بنام خداوند جان و خرد

Interpretability
in (Convolutional) Neural Networks

Mohammad Taher Pilehvar

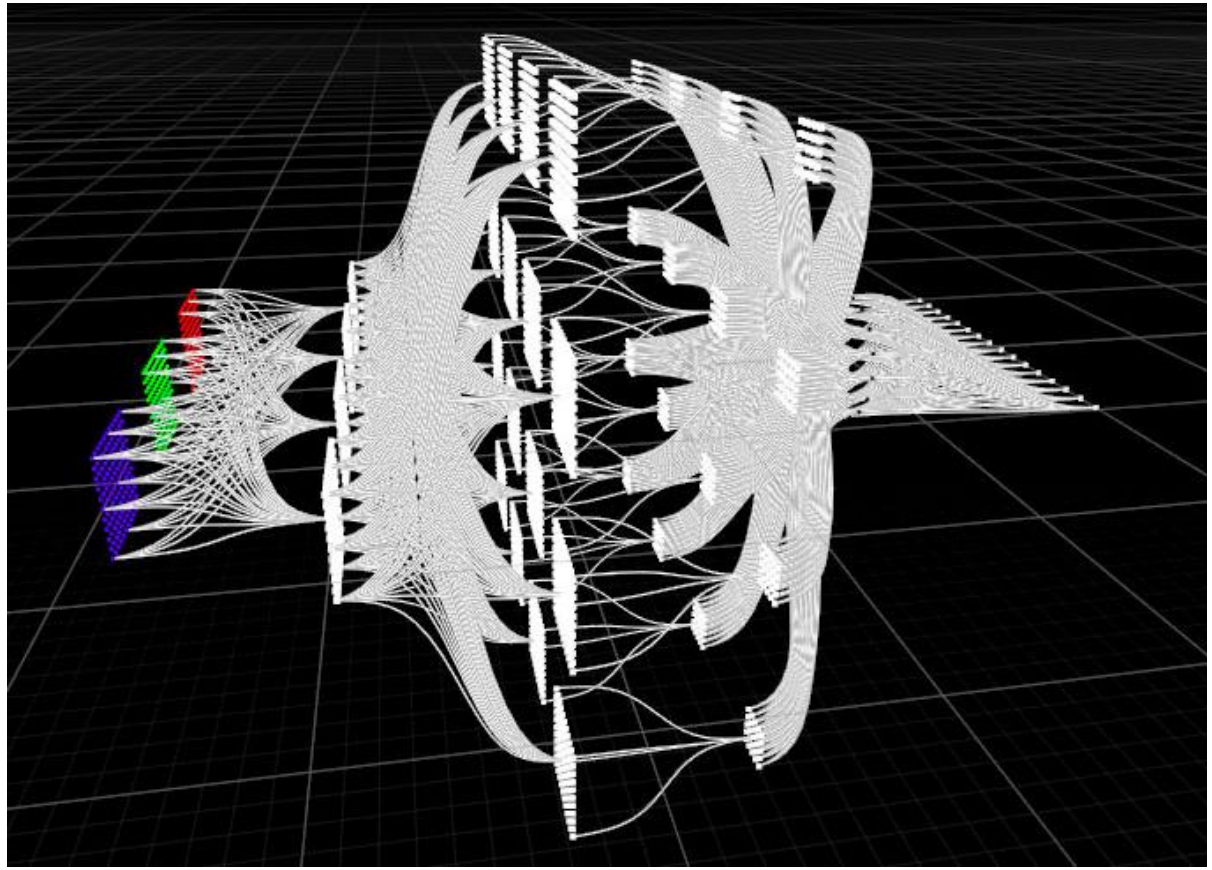
Deep Learning 99
https://teias-courses.github.io/dl99/

Image source

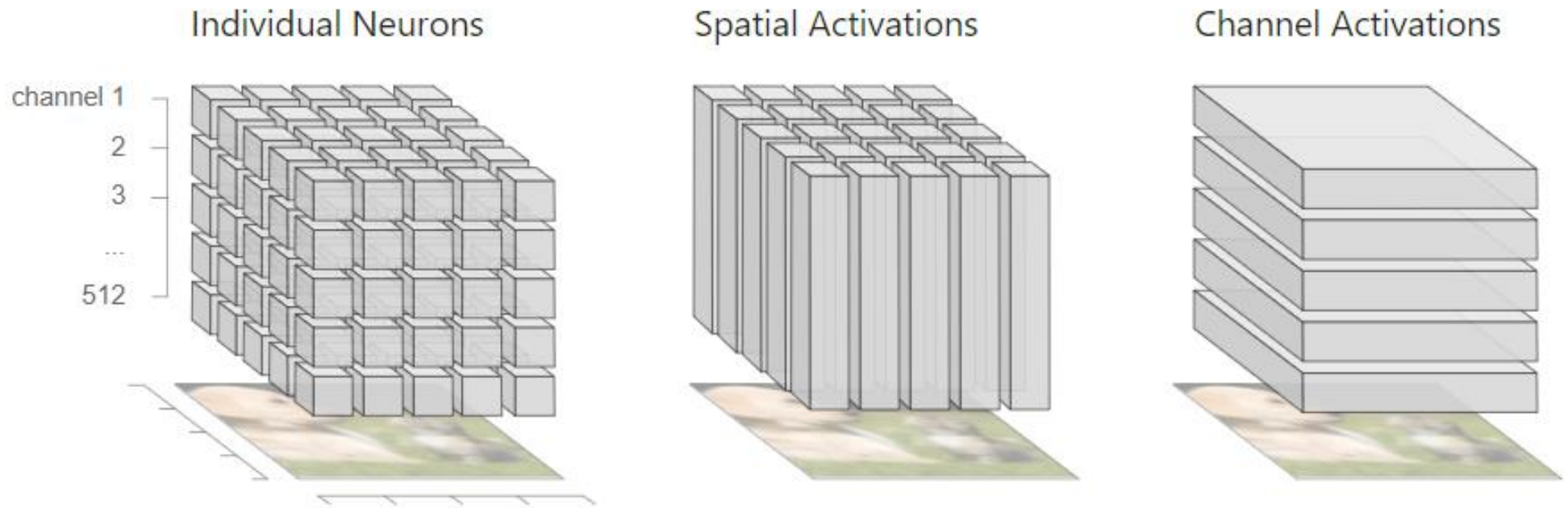
Many of the slides are borrowed from Stanfords's CS230 course and from the distill.pub website

Visualization

- A way of interpreting the outputs of neural networks
- Sheds light on the reasons behind their decision making



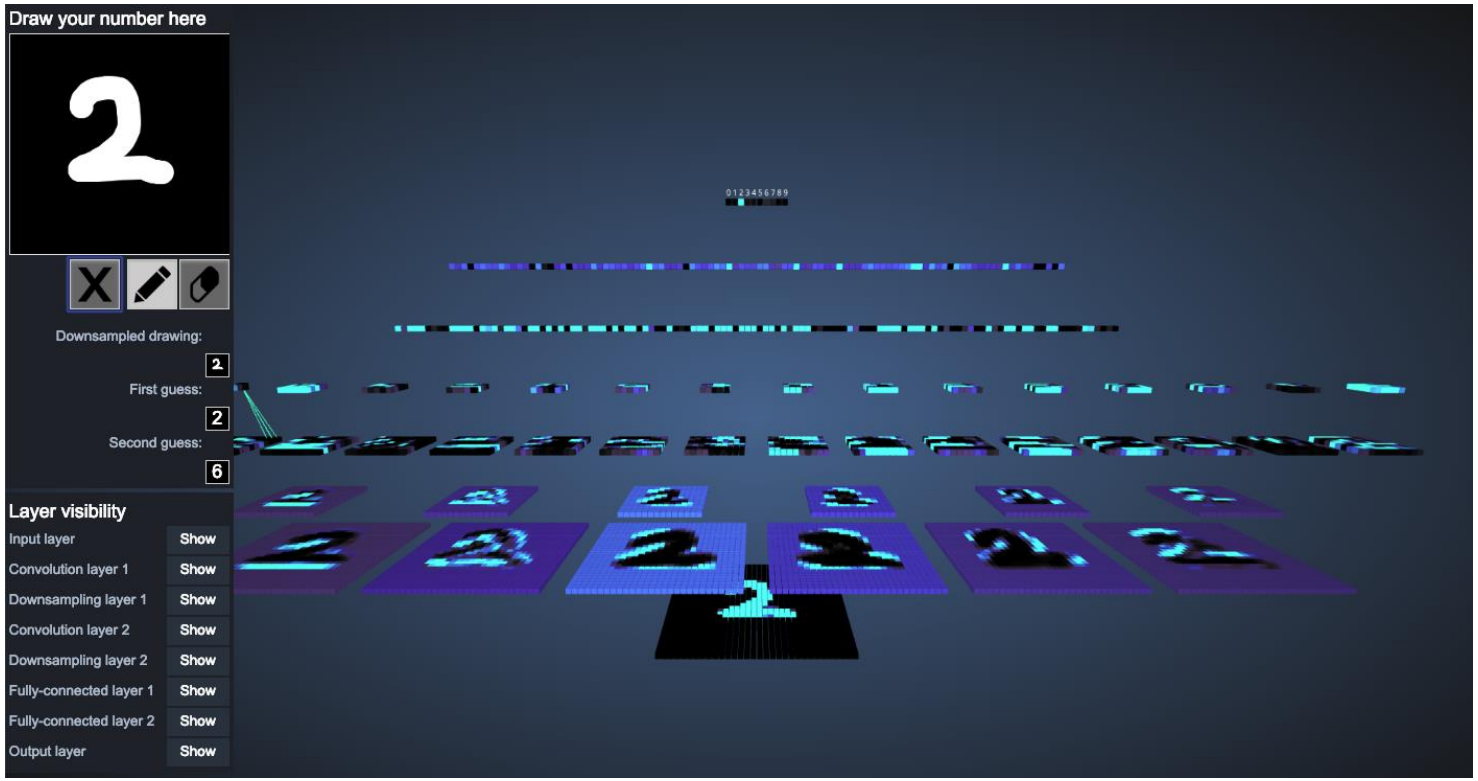
A (hidden) CNN layer



The cube of activations that a neural network for computer vision develops at each hidden layer. Different slices of the cube allow us to target the activations of individual neurons, spatial positions, or channels.

