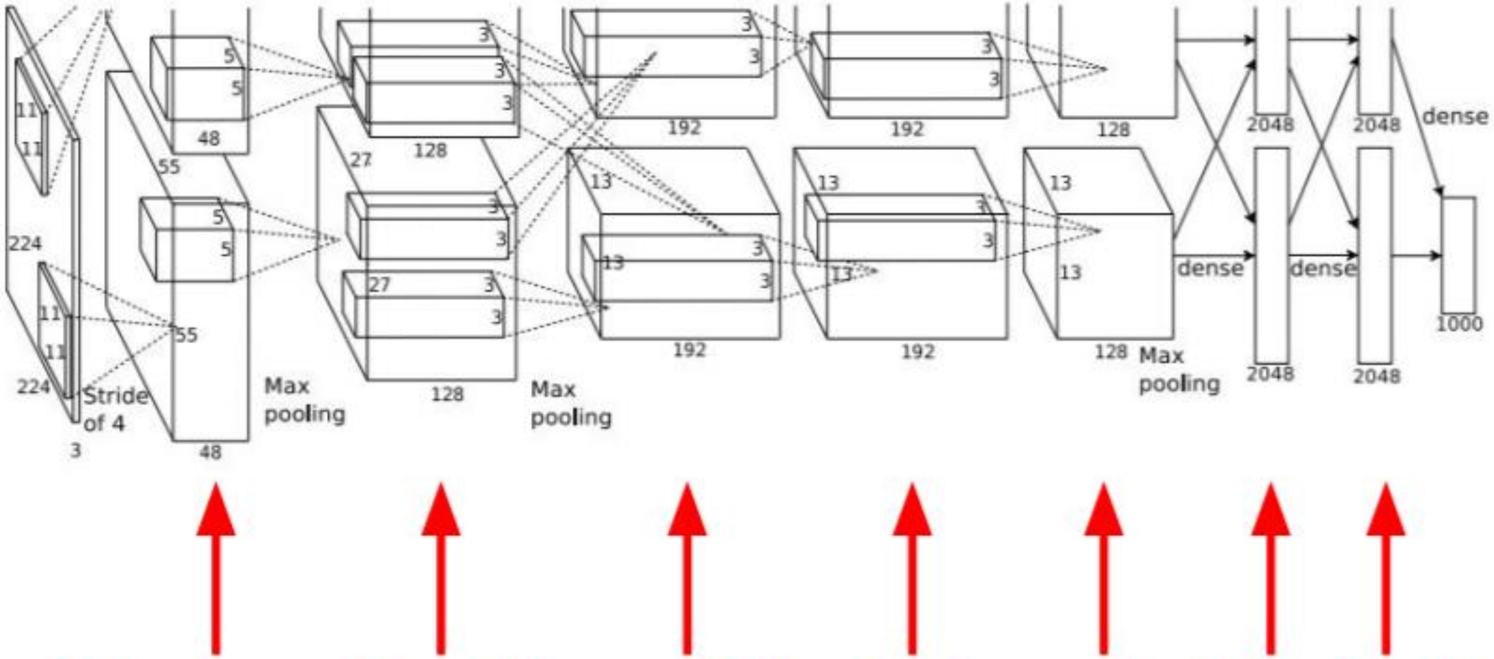
M. Soleymani
Sharif University of Technology
Fall 2017

Slides are based on Fei Fei Li and colleagues lectures, cs231n, Stanford 2017.

Interpretation for CNN layers

- Visualizing features to gain intuition about

Input Image:
3 x 224 x 224

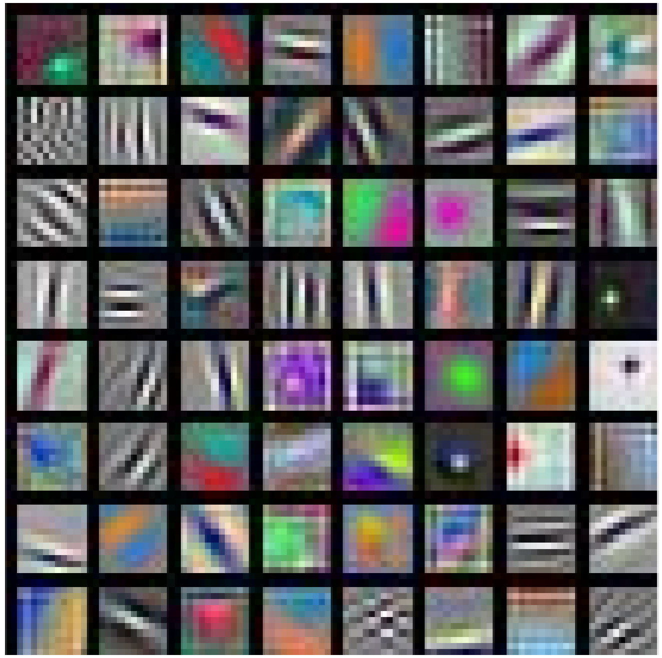


Class Scores:
1000 numbers

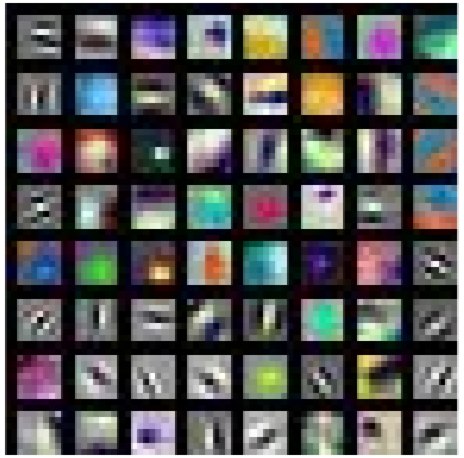
What are the intermediate features looking for?

Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

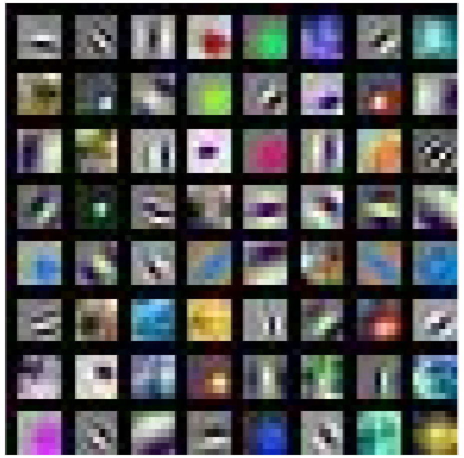
First Layer: Visualize Filters



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

Krizhevsky, “One weird trick for parallelizing convolutional neural networks”, arXiv 2014
He et al, “Deep Residual Learning for Image Recognition”, CVPR 2016
Huang et al, “Densely Connected Convolutional Networks”, CVPR 2017

Visualize the filters/kernels (raw weights)

- We can visualize filters at higher layers, but not that interesting

Source: http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

layer 1 weights
16 x 3 x 7 x 7

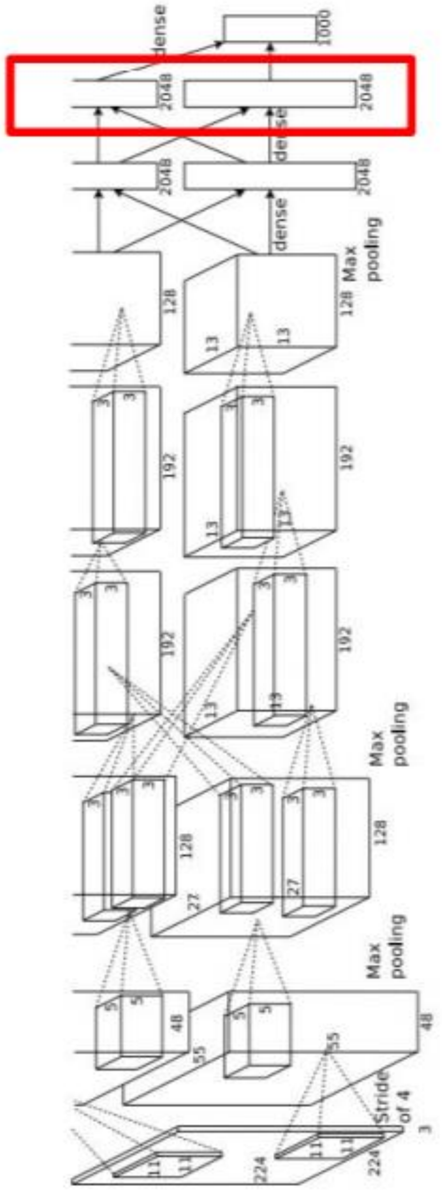
layer 2 weights
20 x 16 x 7 x 7

layer 3 weights
20 x 20 x 7 x 7

Last layer

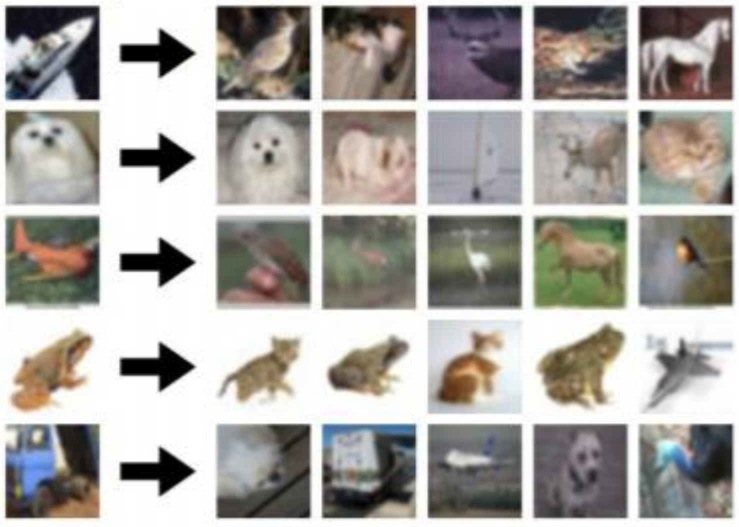
- 4096-dimensional feature vector for an image (layer immediately before the classifier)