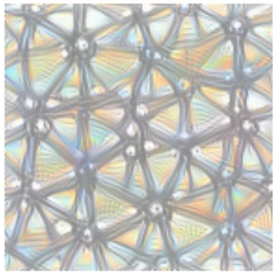
Visualizing intermediate activations

- http://scs.ryerson.ca/~aharley/vis/conv/flat.html

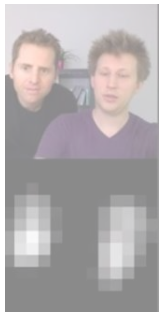


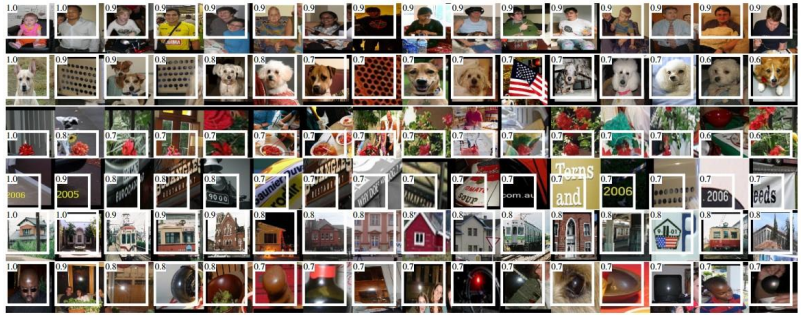
Visualization

Different ways to visualize or interpret NN representations



- Retrieve from real images



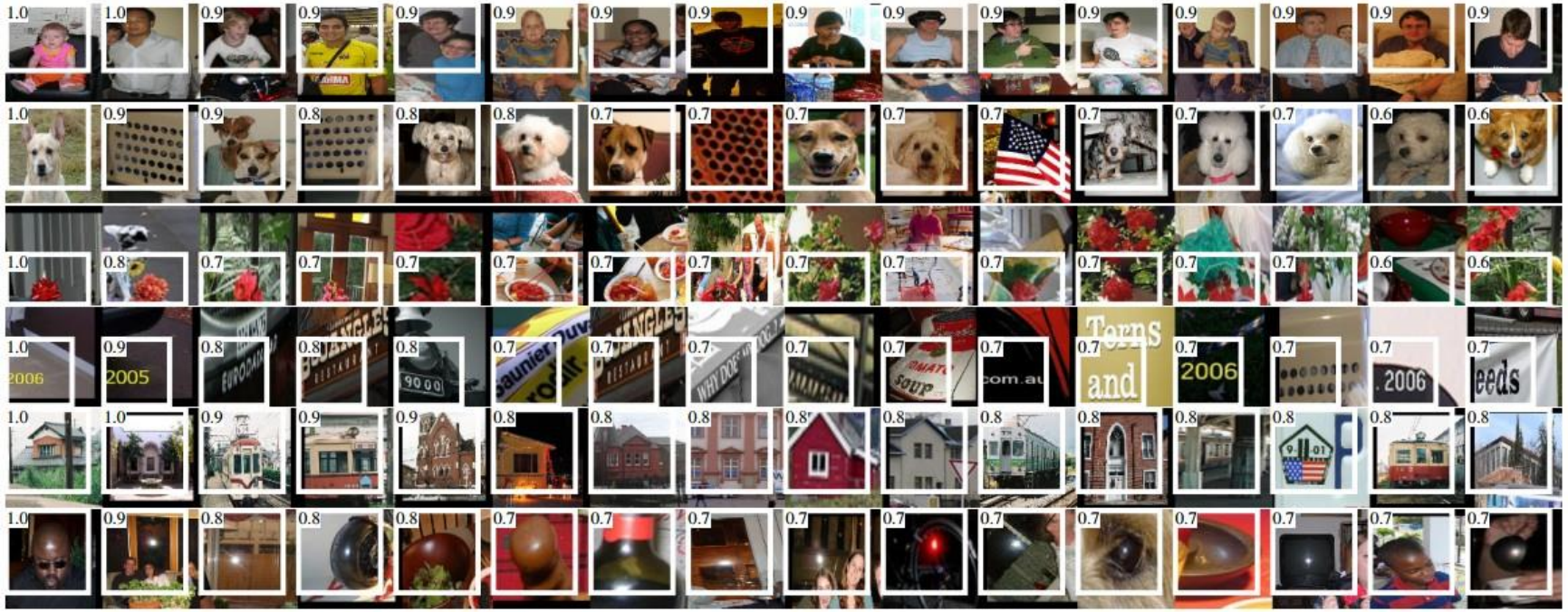


- Visualize layer activations
  - Deconvolution

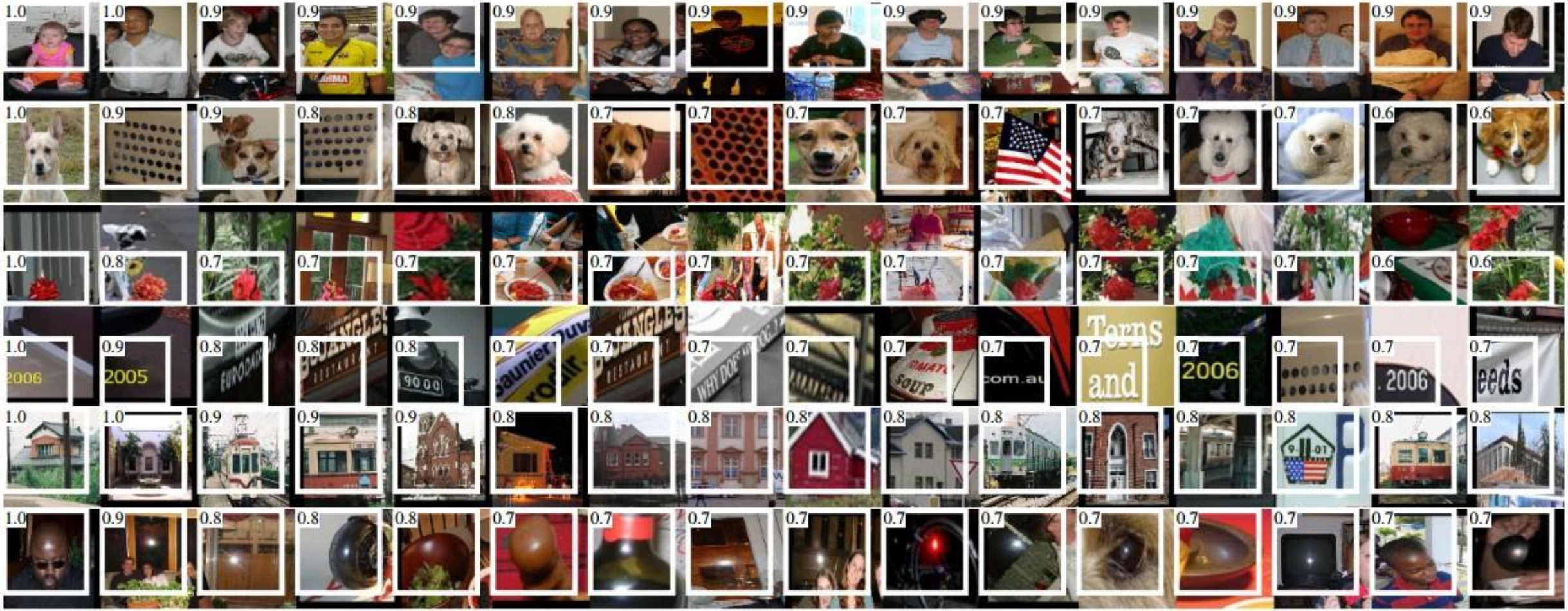
- Feature visualization by optimization

- Attribution
- Dimensionality reduction

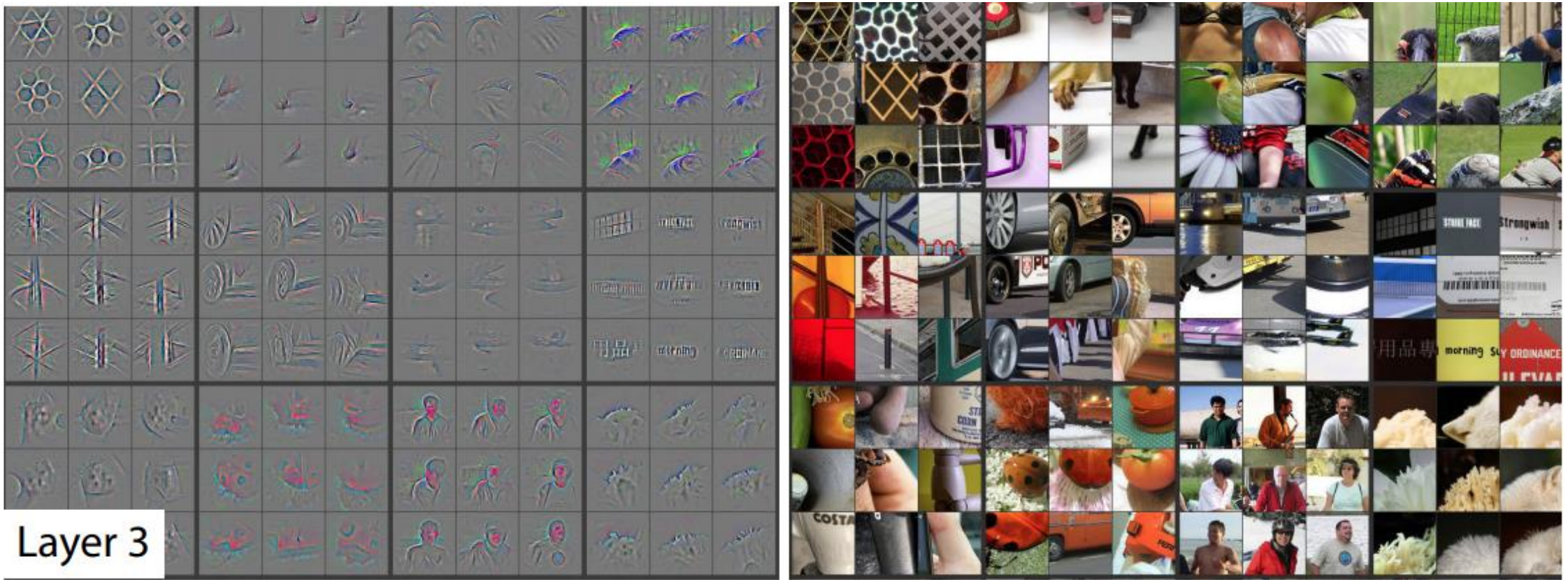
Visualization by Retrieving from a dataset



Girshick et al (2014) -- take a large dataset of images, feed them through the network and keep track of which images maximally activate some neuron

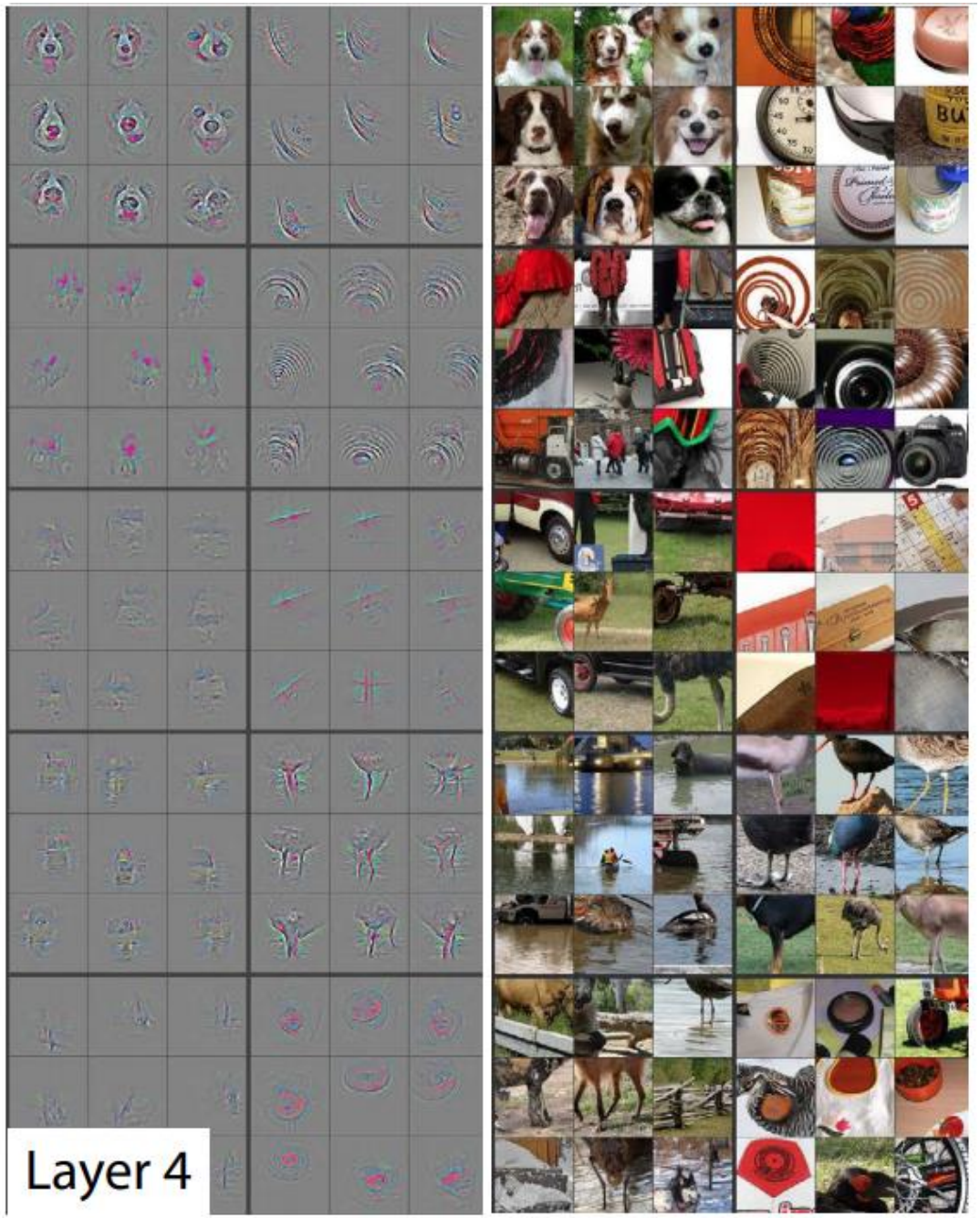


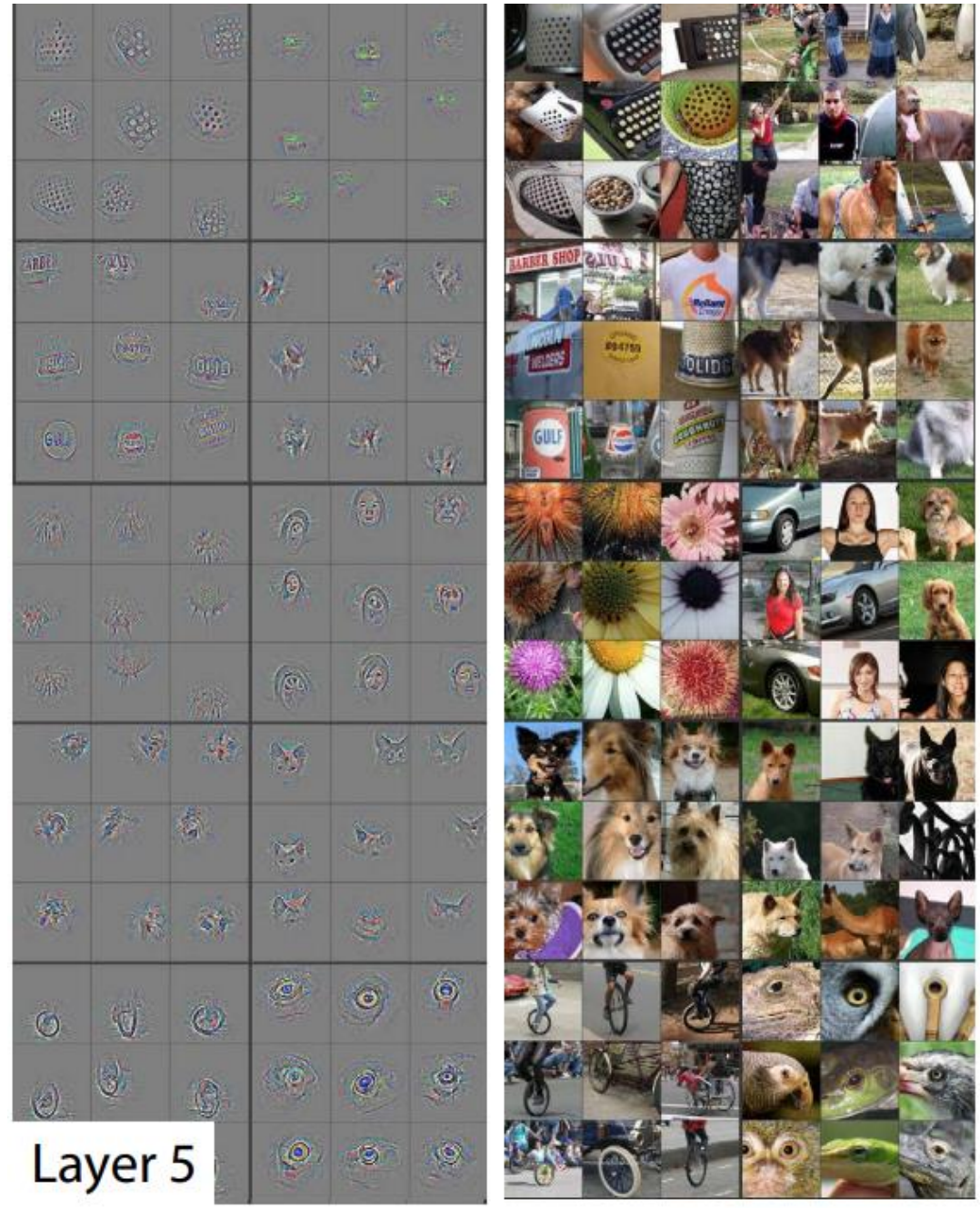
Maximally activating images for some POOL5 (5th pool layer) neurons of an AlexNet. The activation values and the receptive field of the particular neuron are shown in white. (In particular, note that the POOL5 neurons are a function of a relatively large portion of the input image!) It can be seen that some neurons are responsive to upper bodies, text, or specular highlights.



Visualizing and understanding convolutional networks (Zeiler and Fergus, 2013)

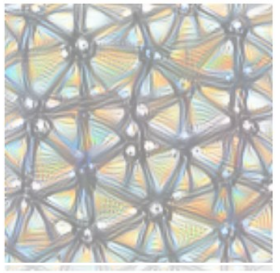
For layers 2-5 we show the top 9 activations in a random subset of feature maps across the validation data


Layer 4

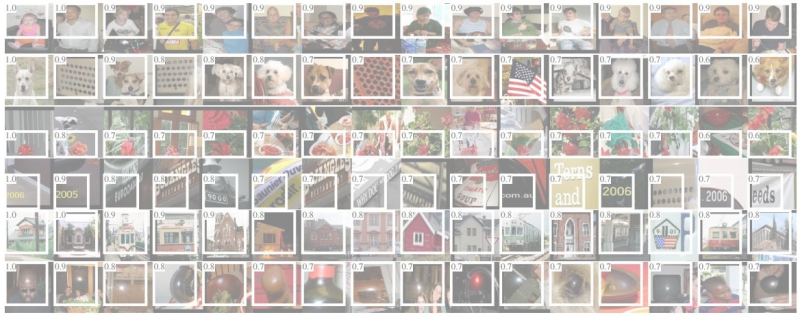

Layer 5

Visualization

Different ways to visualize or interpret NN representations

- Retrieve from real images

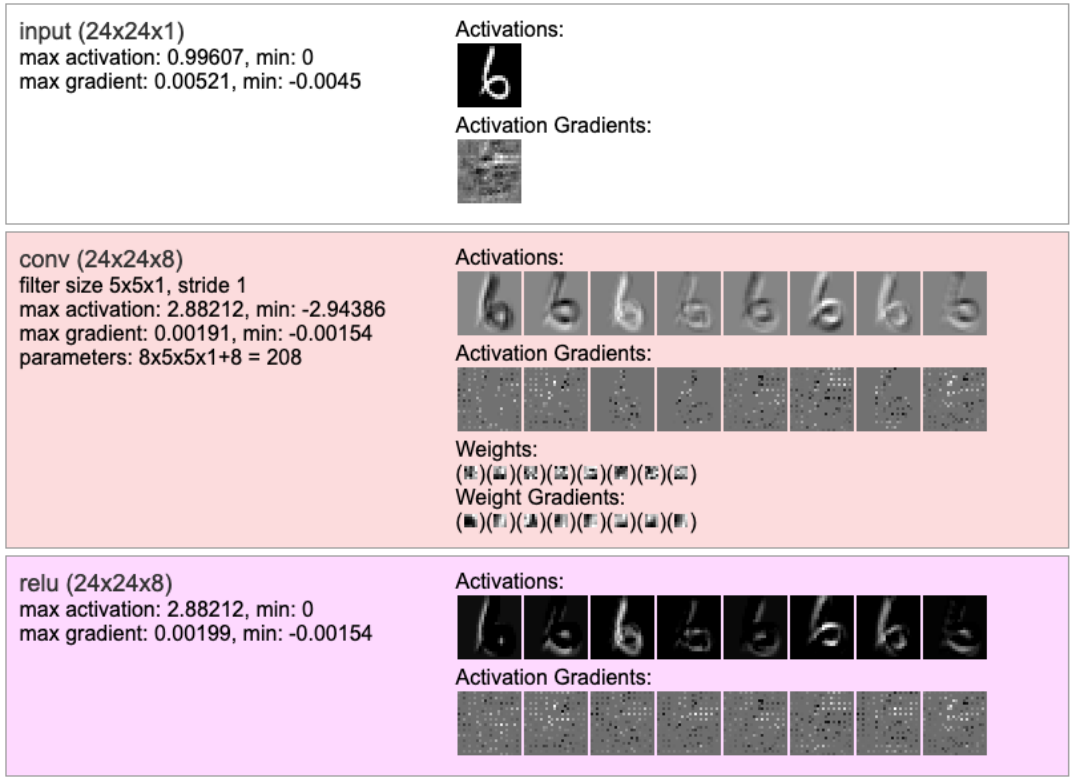



- Visualize layer activations
  - Deconvolution
- Feature visualization by optimization
- Attribution
- Dimensionality reduction

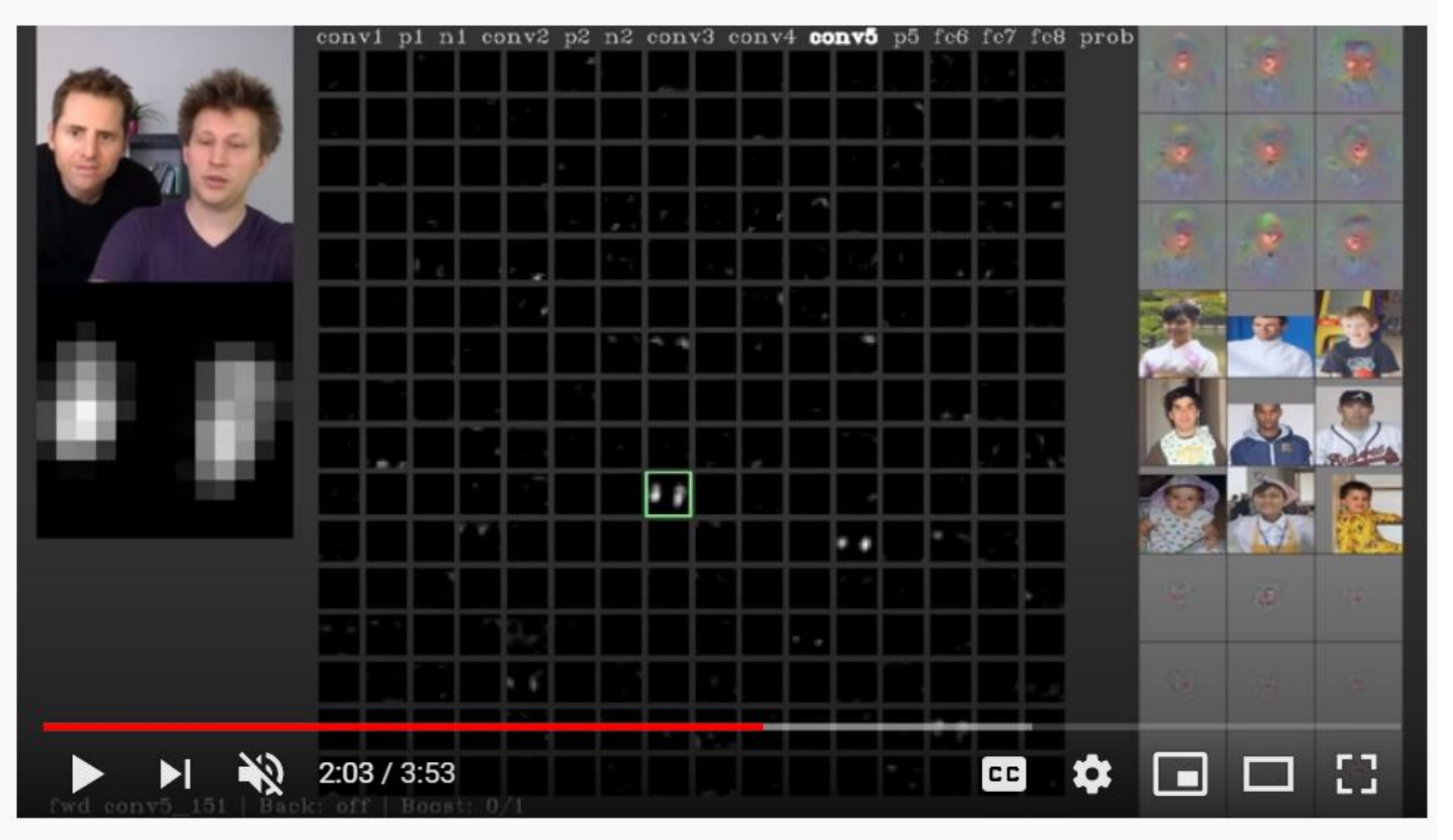


Visualizing intermediate activations

- https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html



Visualizing intermediate activations



https://www.youtube.com/watch?v=AgkfIQ4IGaM

Deep Visualization Toolbox

yosinski.com/deepvis

#deepvis

Jason Yosinski

Jeff Clune

Anh Nguyen

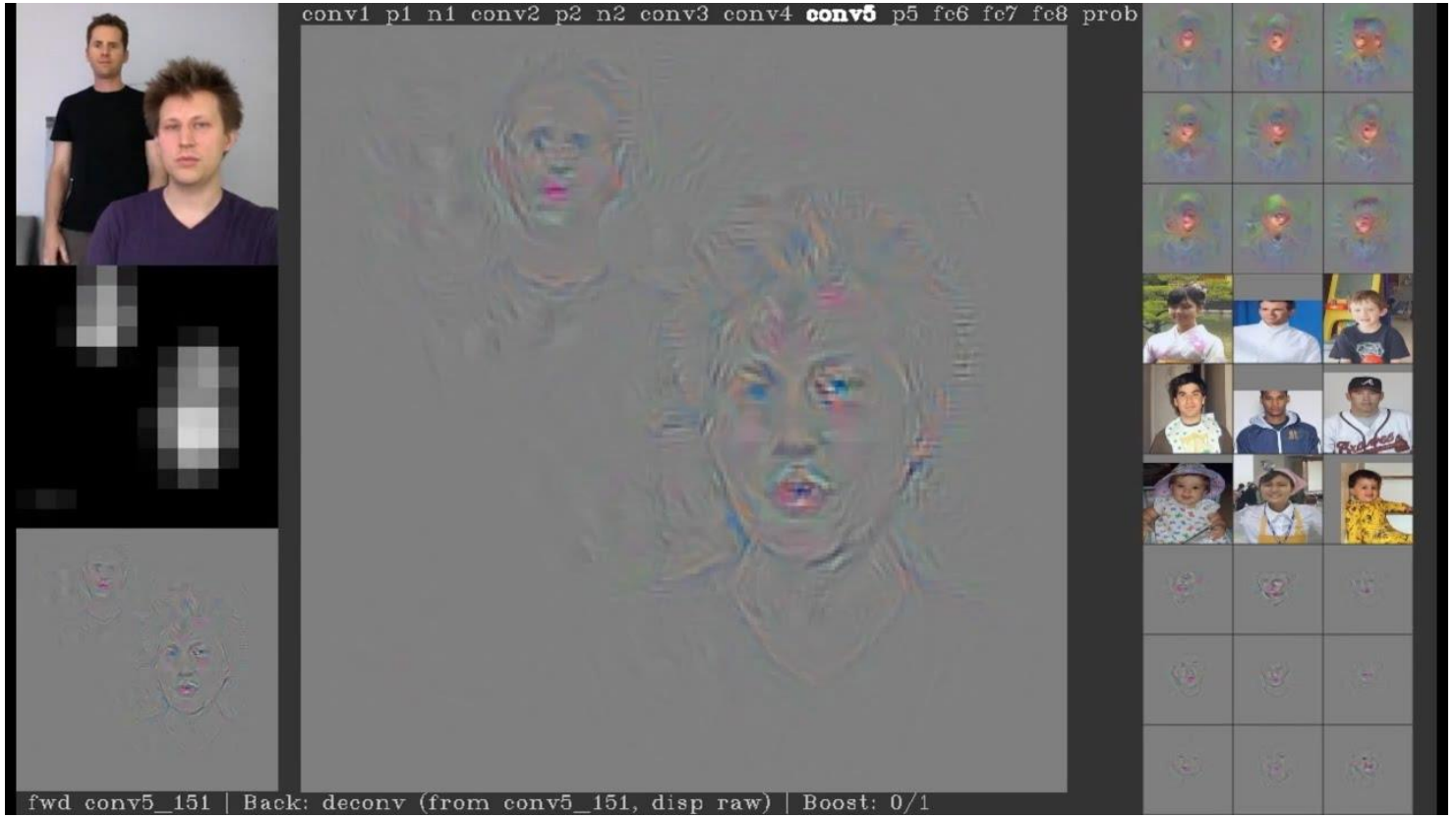
Thomas Fuchs

Hod Lipson

Cornell University

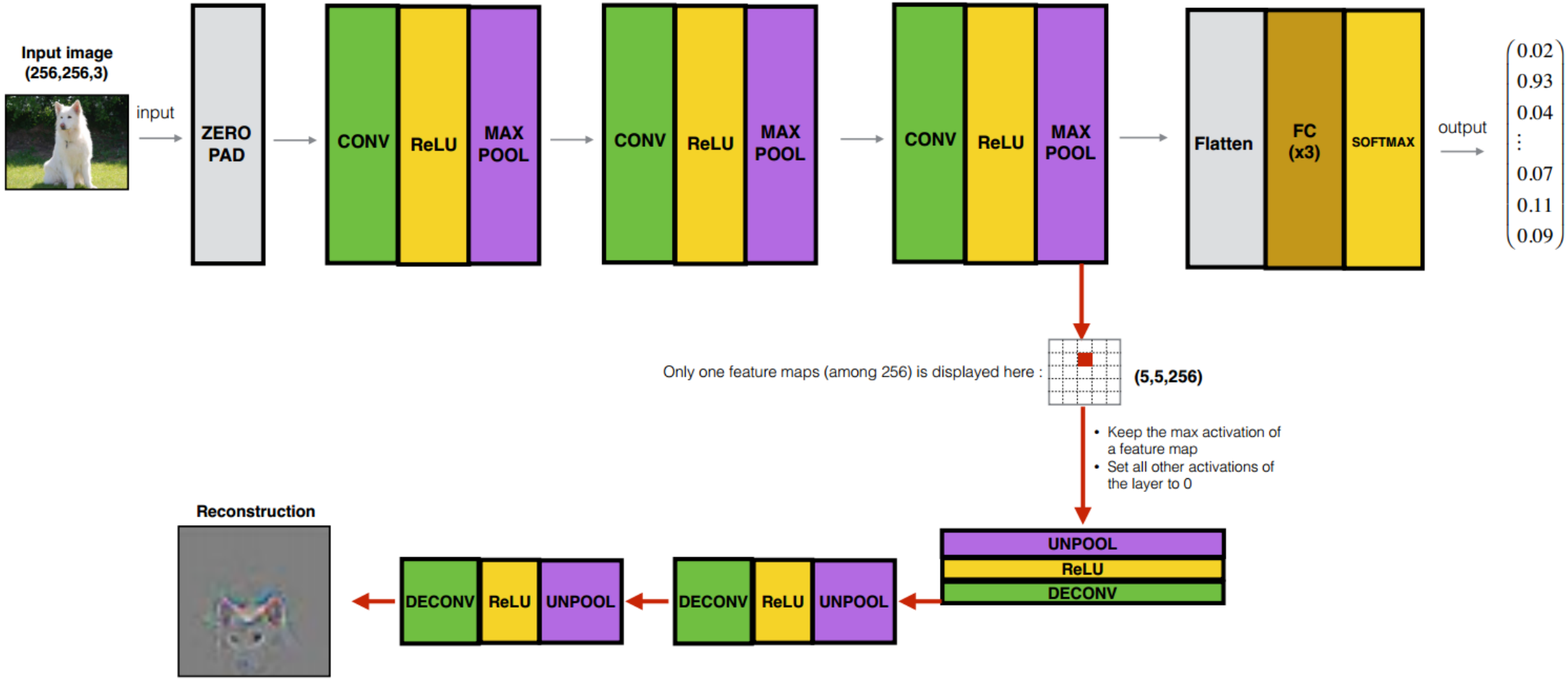
University of Wyoming

Jet Propulsion Laboratory
California Institute of Technology



Deconvolution for activation visualization

Deconvolution

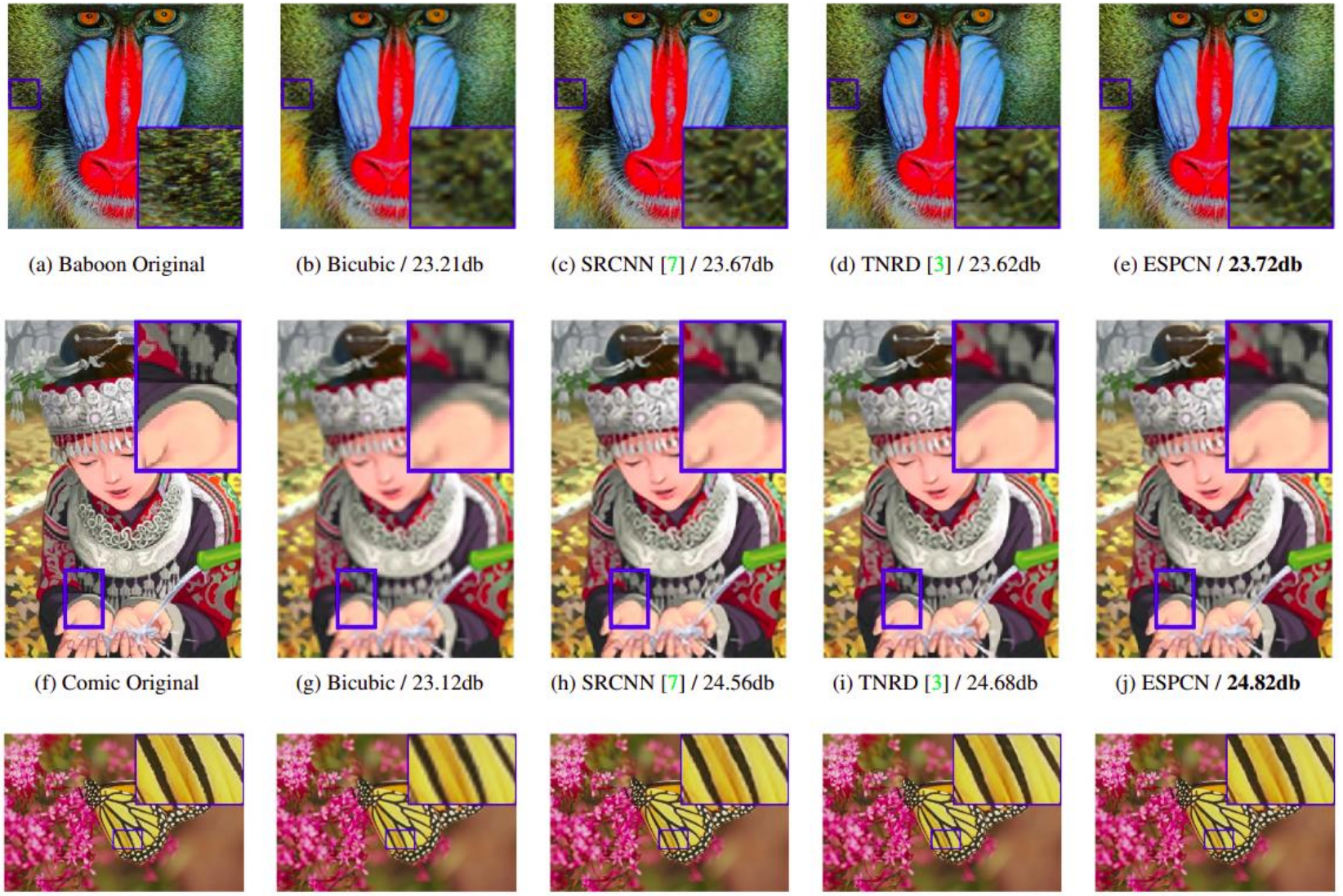


[Matthew D. Zeiler and Rob Fergus (2013): Visualizing and Understanding Convolutional Networks]