- Run the network on many images, collect the feature vectors

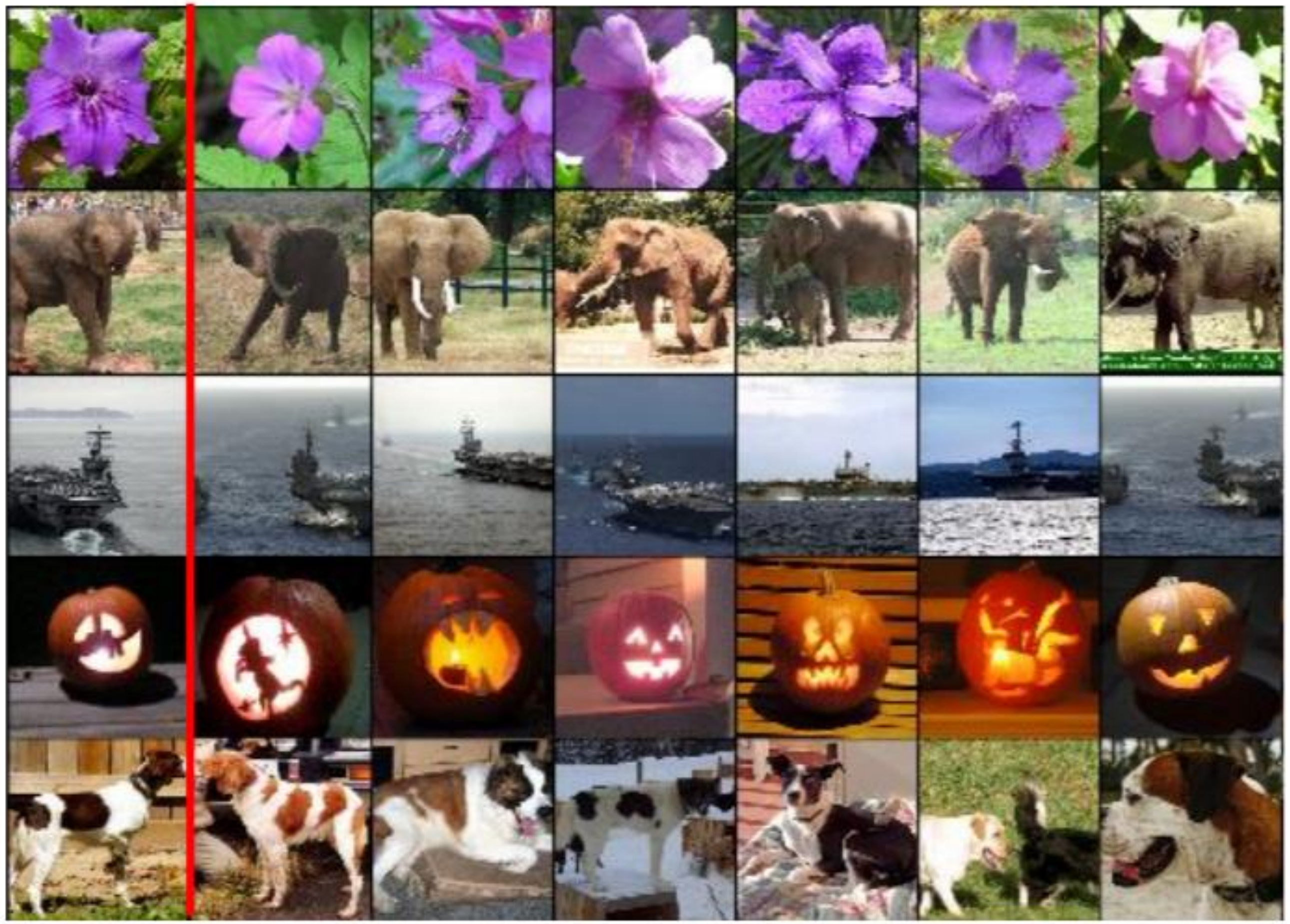


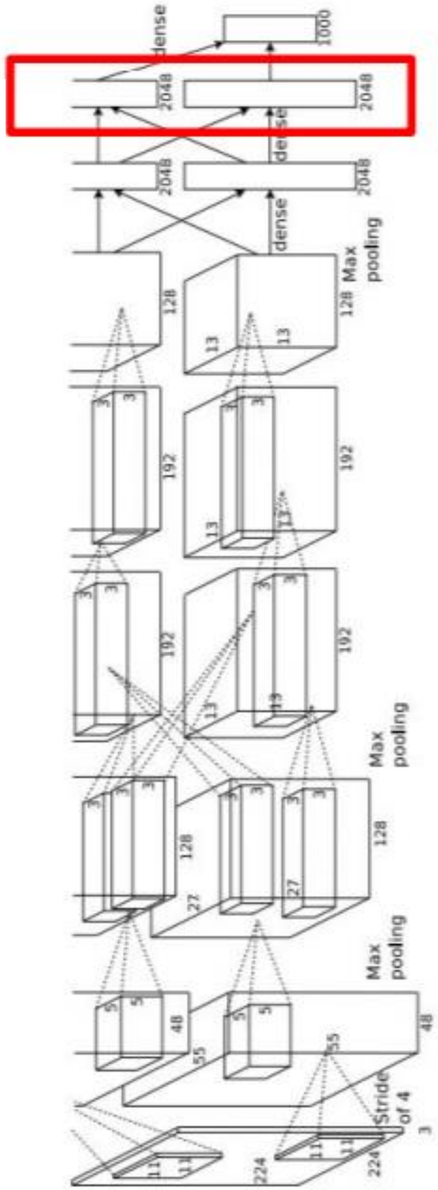
Last layer: Nearest neighbors

Nearest neighbors in pixel space



Test image   L2 Nearest neighbors in feature space

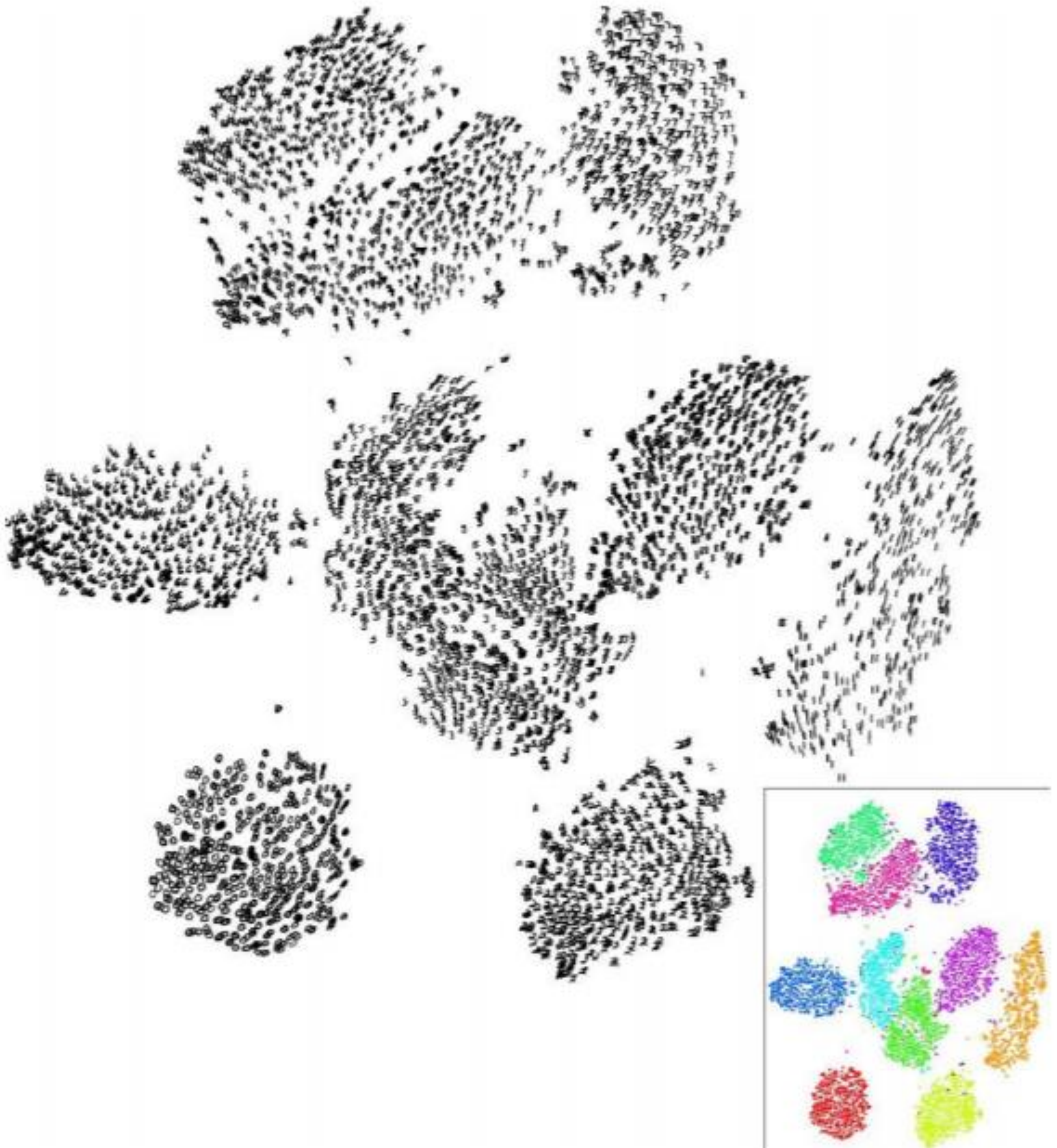




Krizhevsky et al, “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012.
Figures reproduced with permission.