Kian Katanforoosh

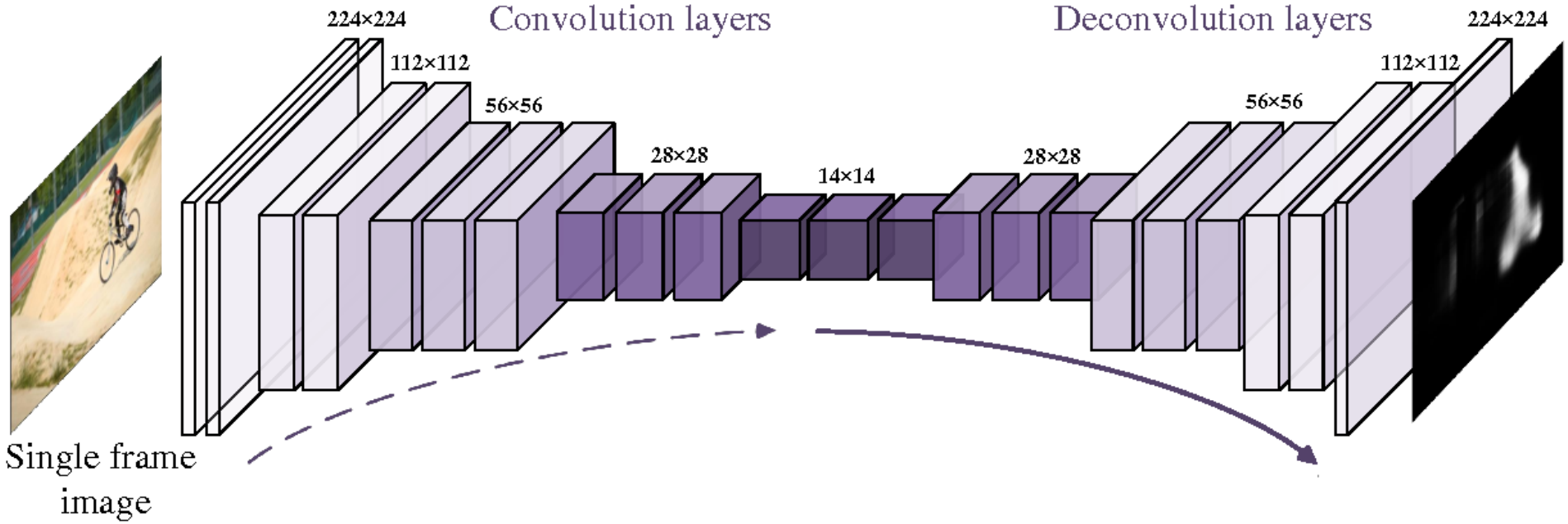
Transposed convolution

- Super-resolution



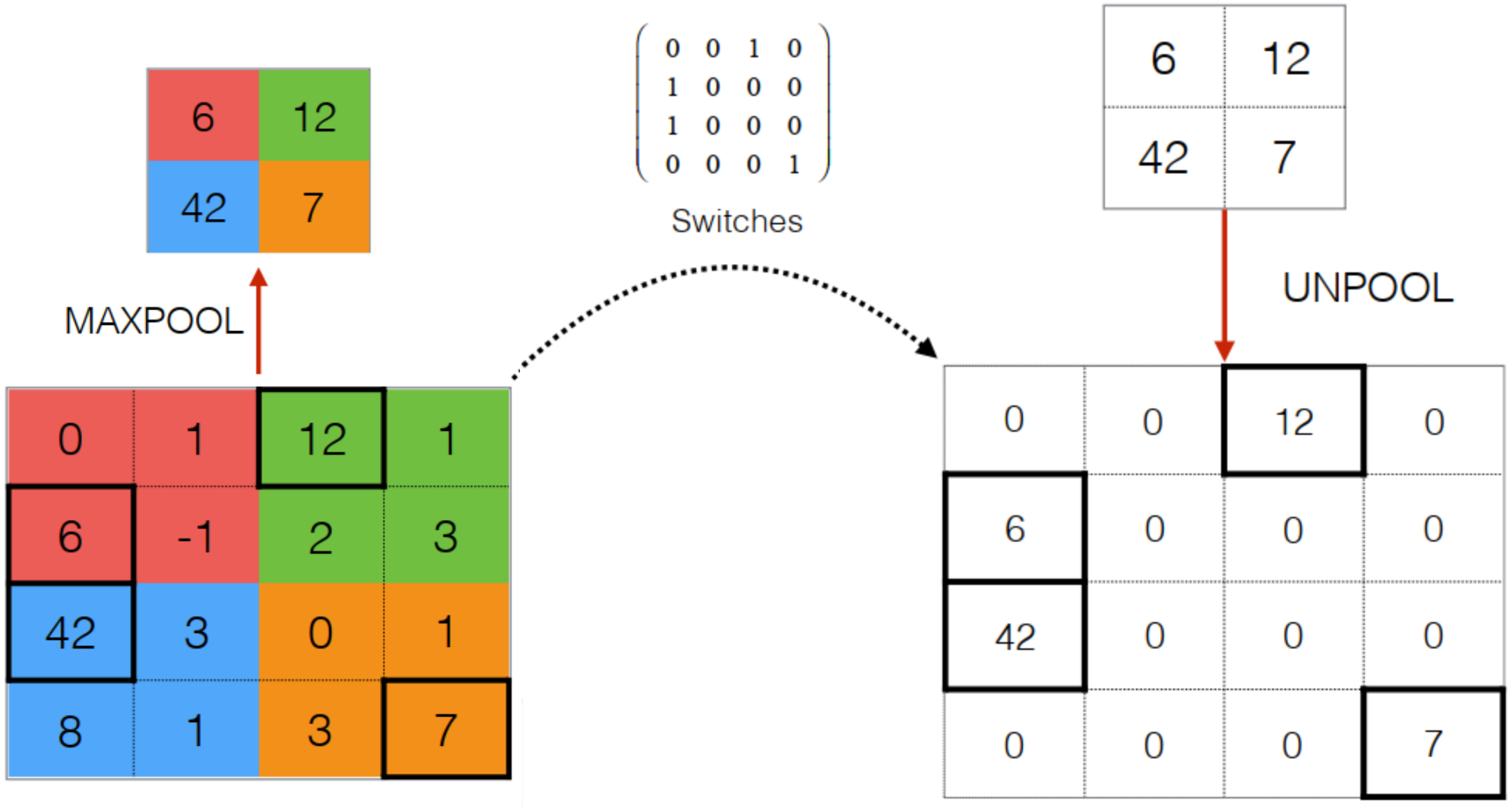
Shi et al (2016)
source

Transposed convolution (in visualization)

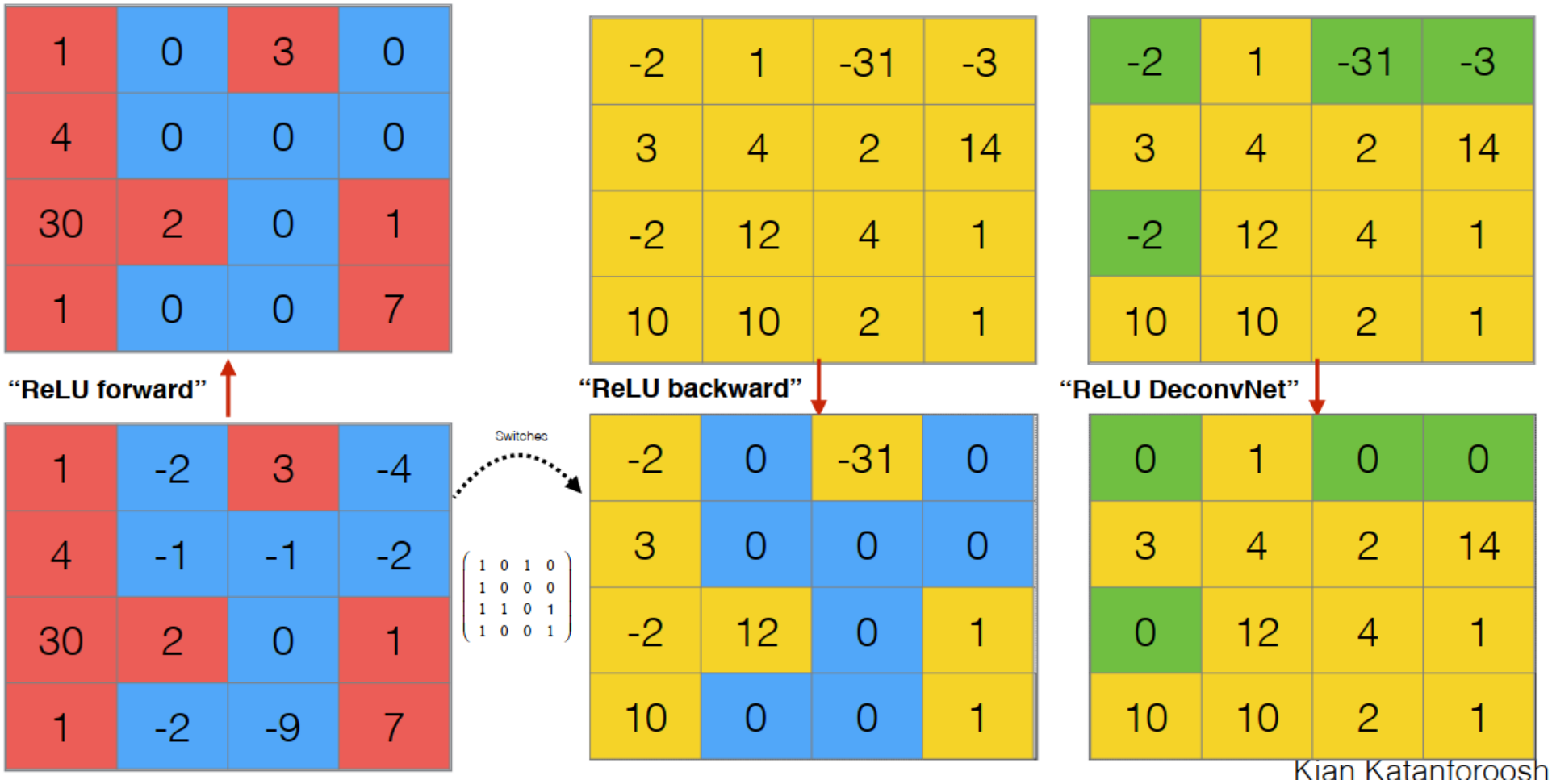


source

Unpooling


6 12
42 7
MAXPOOL
0 1 12 1
6 -1 2 3
42 3 0 1
8 1 3 7
0 0 1 0
1 0 0 0
1 0 0 0
0 0 0 1
Switches
6 12
42 7
UNPOOL
0 0 12 0
6 0 0 0
42 0 0 0
0 0 0 7

Backward ReLU



Kian Katanforoosh

Convolution as a normal matrix multiplication

1	2	3
6	5	3
1	4	1

3x3 Input

1	2
2	1

2x2 Kernel

1	2	0	2	1	0	0	0	0
0	1	2	0	2	1	0	0	0
0	0	0	1	2	0	2	1	0
0	0	0	0	1	2	0	2	1

x

1
2
3
6
5
3
1
4
1

=

22
21
22
20

22	21
22	20

Image source

From output back to input

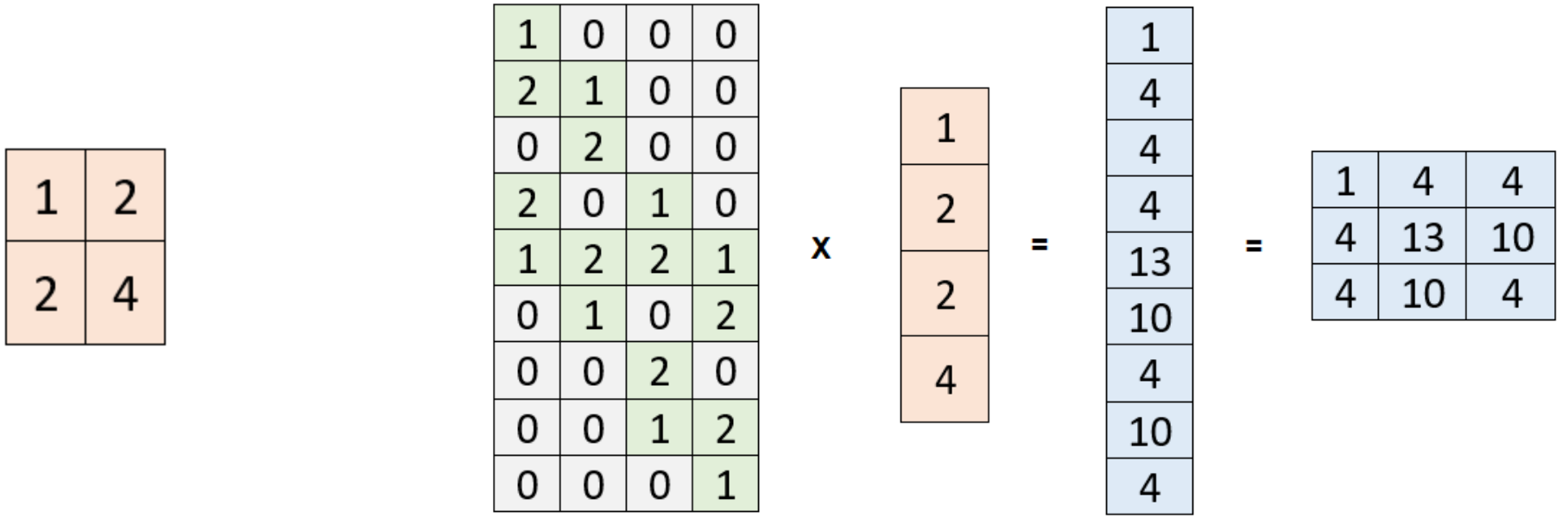


Image [source]