Last Layer: Dimensionality Reduction

- Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2
  - dimensions Simple algorithm: Principle Component Analysis (PCA)
  - More complex: t-SNE



Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008

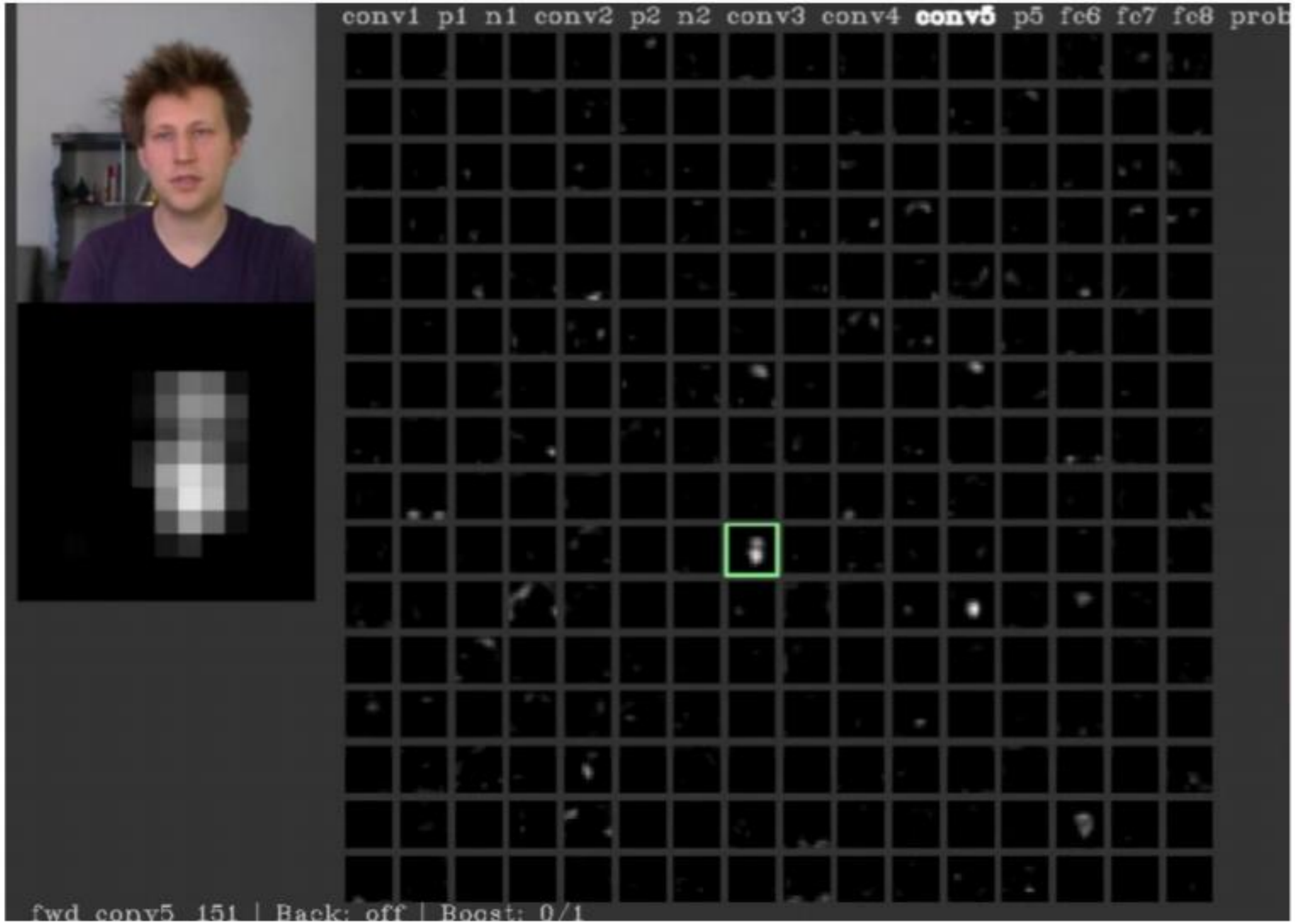
Last Layer: Dimensionality Reduction

Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

See high-resolution versions at
http://cs.stanford.edu/people/karpathy/cnnembed/

Visualizing Activations

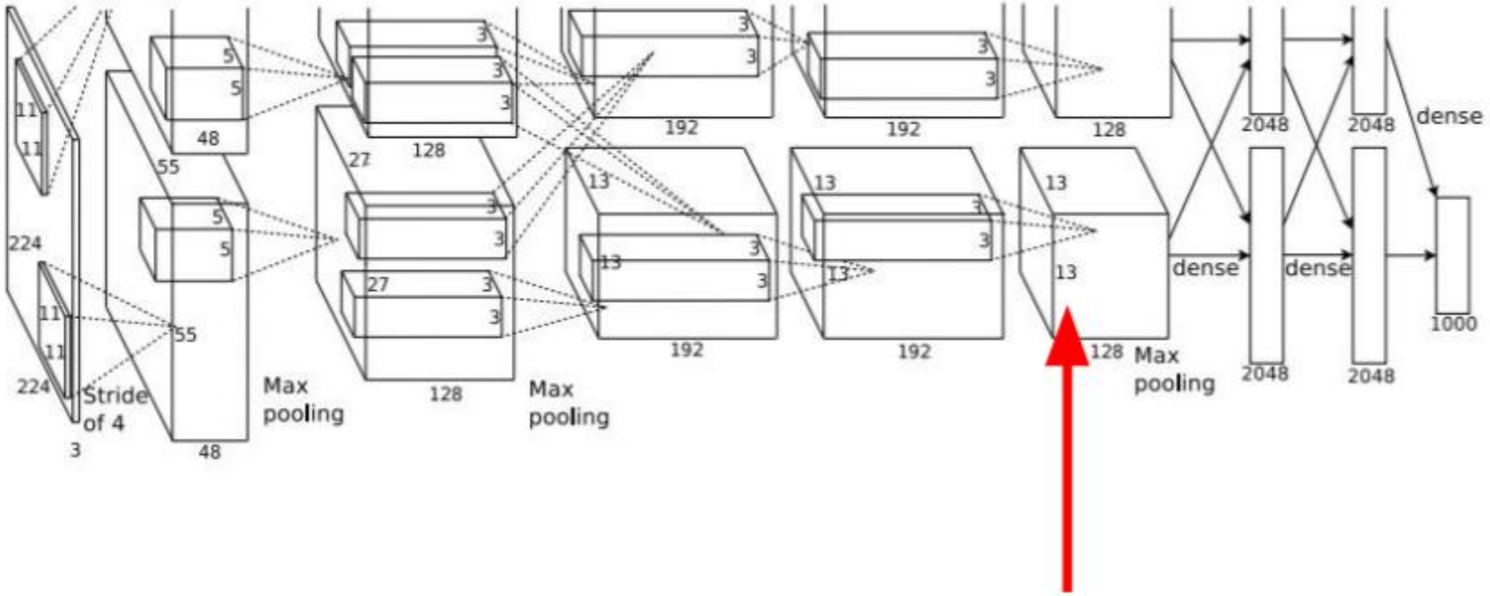
- conv5 feature map is 128x13x13;
  - visualize as 128 13x13 grayscale images

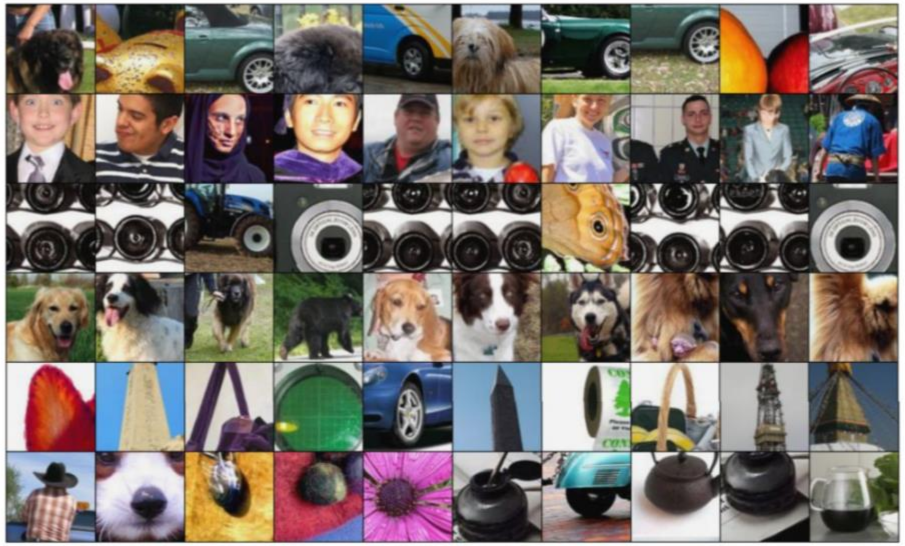


Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

Maximally Activating Patches

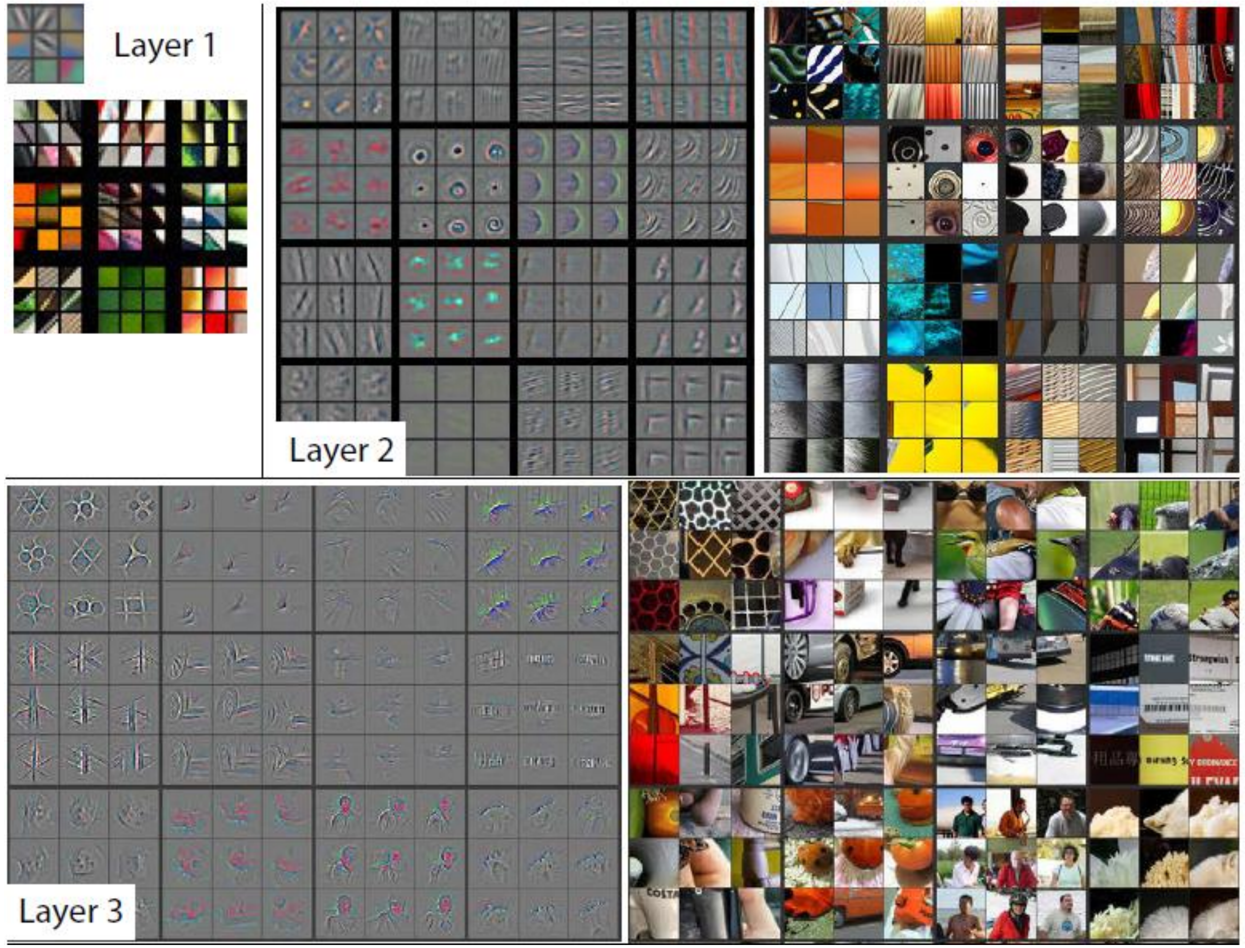
- Pick a layer and a channel;
  - e.g. conv5 is 128 x 13 x 13, pick channel 17/128
  - Run many images through the network, record values of chosen channel
  - Visualize image patches that correspond to maximal activations





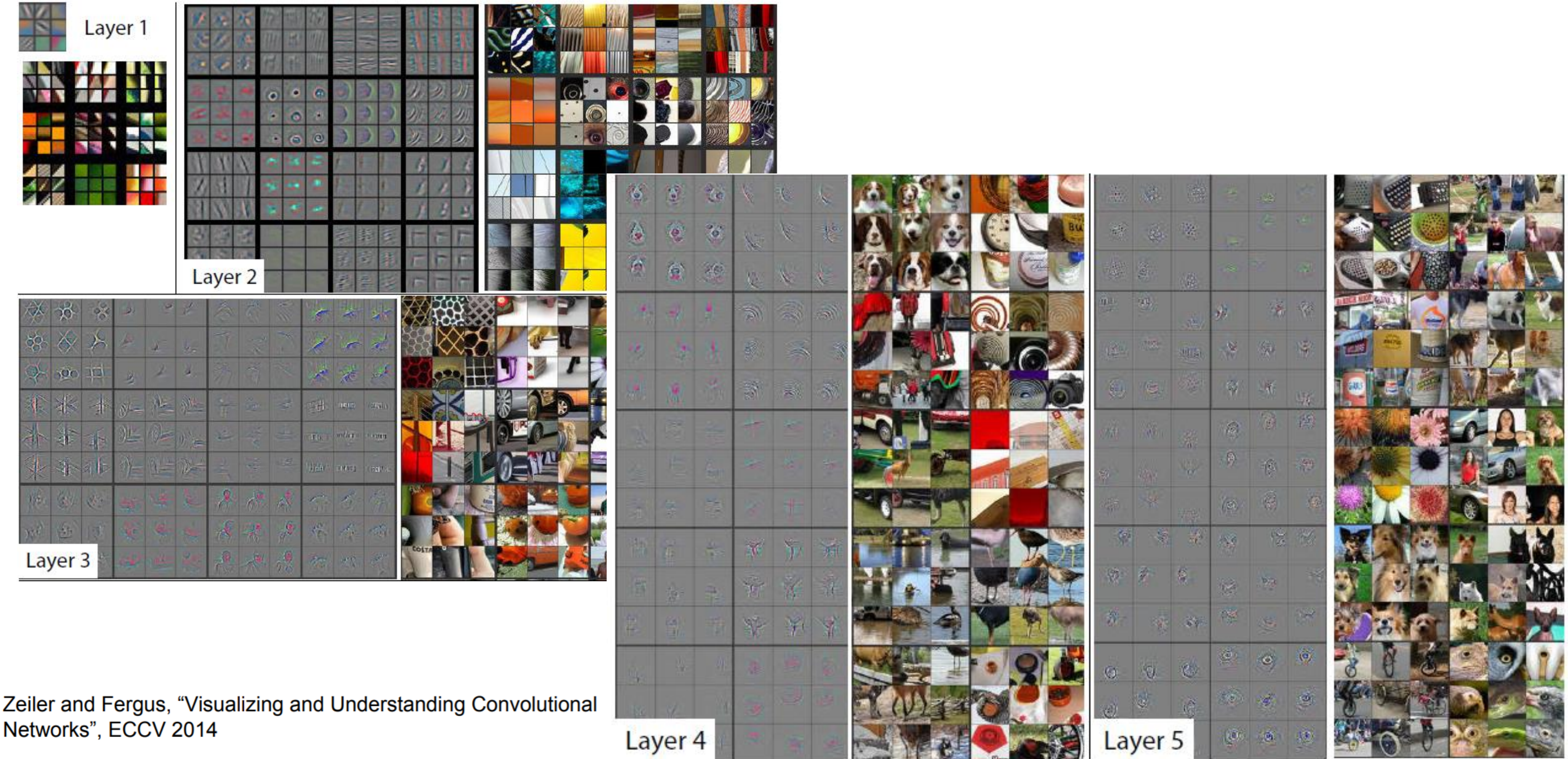
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Visualization with a Deconv Net



Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

Visualization with a Deconv Net

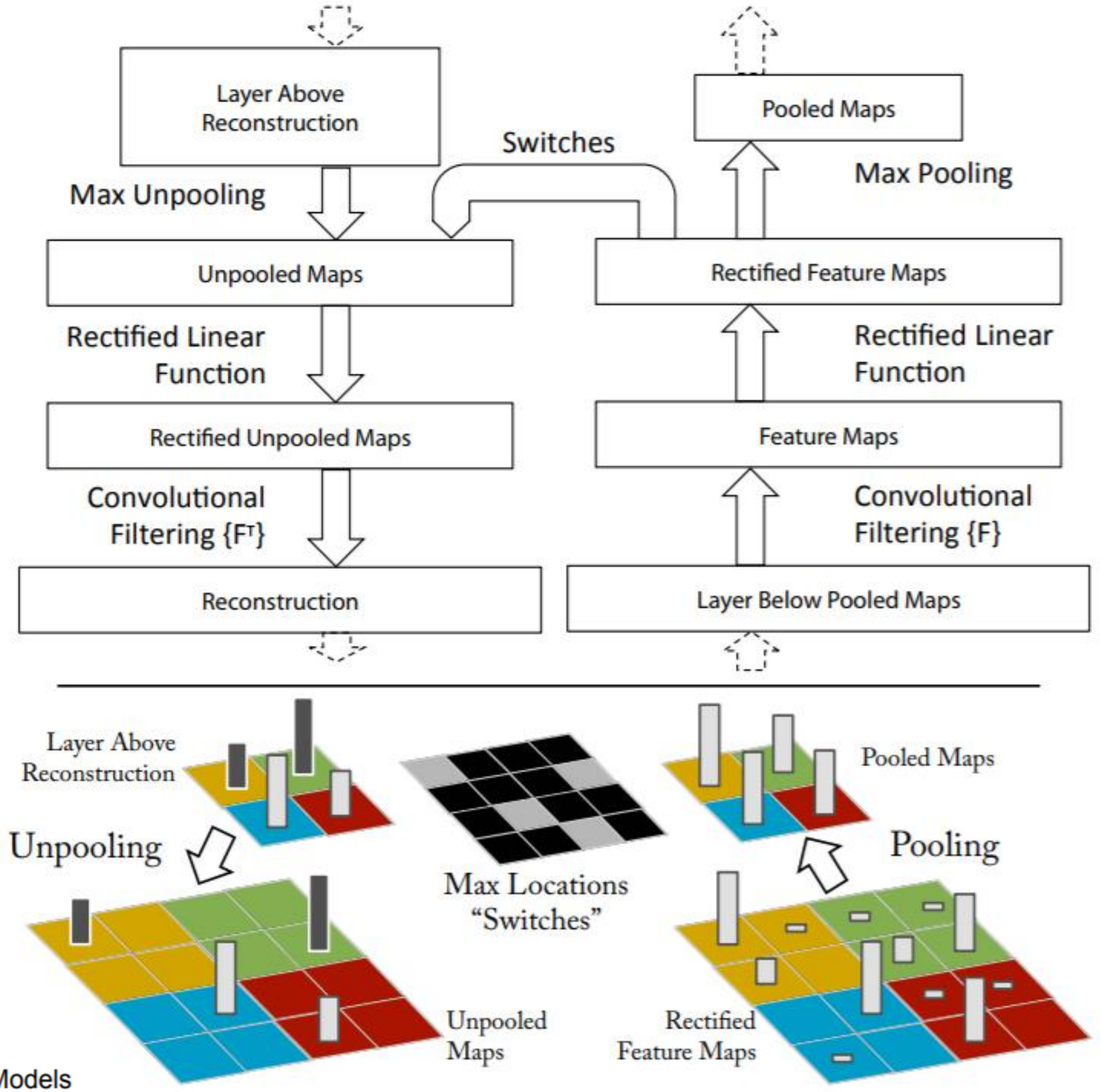


Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

Visualization with a Deconv Net

- Map activations back into pixel space by a DeconvNet

- Later, showed by Simonyan et al. that apart from the RELU layer, computing the reconstruction by DeconvNet is equivalent to computing the gradient w.r.t. the input image