Sub-pixel convolution

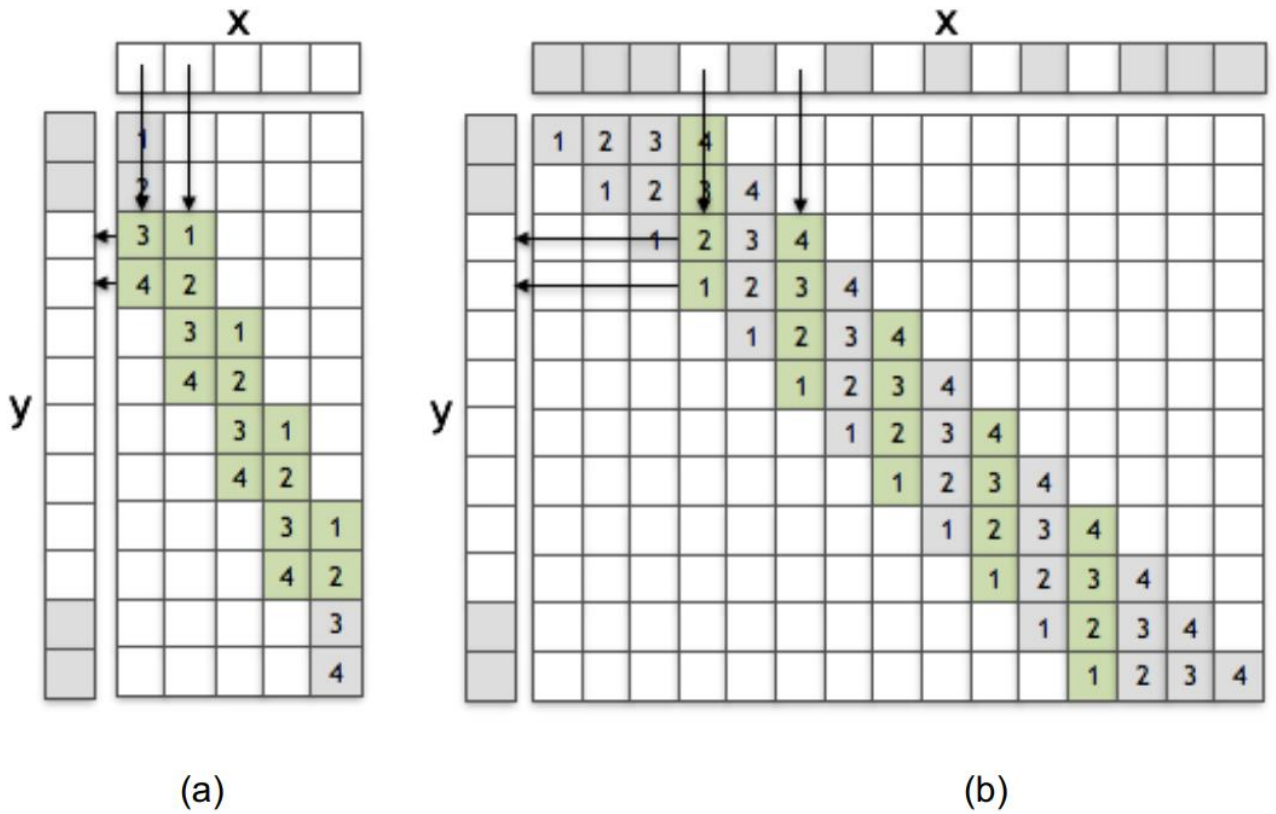
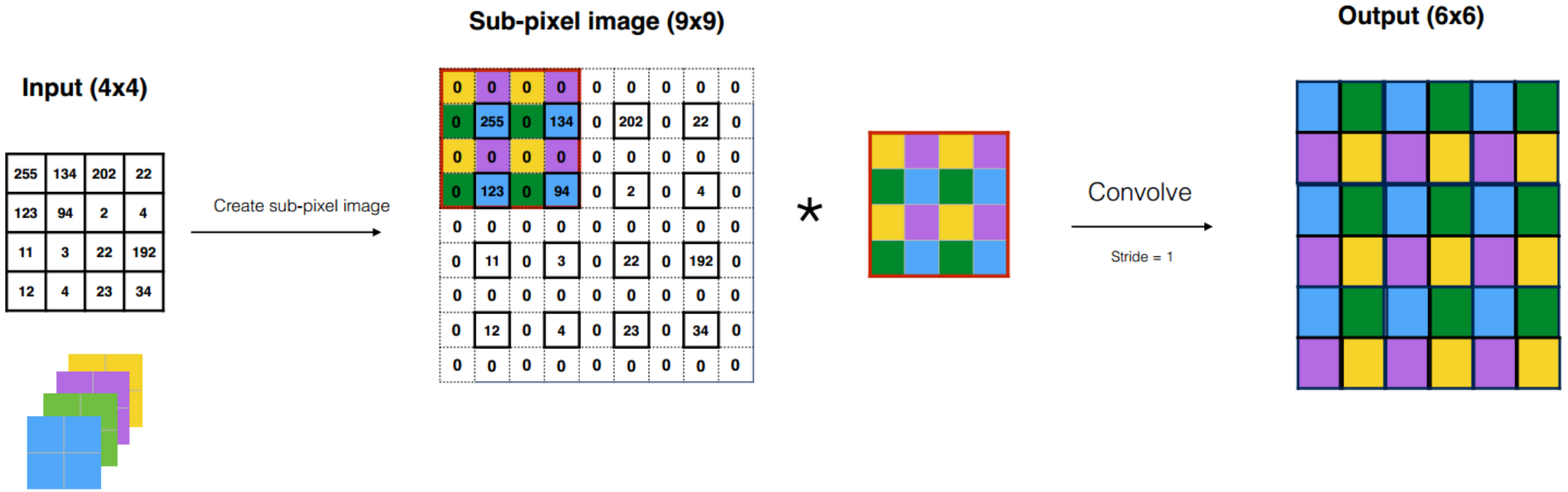


Figure 3: (a) Transposed convolution with stride 2 and (b) sub-pixel convolution with stride ½ in 1D

Shi et al: Is the deconvolution layer the same as a convolutional layer?

Sub-pixel convolution



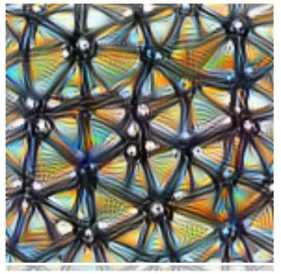
This allows us to upsample an encoding into an image.

[Hongyang Gao et al. : Pixel Deconvolutional Networks]
[Matthew Zeiler et al.: Deconvolutional Networks]
[Vincent Dumoulin and Francesco Visin : A guide to convolution arithmetic for deep learning]
[Wenzhe Shi, et al. : Is the deconvolution layer the same as a convolutional layer?]

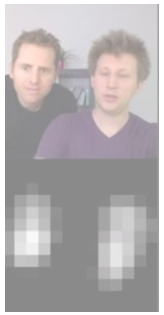
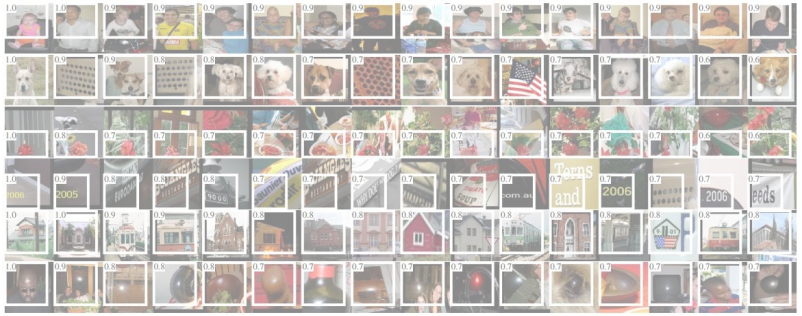
Kian Katanforoosh

Visualization

Different ways to visualize or interpret NN representations

- Retrieve from real images
- Visualize layer activations
  - Deconvolution
- **Feature visualization** by optimization
- Attribution
- Dimensionality reduction


Feature visualization by optimization

Different **optimization objectives** show what different parts of a network are looking for.

`n` layer index
`x,y` spatial position
`z` channel index
`k` class index

Neuron
$layer_n[x,y,z]$

Channel
$layer_n[:,:,z]$

Layer/DeepDream
$layer_n[:,:,:]^2$

Class Logits
`pre_softmax[k]`

Class Probability
`softmax[k]`

Source of this set of slides: distill.pub

Feature visualization by optimization

- Repeat:
  - Forward propagate image x
  - Compute the objective L
  - Backpropagate to get dL/dx
  - Update x's pixels with gradient ascent

Class Logits

`pre_softmax[k]`

$$L = s_{dog}(x) - \lambda \left\| x \right\|_2^2$$

$$x = x + \alpha \frac{\partial L}{\partial x}$$

Feature visualization by optimization

Brown bear

Pug

Saxophone

Source: [Visualizing GoogLeNet Classes]

Why visualize by optimization?

Dataset Examples show us what neurons respond to in practice

Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.

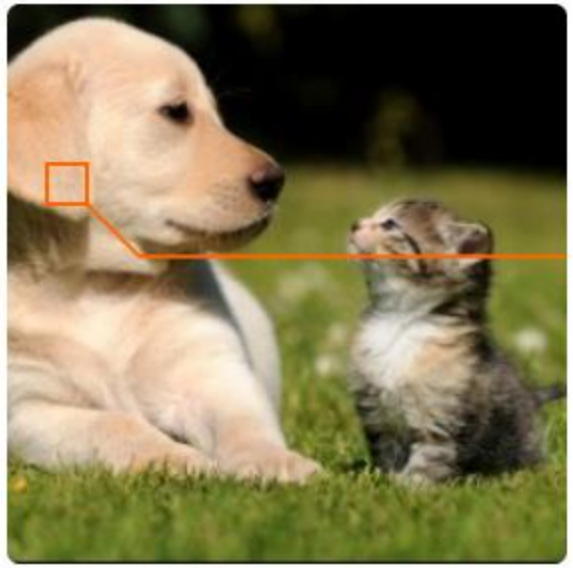
Baseball—or stripes?
mixed4a, Unit 6

Animal faces—or snouts?
mixed4a, Unit 240

Clouds—or fluffiness?
mixed4a, Unit 453

Buildings—or sky?
mixed4a, Unit 492

Semantic dictionary

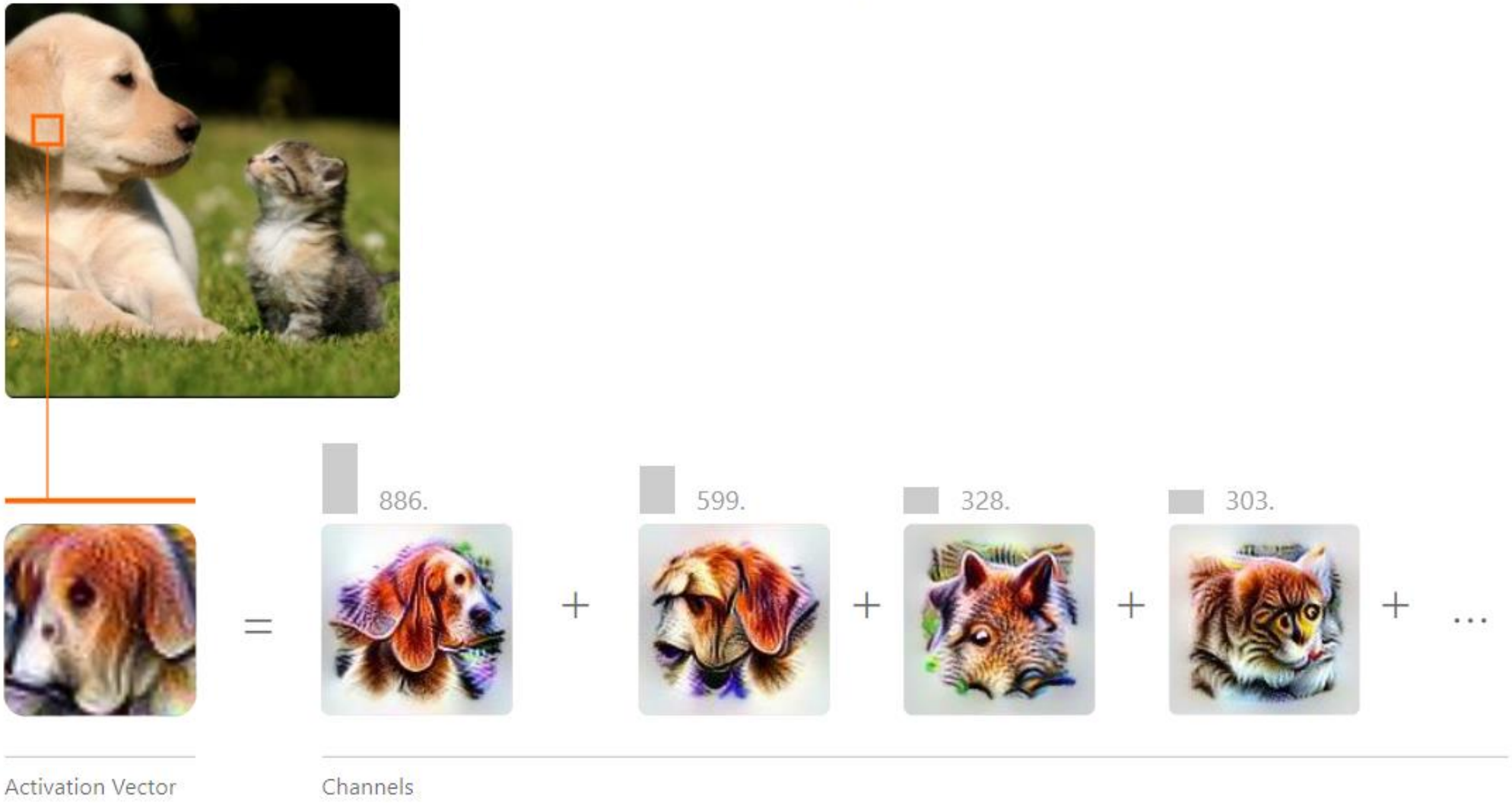


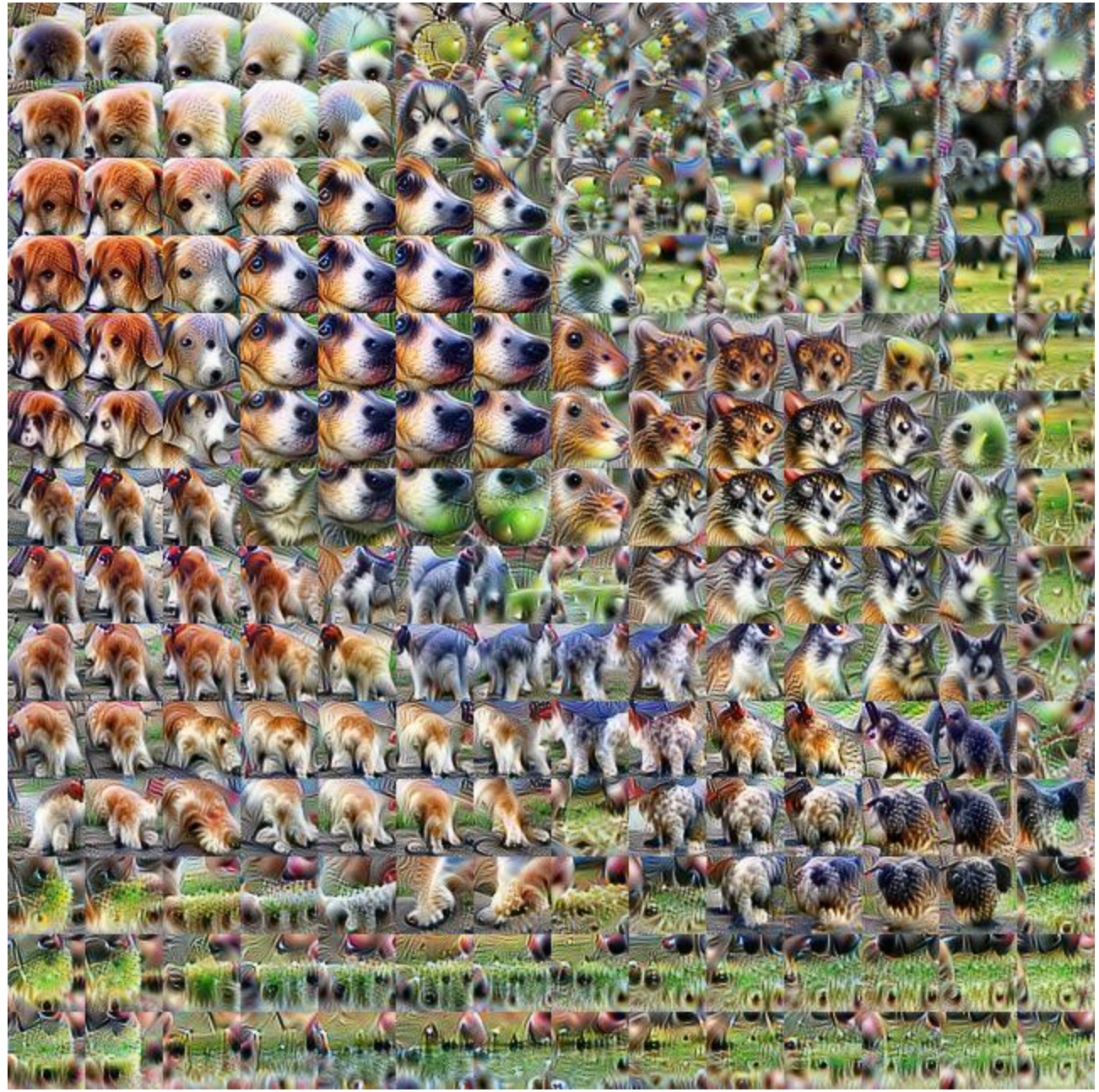
Making sense of these activations is hard because we usually work with them as abstract vectors:

> $a_{4,1}$ = [0, 0, 0, 25.2, 164.1, 0, 42.7, 4.51, 115.0, 51.3, 0, 0,...