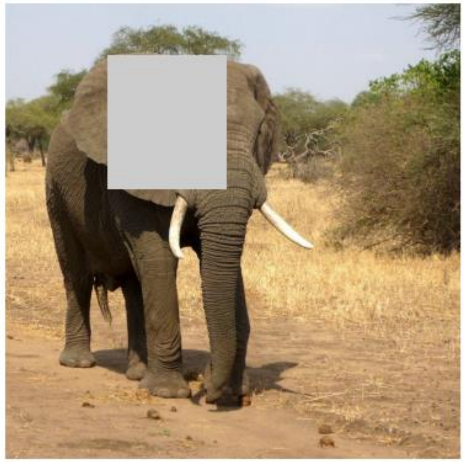


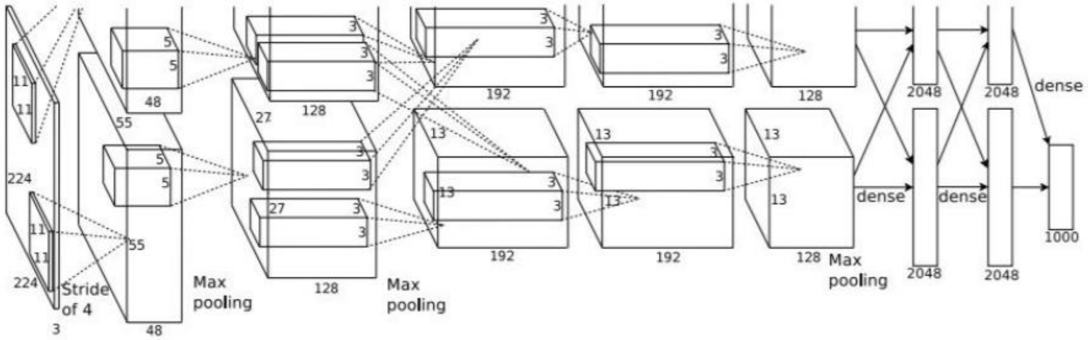
Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

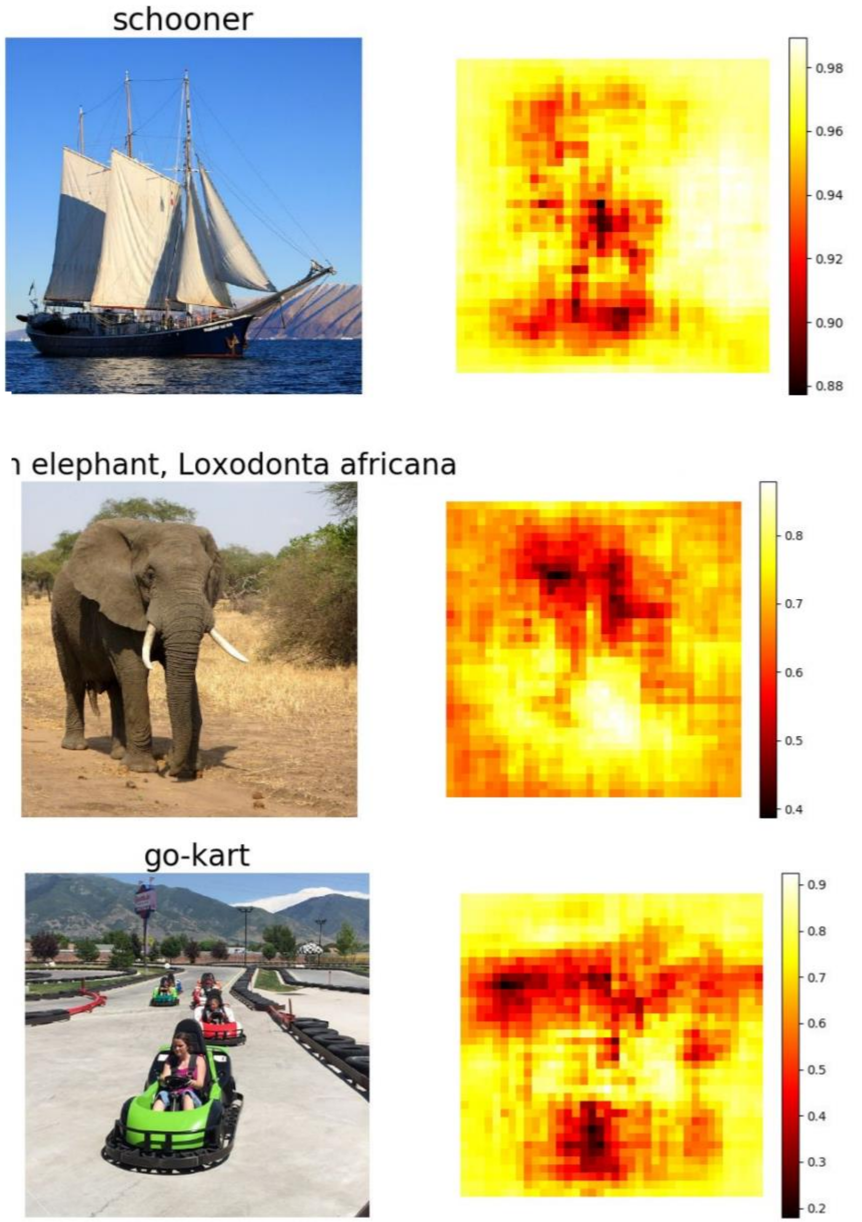
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Occlusion Experiments

- Mask part of the image before feeding to CNN, draw heatmap of probability at each mask location



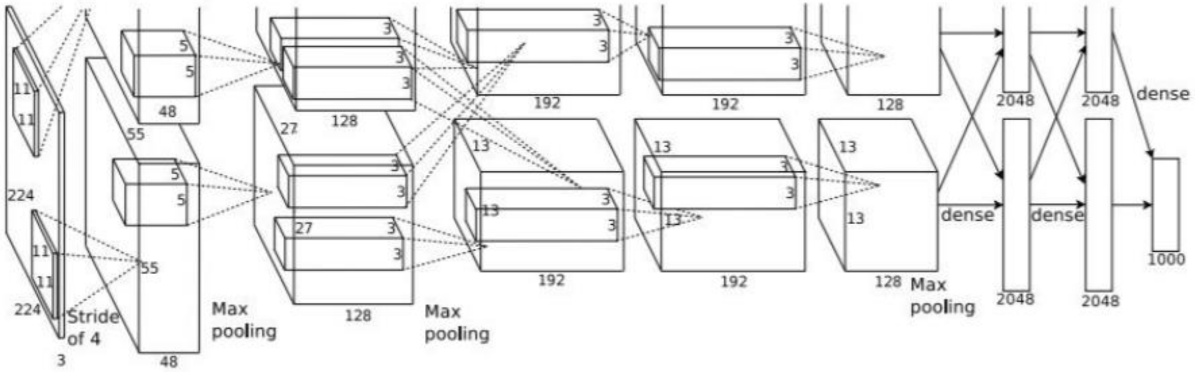




Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

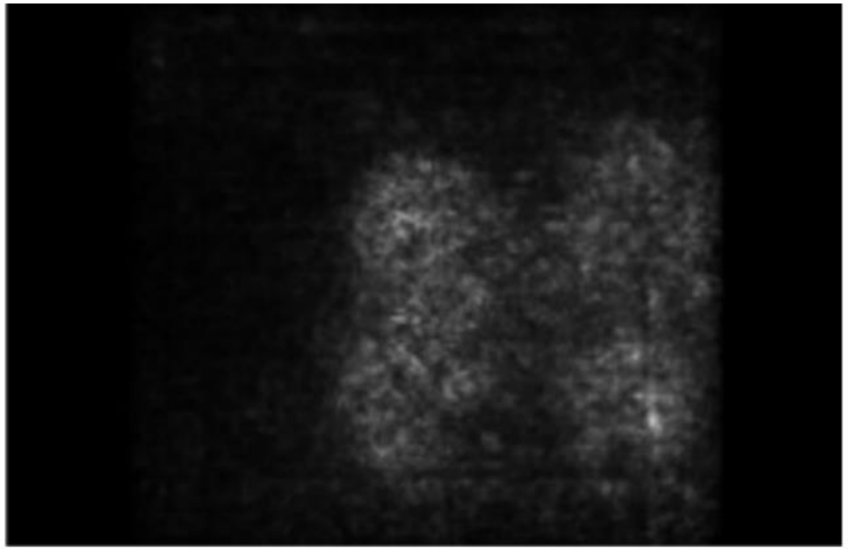
Saliency Maps

How to tell which pixels matter for classification?