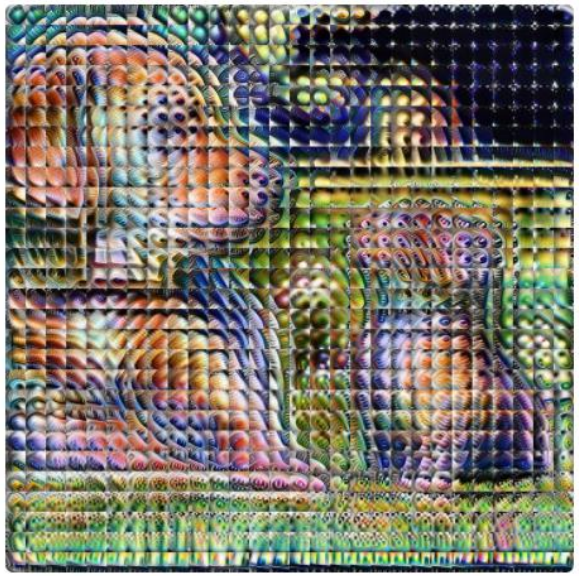
With feature visualization, however, we can transform this abstract vector into a more meaningful "semantic dictionary".

Visualizing spatial activations

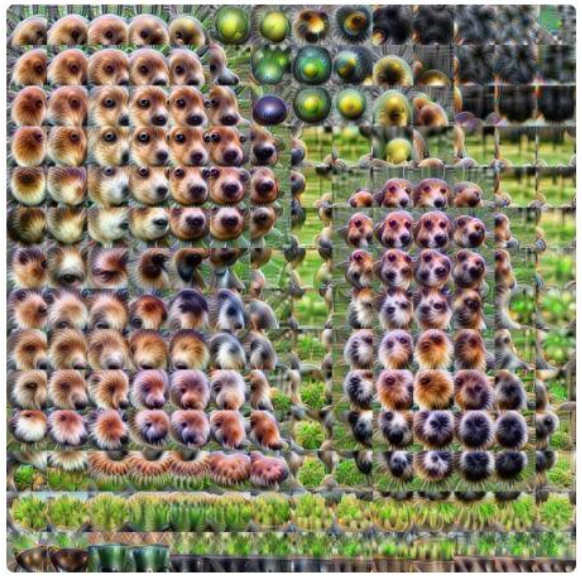




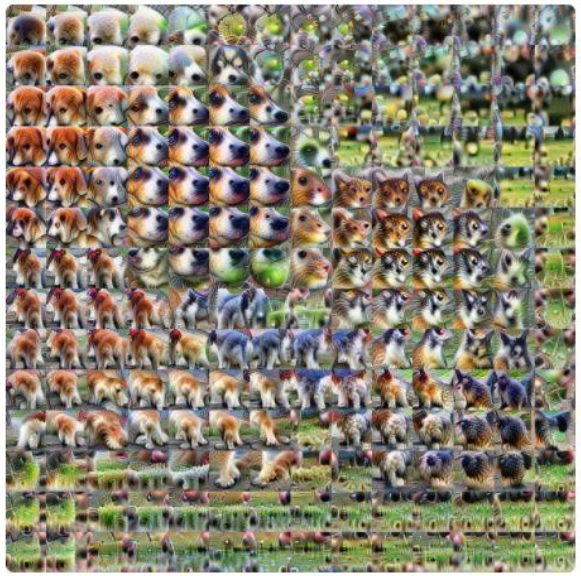
Evolving understanding of the network



MIXED3A



MIXED4A

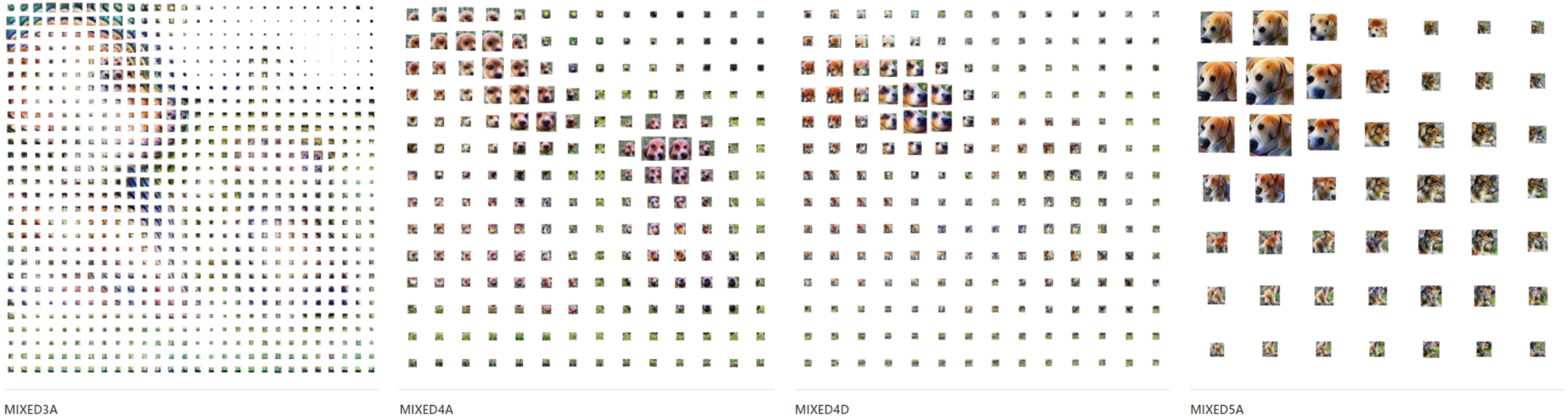


MIXED4D

MIXED5A

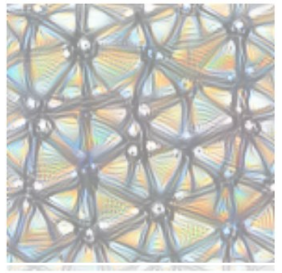
Evolving understanding of the network

Normalized by the magnitude of the activations

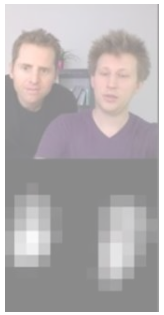


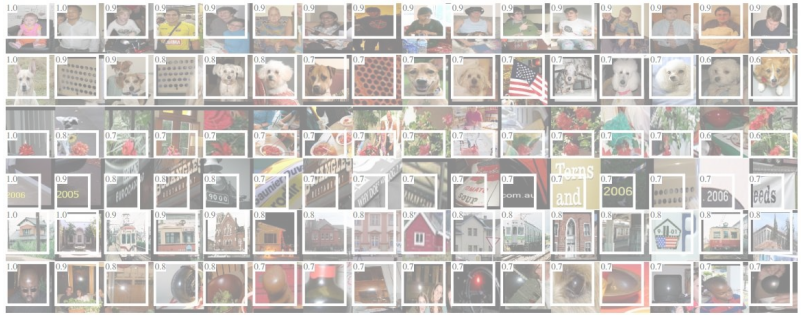
Visualization

Different ways to visualize or interpret NN representations

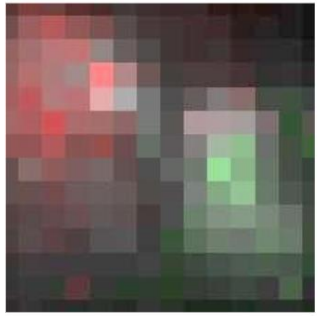


- Retrieve from real images



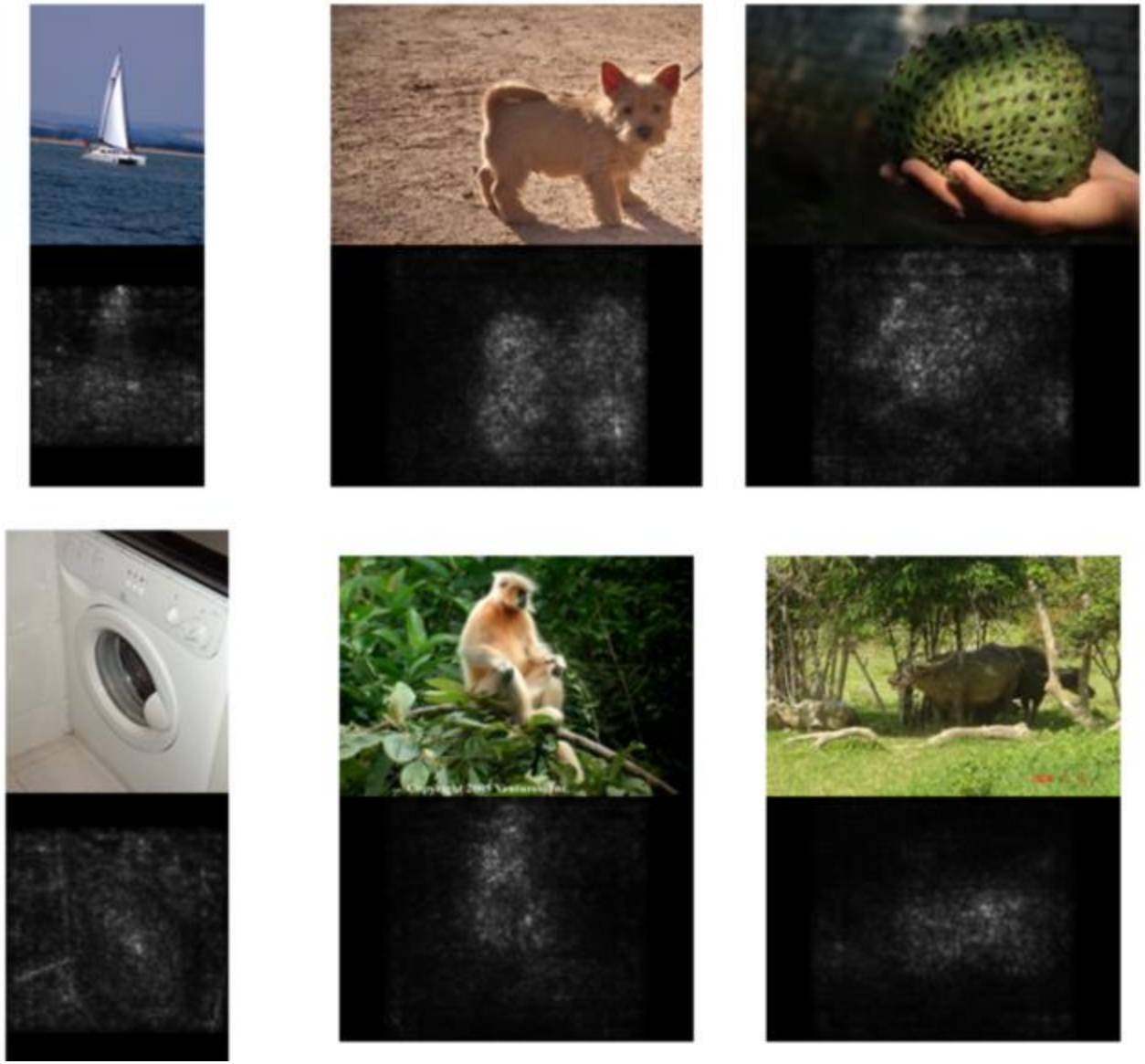


- Visualize layer activations
  - Deconvolution
- Feature visualization by optimization



- Attribution
- Dimensionality reduction

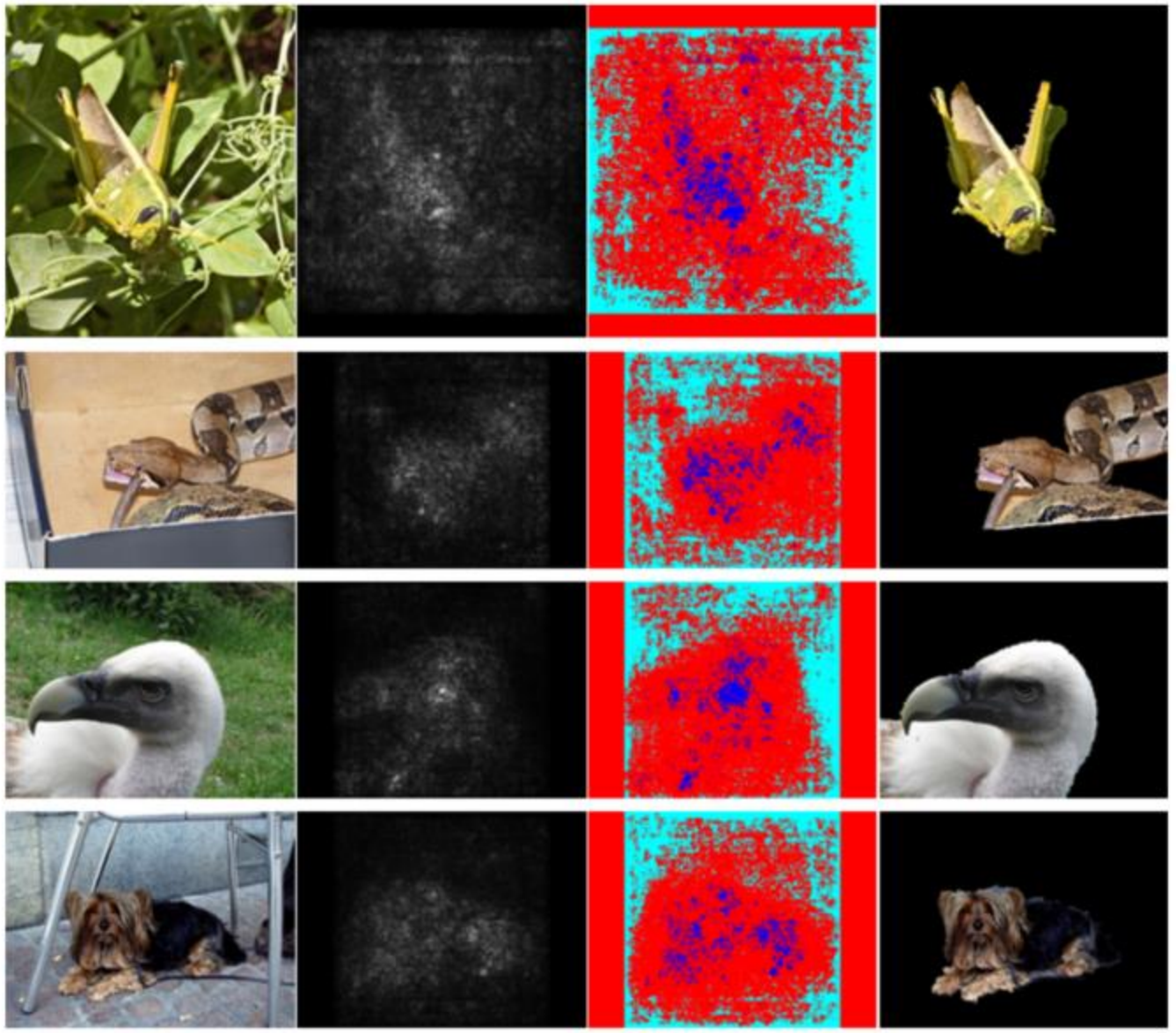
Saliency map



$$w = \left. \frac{\partial S_c}{\partial I} \right|_{I_0}$$

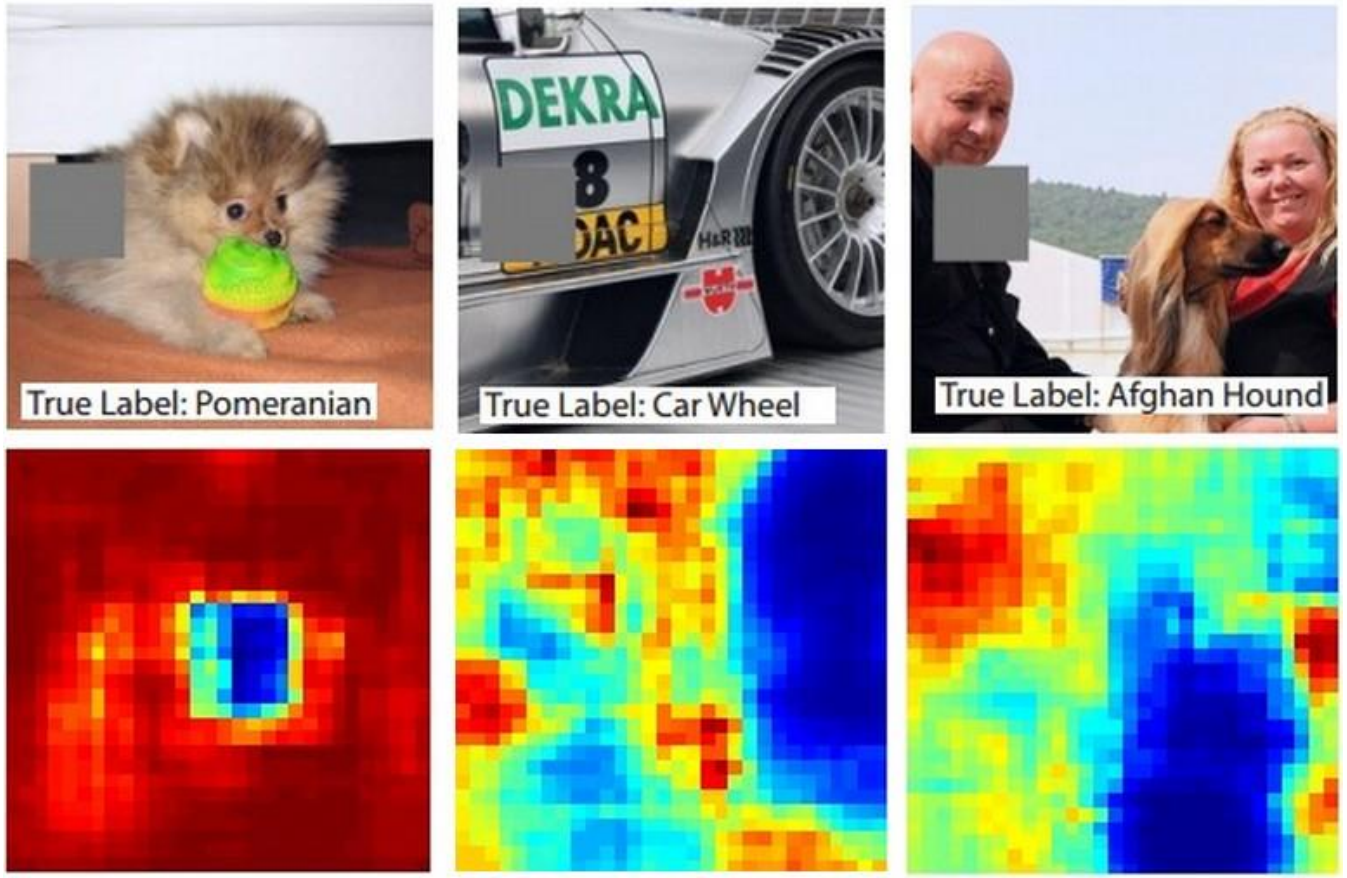
Simonyan et al (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