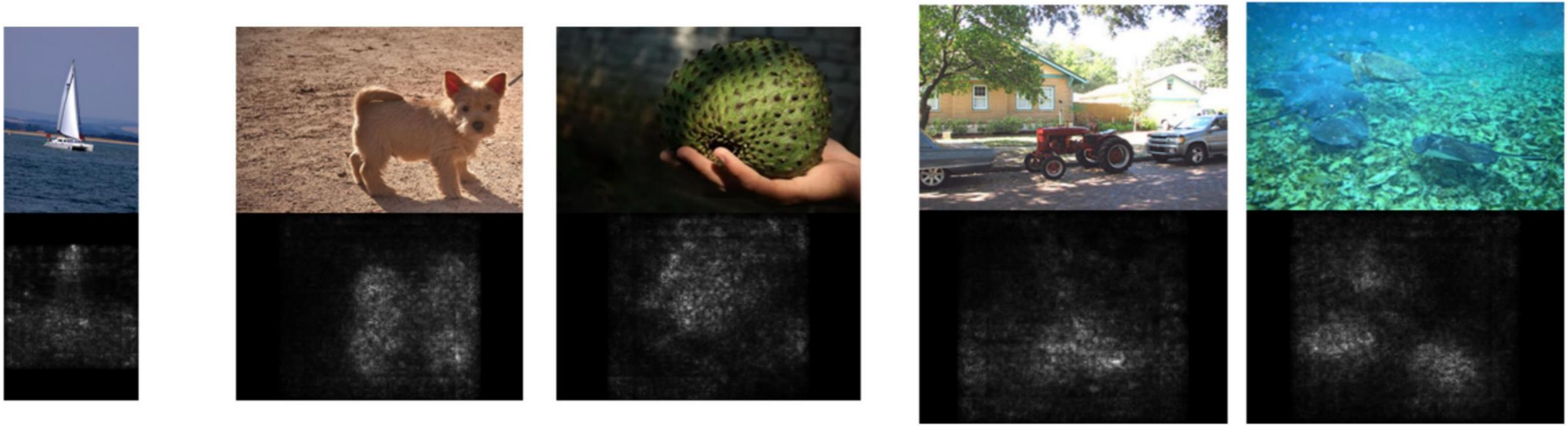
Dog

- Compute gradient of (unnormalized) class score with respect to image pixels,
  - It is computed for a pair of class and image
  - take absolute value and max over RGB channels



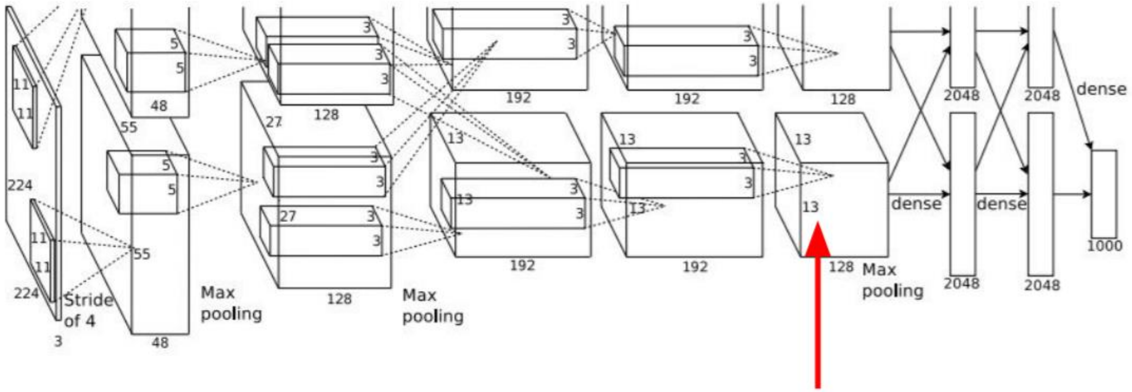
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Saliency Maps



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

Intermediate features via (guided) backprop

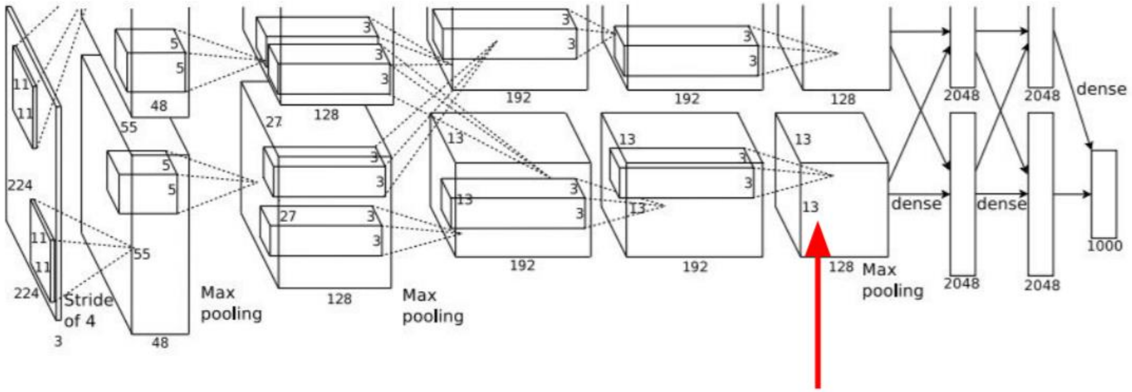


Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

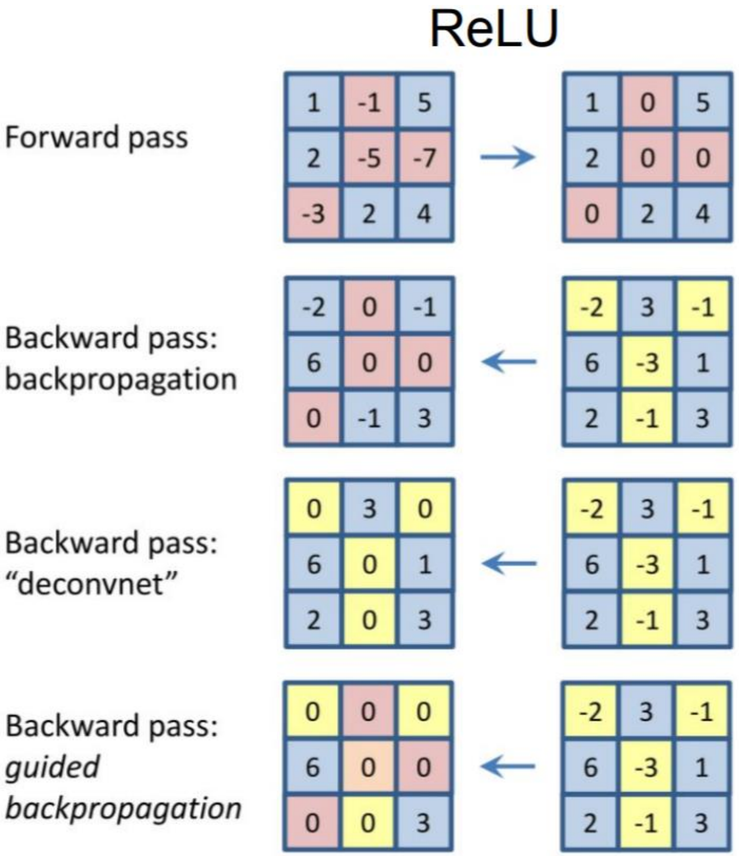
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014
Springenberg et al, “Striving for Simplicity: The All Convolutional Net”, ICLR Workshop 2015

Intermediate features via (guided) backprop



Pick a single intermediate neuron, e.g. one value in 128 x 13 x 13 conv5 feature map

Compute gradient of neuron value with respect to image pixels

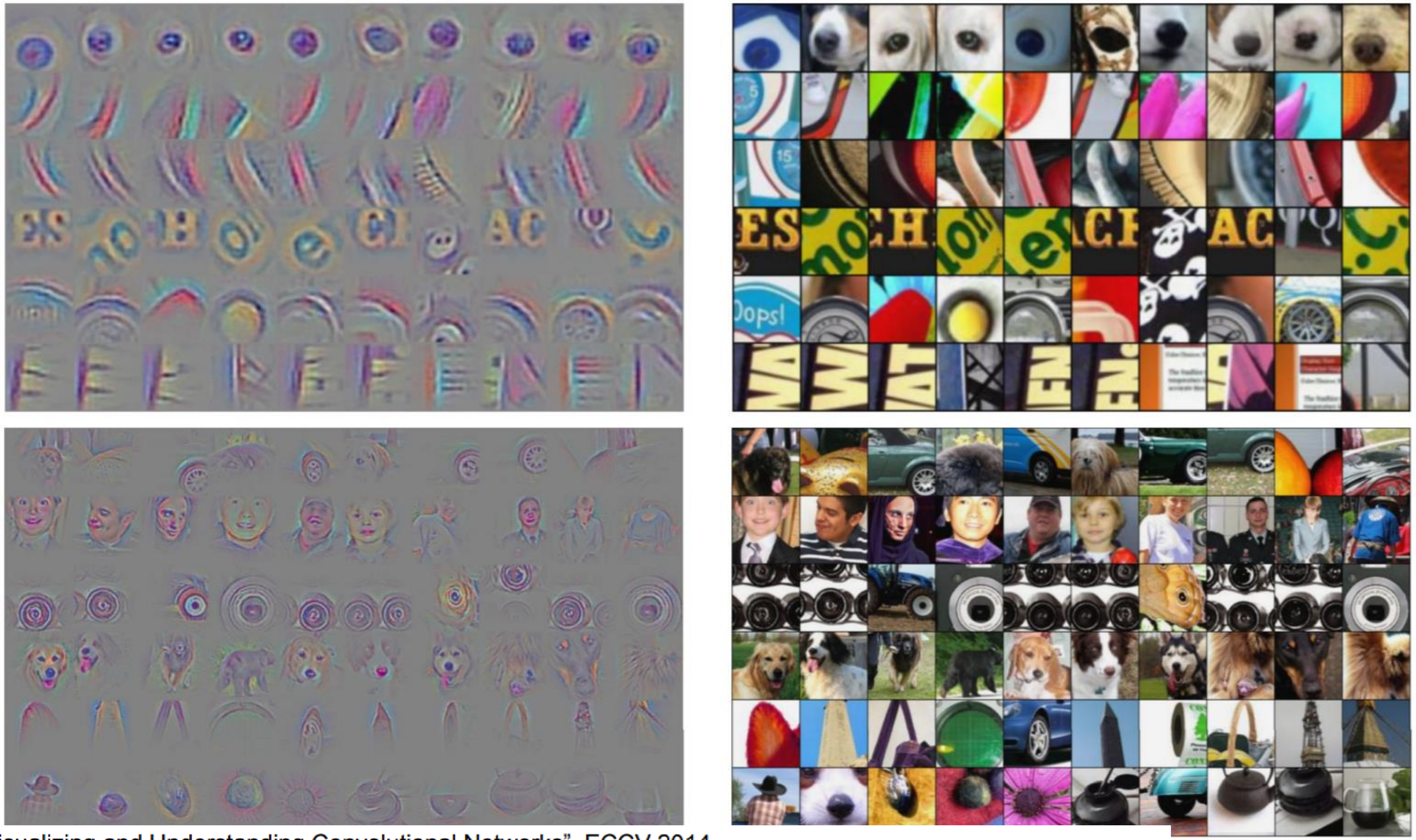


Images come out nicer if you only backprop positive gradients through each ReLU (guided backprop)

Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014
Springenberg et al, "Striving for Simplicity: The All Convolutional Net", ICLR Workshop 2015

Intermediate features via (guided) backprop

- (Guided) backprop: Find the part of an image that a neuron responds to



Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014
Springenberg et al, “Striving for Simplicity: The All Convolutional Net”, ICLR Workshop 2015

Visualizing CNN features: Gradient Ascent

- Gradient ascent: Generate a synthetic image that maximally activates a neuron

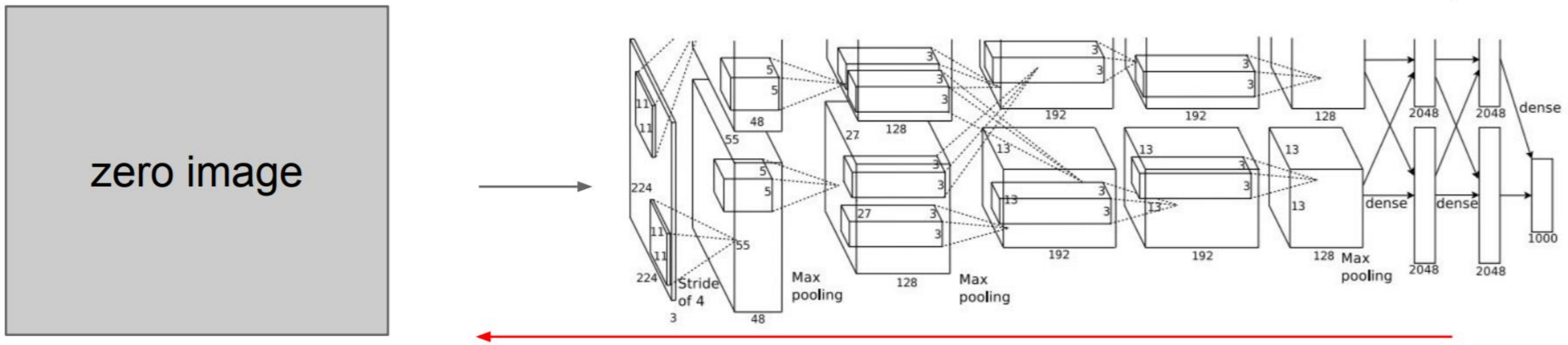
$$I^* = \arg\max_I \underbrace{\boxed{f(I)}}_{\text{Neuron value}} + \underbrace{\boxed{R(I)}}_{\text{Natural image regularizer}}$$

Visualizing CNN features: Gradient Ascent

$$\arg\max_{I} \boxed{S_c(I)} - \lambda \|I\|_2^2$$

score for class c (before Softmax)

1. Initialize image to zeros

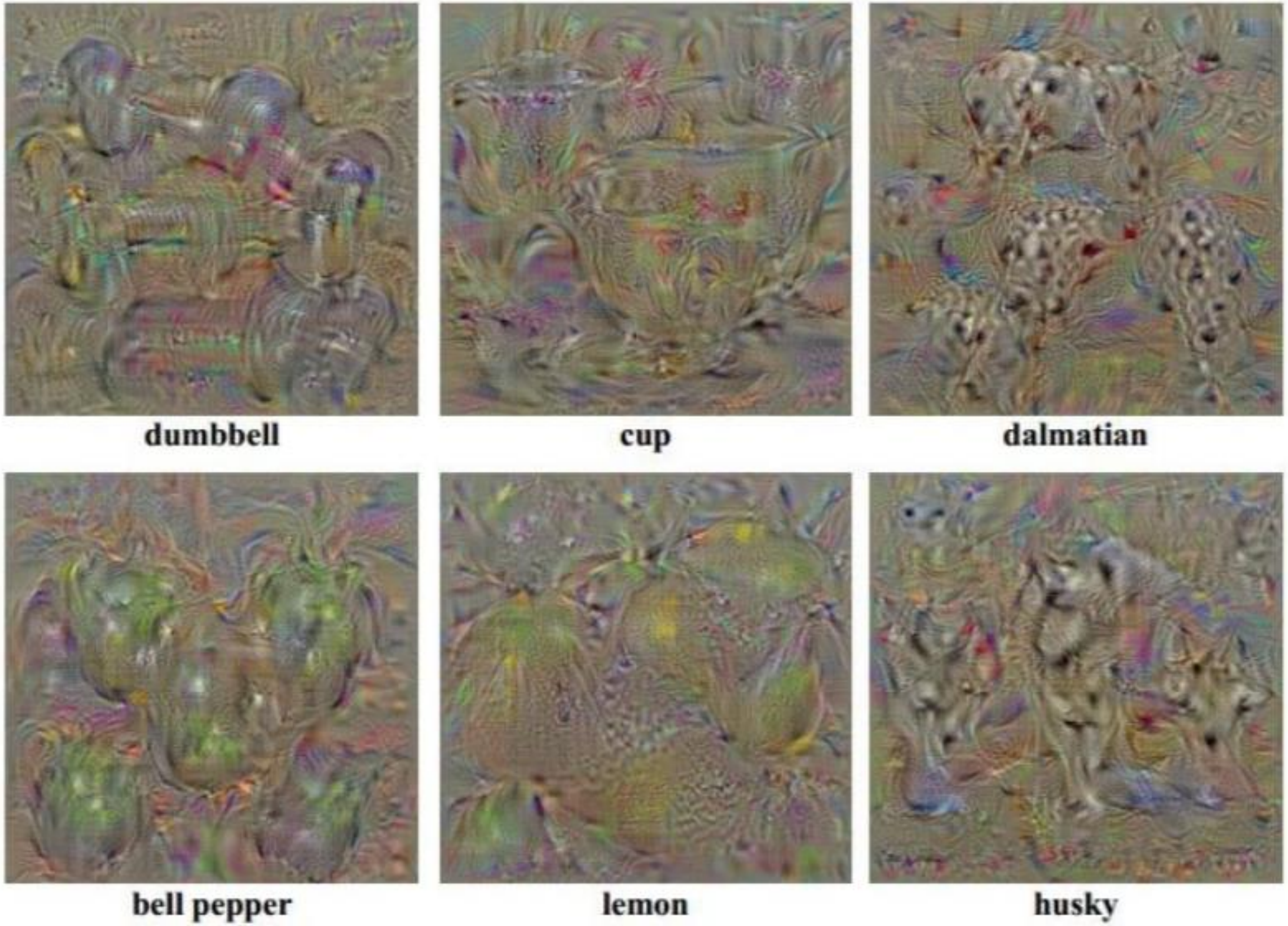


Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Visualizing CNN features: Gradient Ascent