Saliency map



Simonyan et al (2014): Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

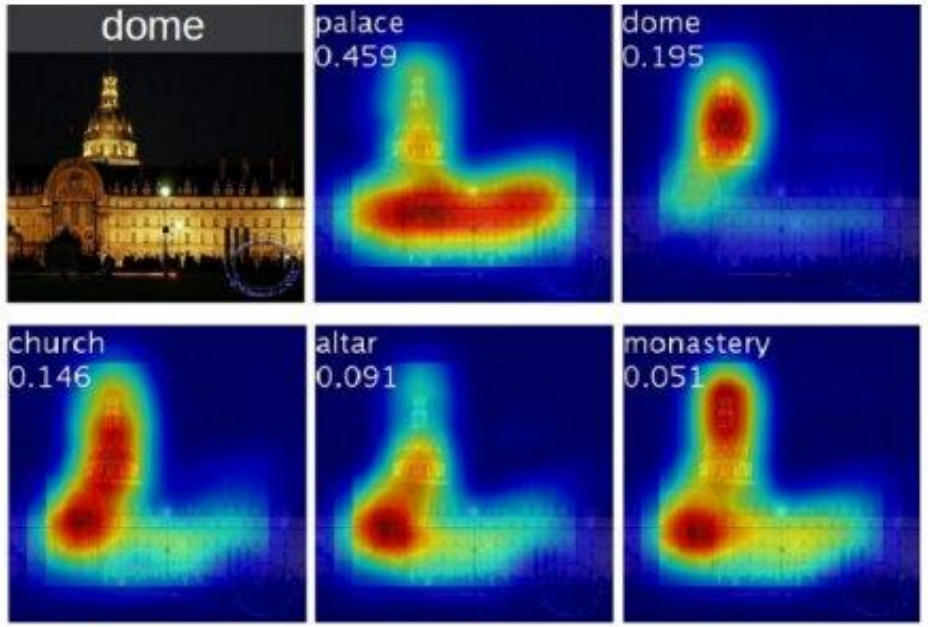
Occlusion-based saliency map



Zeiler and Fergus (2013): Visualizing and Understanding Convolutional Networks

Heatmaps of class activation

- Class activation map (CAM) visualization

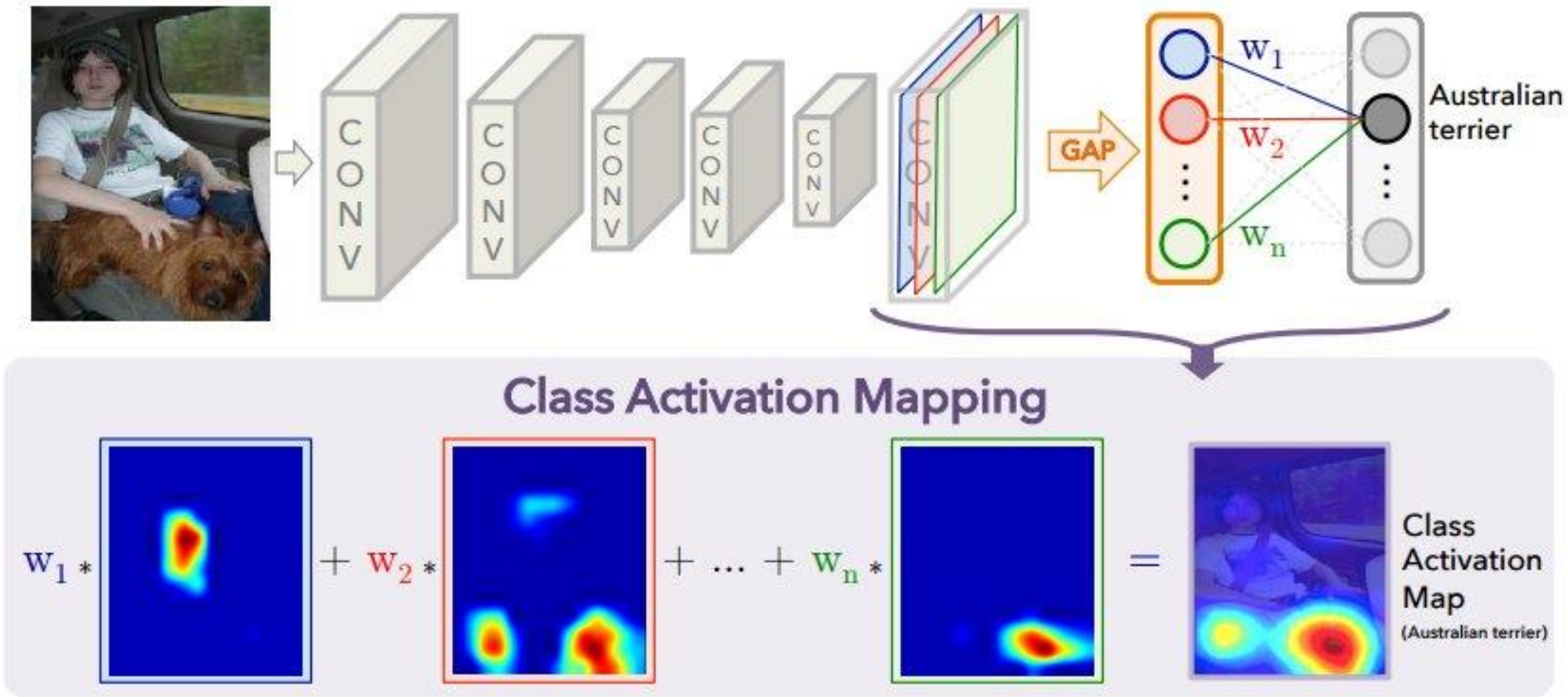


Class activation maps of top 5 predictions

Class activation maps for one object class

https://github.com/metalbubble/CAM

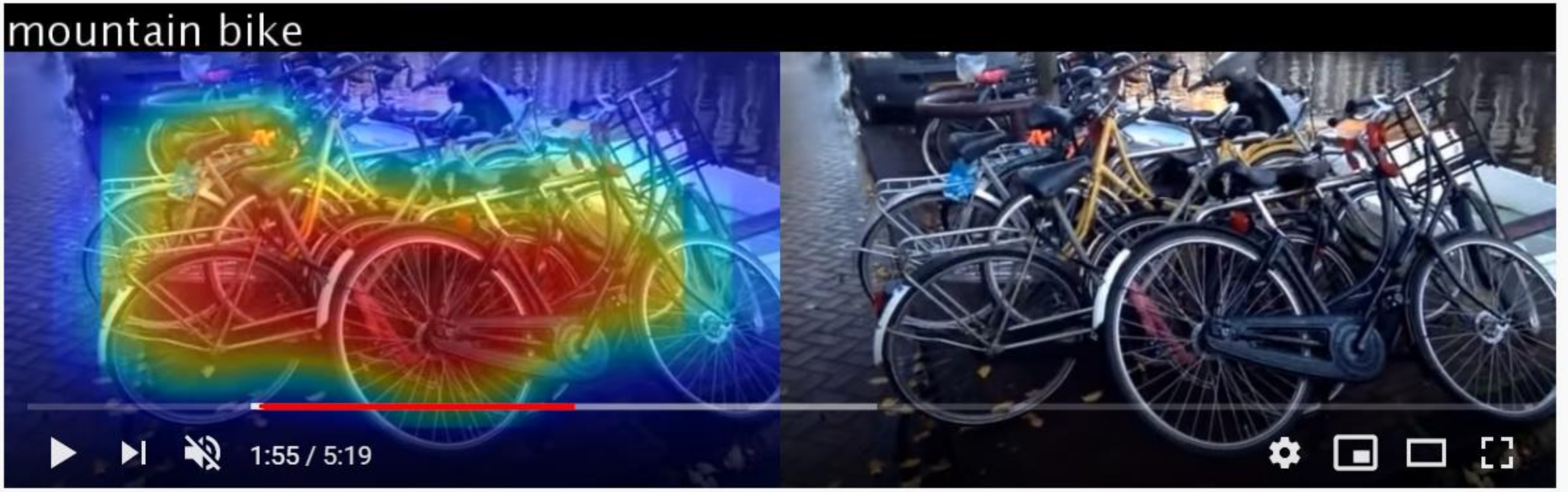
Heatmaps of class activation



https://github.com/metalbubble/CAM

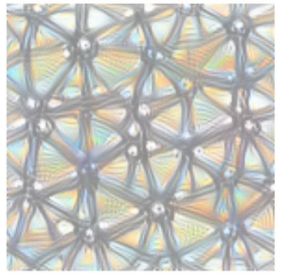
Watch more

https://www.youtube.com/watch?v=fZvOy0VXWAI

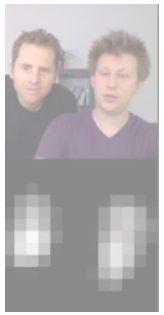


Visualization

Different ways to visualize or interpret NN representations



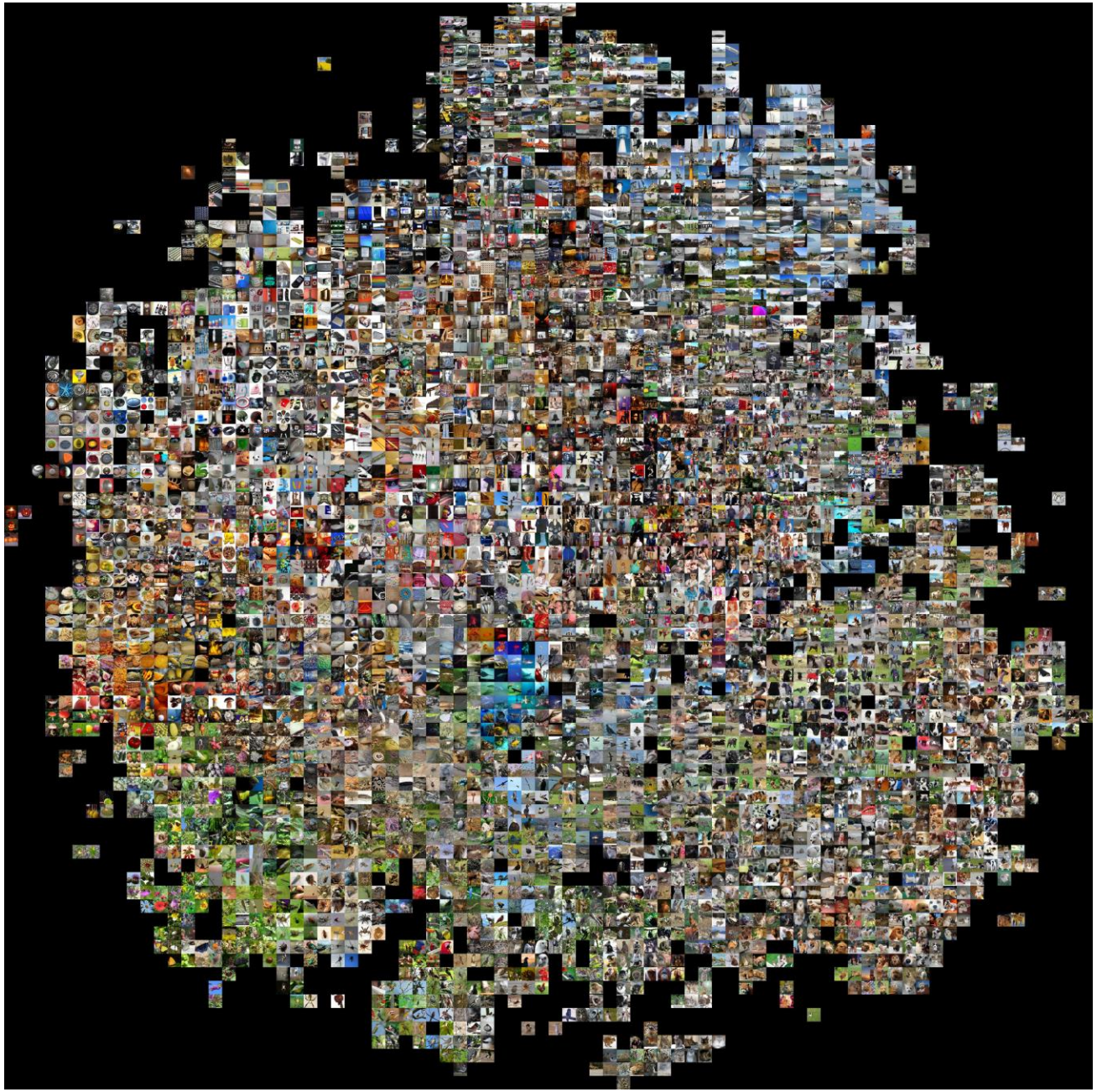
- Retrieve from real images



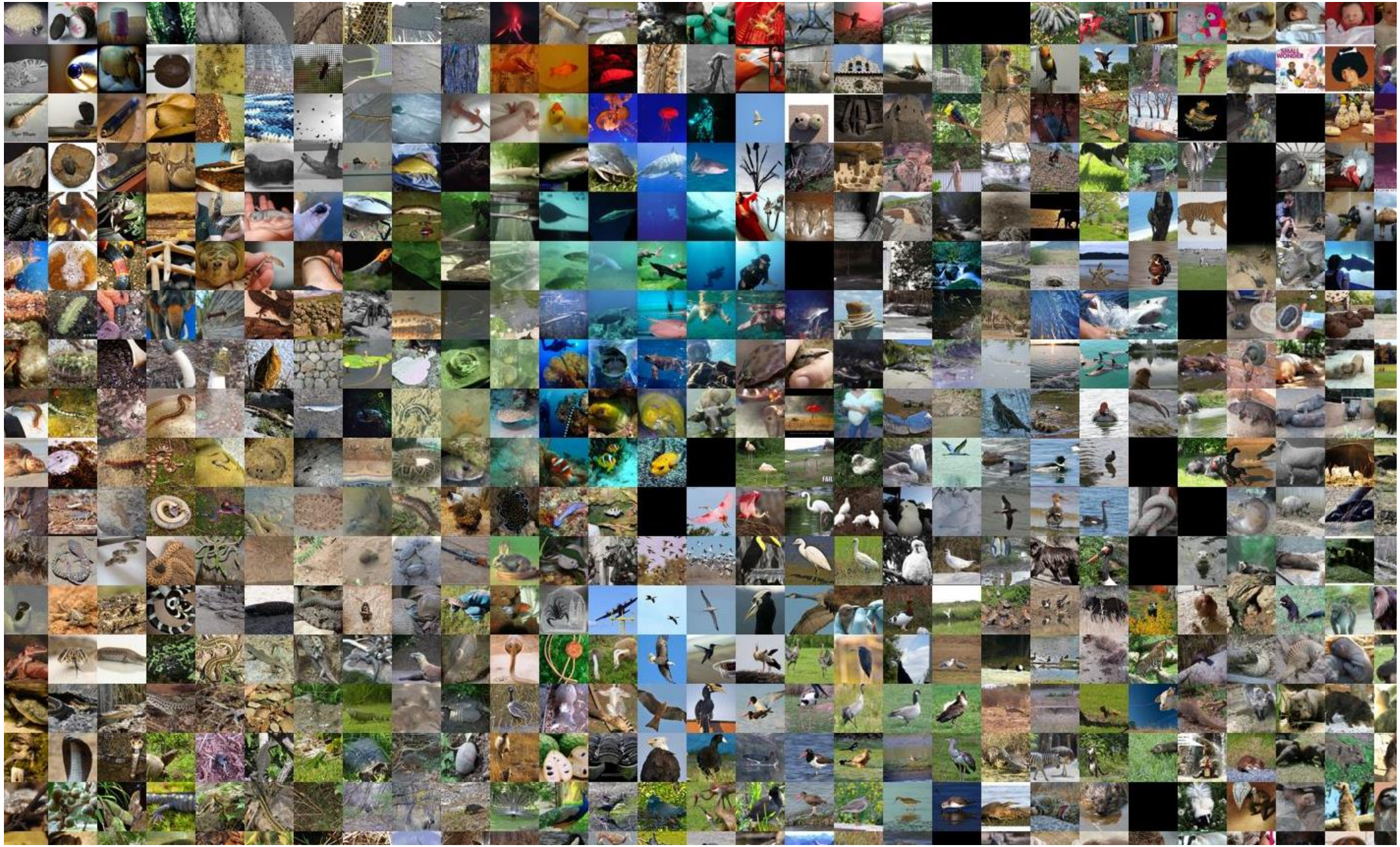
- Visualize layer activations
  - Deconvolution
- Feature visualization by optimization

- Attribution
- Dimensionality reduction

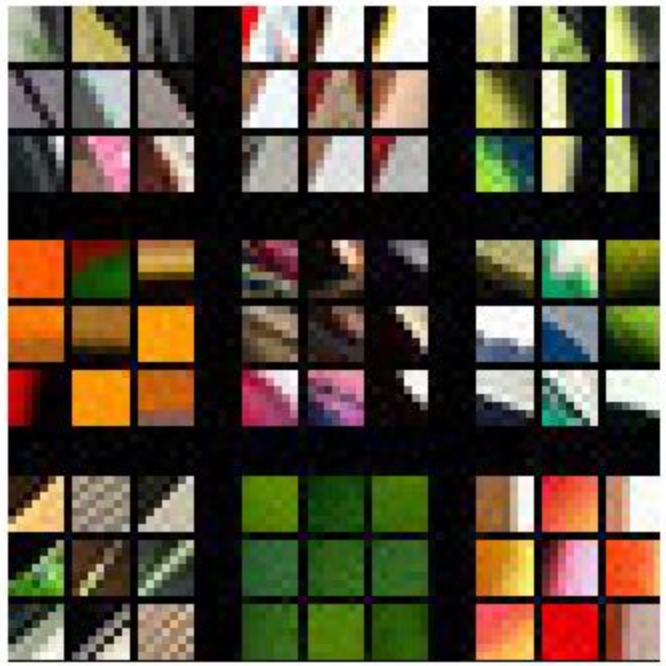
Embedding the codes with t-SNE



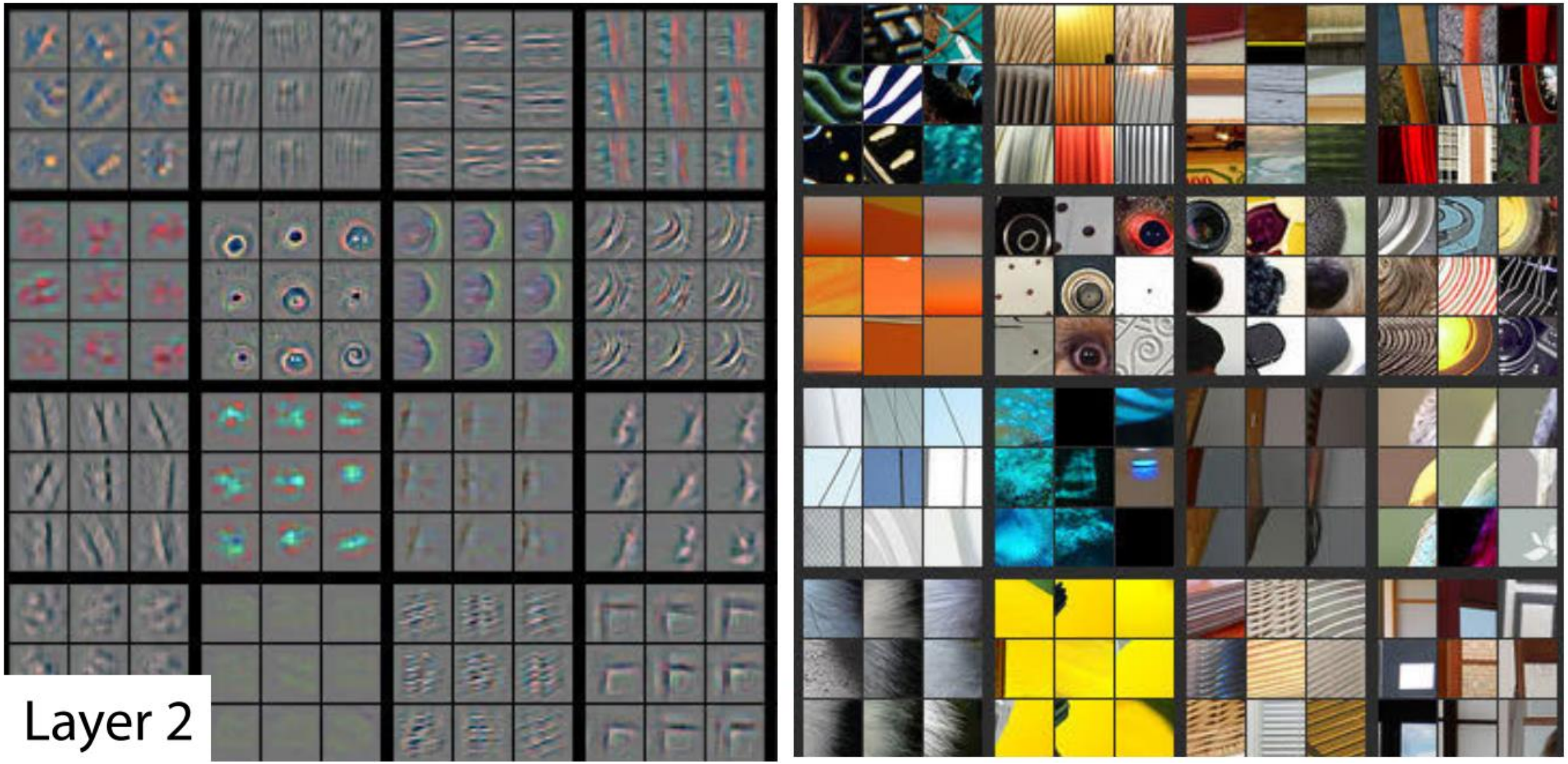
source



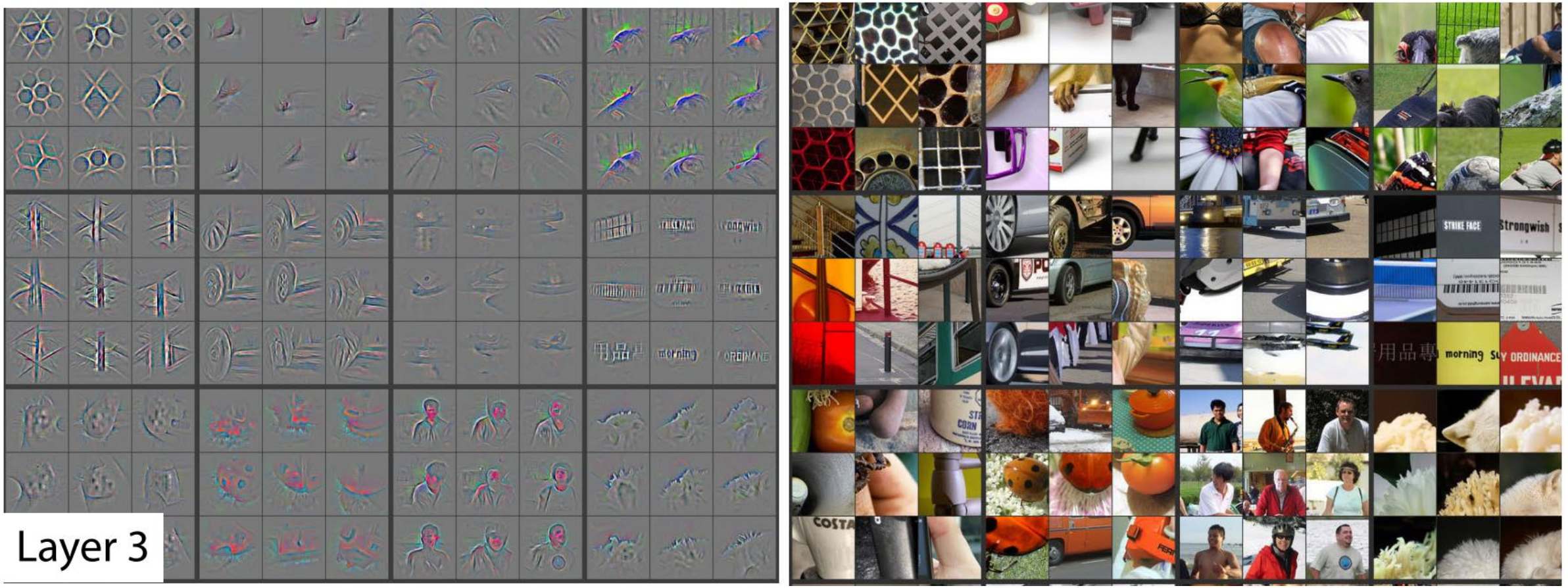
Layer 1



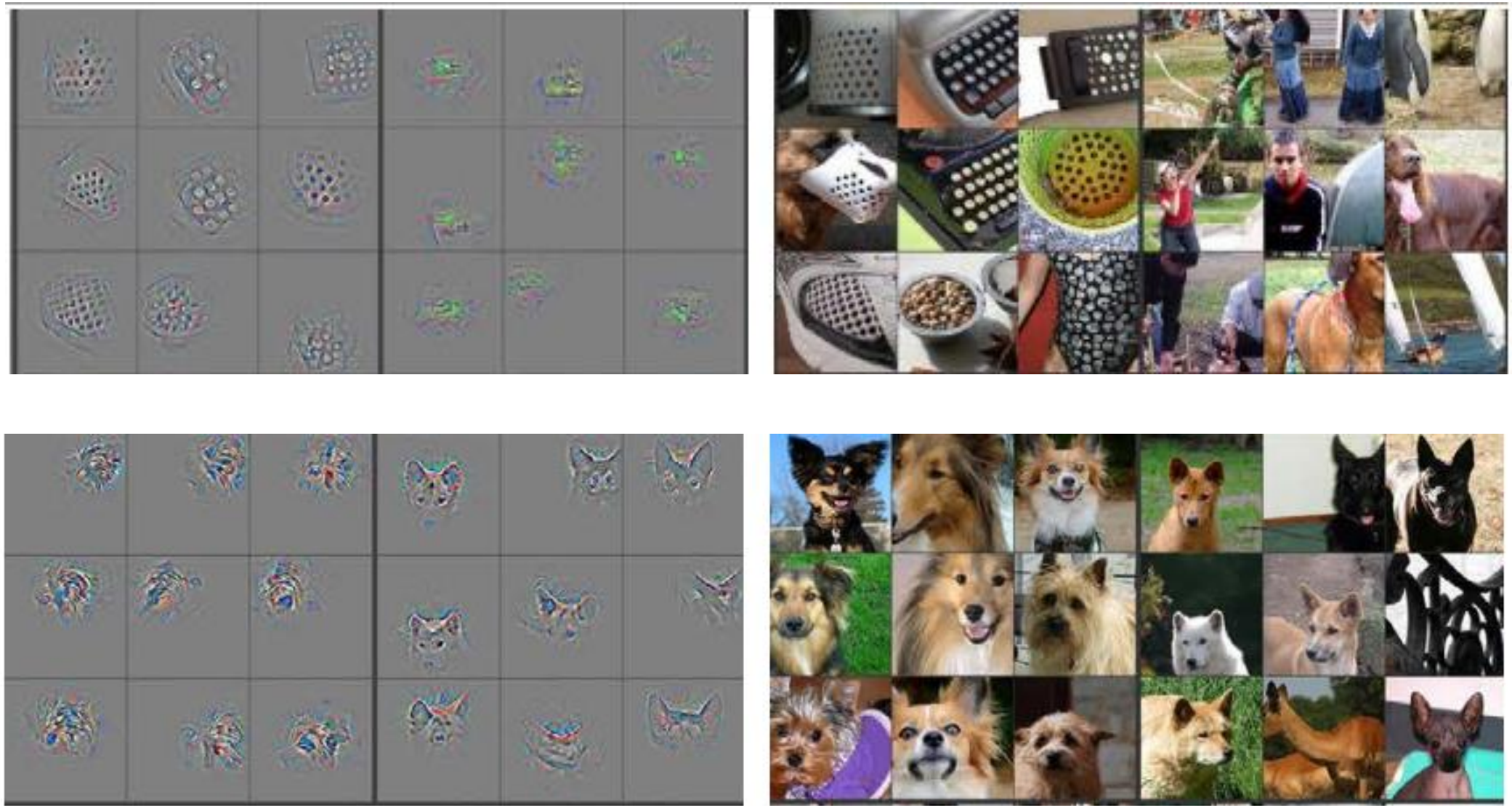
Zeiler and Fergus (2013): Visualizing and Understanding Convolutional Networks



Zeiler and Fergus (2013): Visualizing and Understanding Convolutional Networks



Zeiler and Fergus (2013): Visualizing and Understanding Convolutional Networks



Zeiler and Fergus (2013): Visualizing and Understanding Convolutional Networks