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

Simple regularizer: Penalize L2 norm of generated image

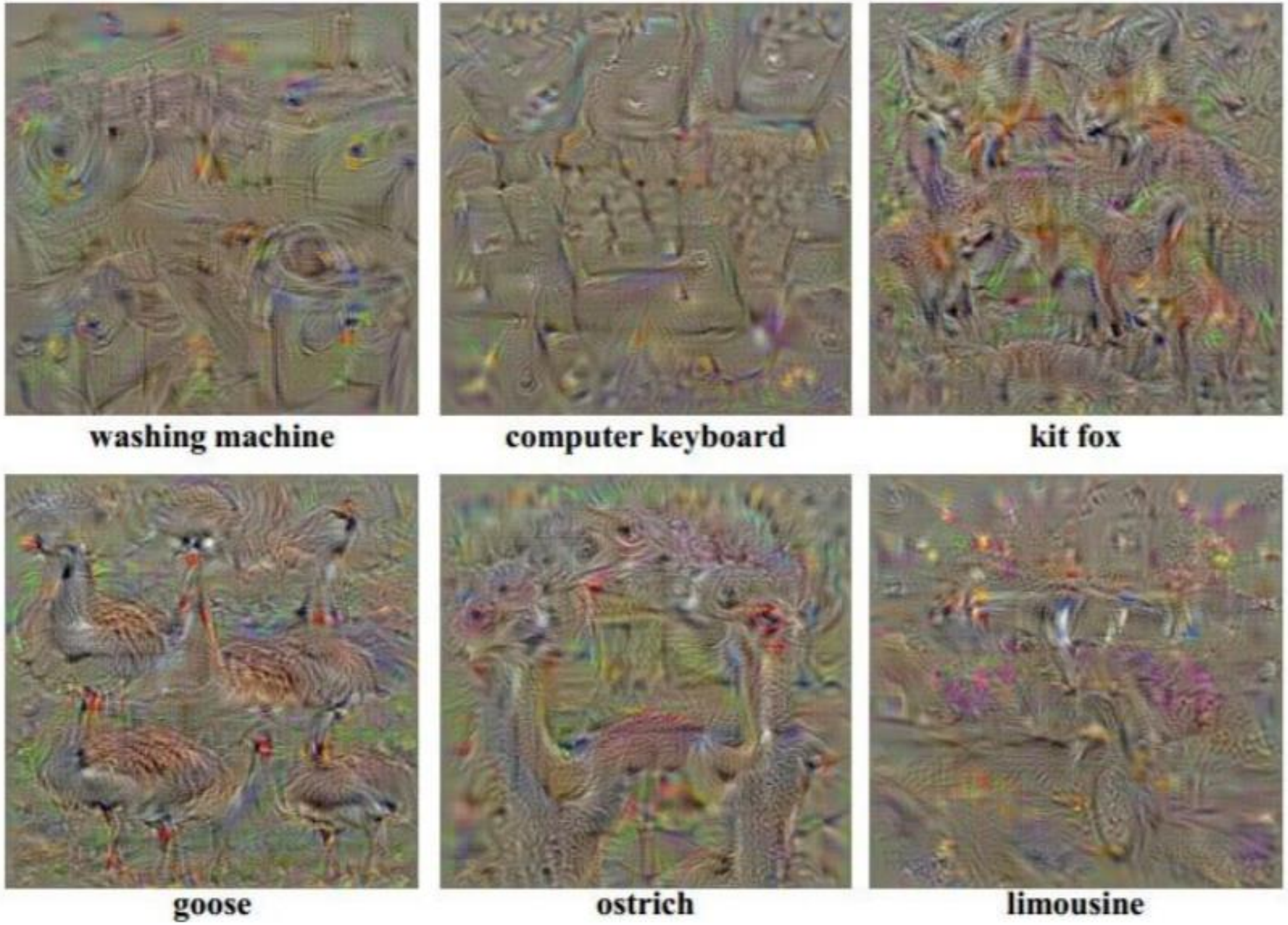


Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Visualizing CNN features: Gradient Ascent

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

Simple regularizer: Penalize L2 norm of generated image



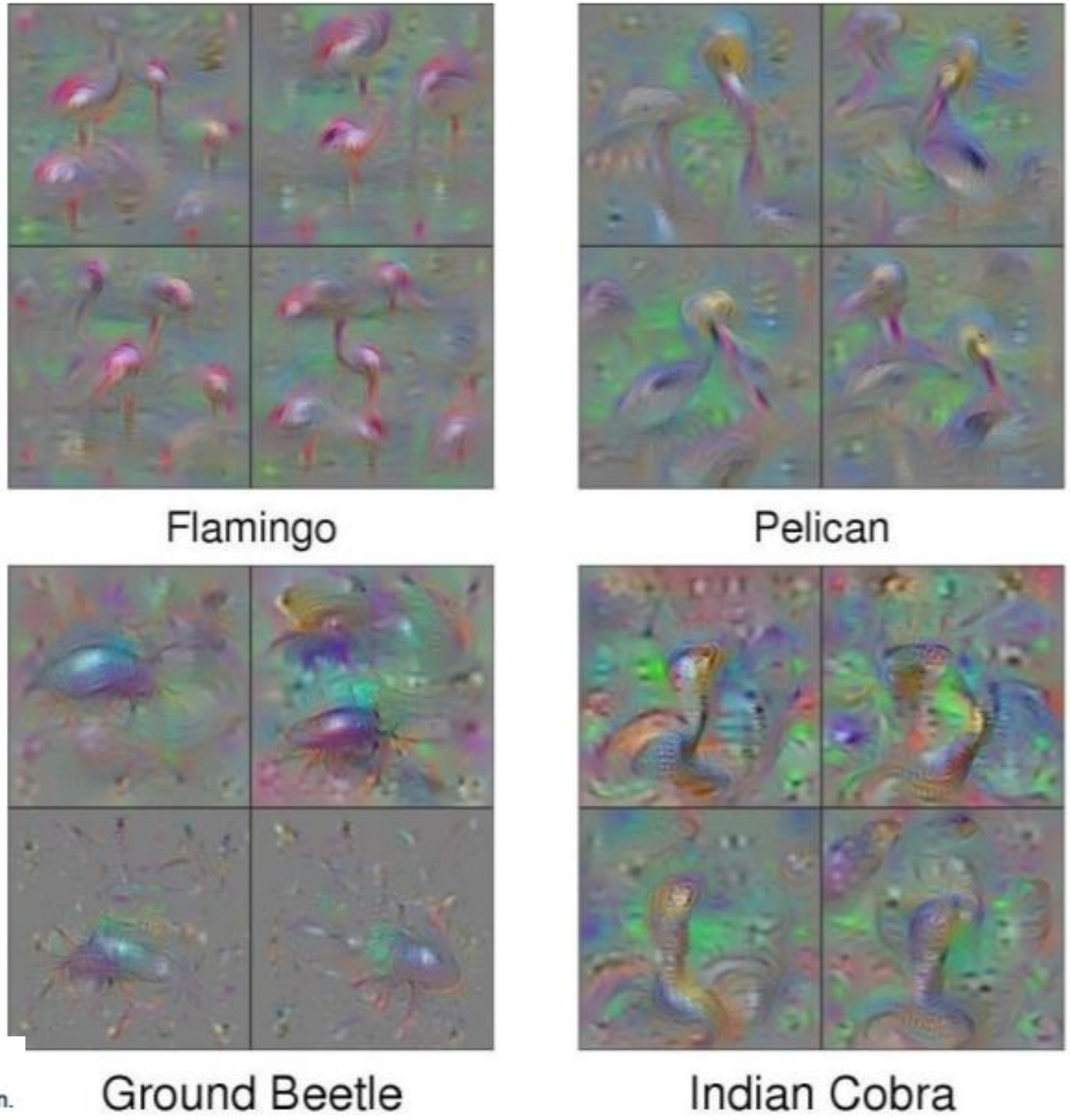
Simonyan, Vedaldi, and Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", ICLR Workshop 2014.

Visualizing CNN features: Gradient Ascent

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

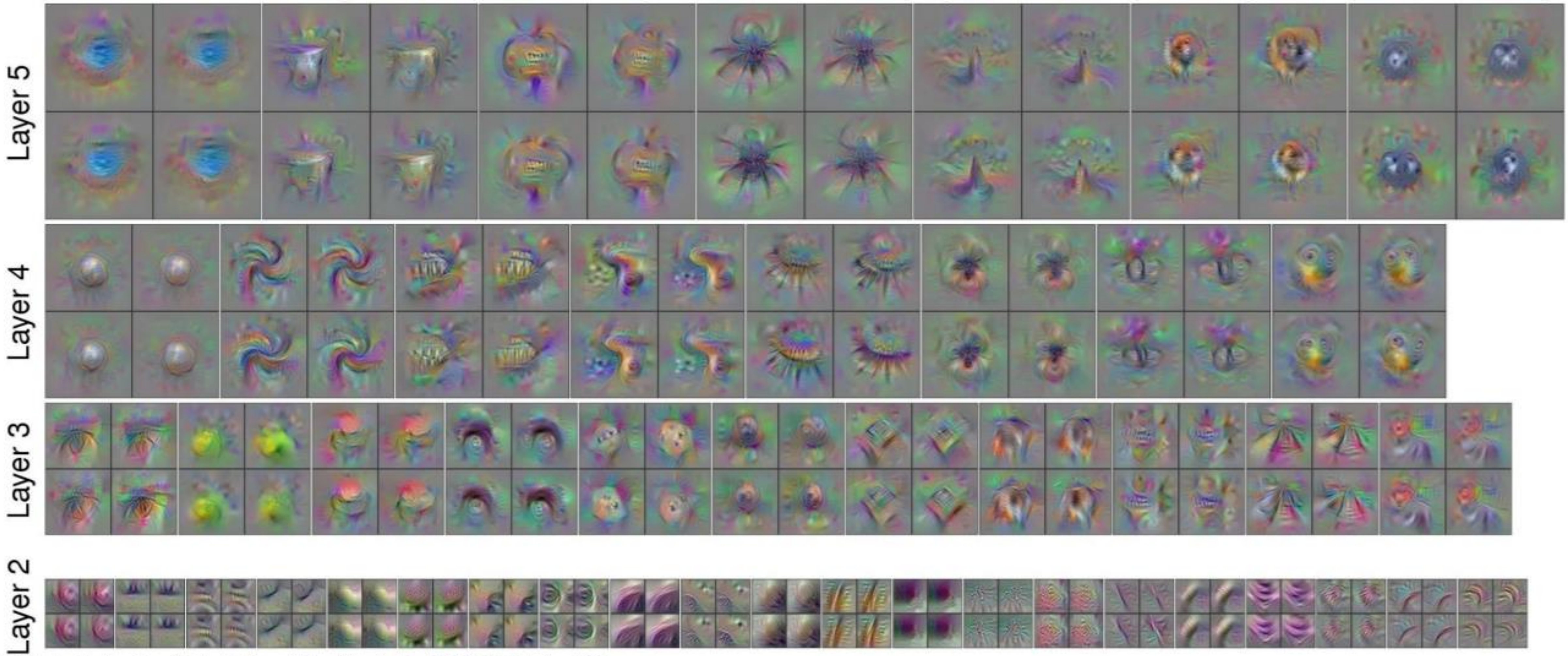
Better regularizer: Penalize L2 norm of image; also during optimization periodically

(1) Gaussian blur image
(2) Clip pixels with small values to 0
(3) Clip pixels with small gradients to 0



Visualizing CNN features: Gradient Ascent

Use the same approach to visualize intermediate features



Yosinski et al, "Understanding Neural Networks Through Deep Visualization", ICML DL Workshop 2014.

Fooling Images / Adversarial Examples

(1) Start from an arbitrary image

(2) Pick an arbitrary class

(3) Modify the image to maximize the class

(4) Repeat until network is fooled

Fooling Images / Adversarial Examples