Yandex

NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS

SCHOOL OF DATA ANALYSIS

Skoltech
Skolkovo Institute of Science and Technology

Explaining Neural Networks

Visualization of neural networks. Information theory of deep learning. Deep Taylor decomposition.

Fourth Machine Learning in High Energy Physics Summer School,
MLHEP 2018, August 6-12

Alexey Artemov[1,2]

[1]Skoltech [2]National Research University Higher School of Economics

How can we understand deep neural networks better?

2

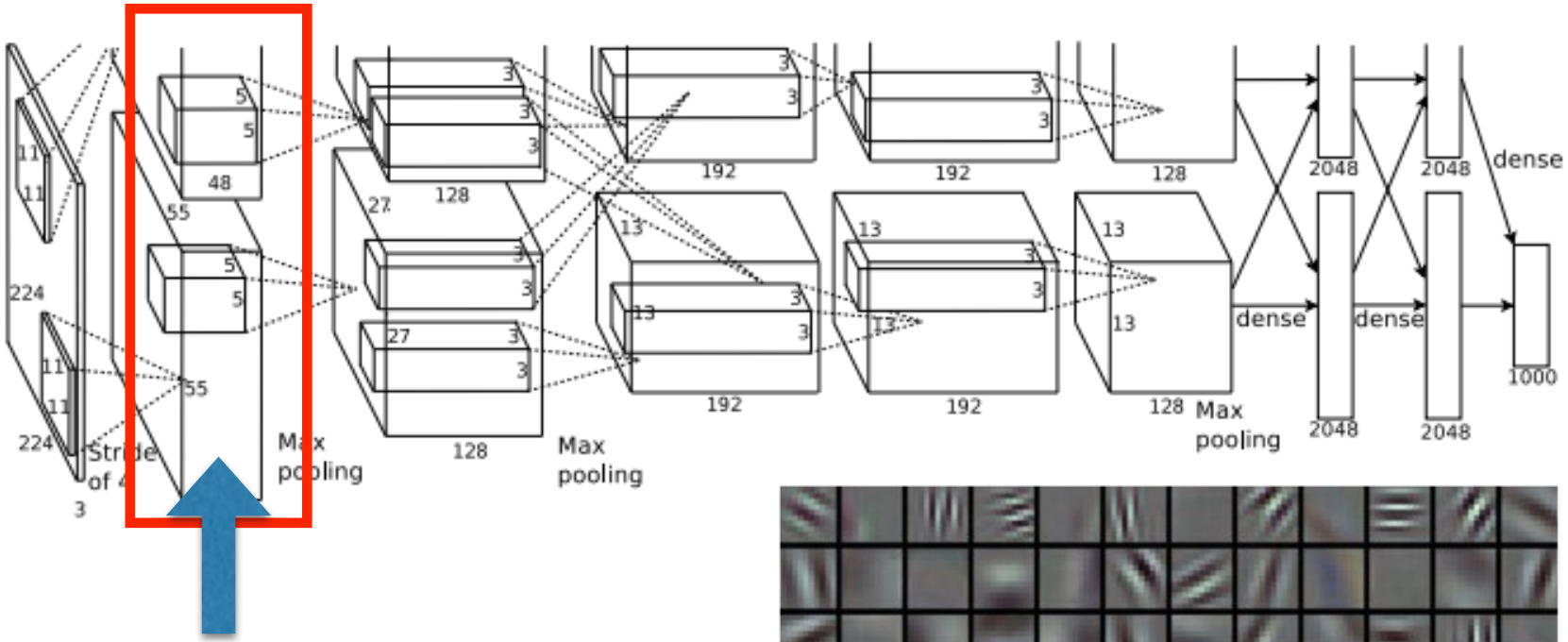
How can we understand DNNs better?

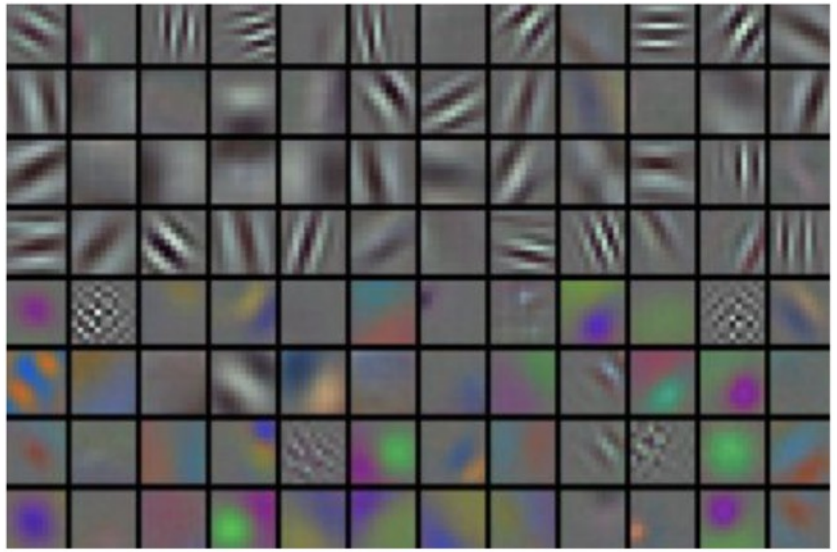
- By visualizing their internal structure!
  - Visualize the weights
  - Visualize patches that maximally activate neurons
  - Visualize the representation space (e.g. with t-SNE)
- Occlusion experiments
- Deconv approaches
- Optimization over input image
- Exploring the information plane
- Decomposing input according to its influence on the output

3

The visualization of neural networks

Visualize filter kernels





Layer 1 filter kernels make direct sense (i.e. are interpretable)

5

Visualize filter kernels (ConvNetJS)

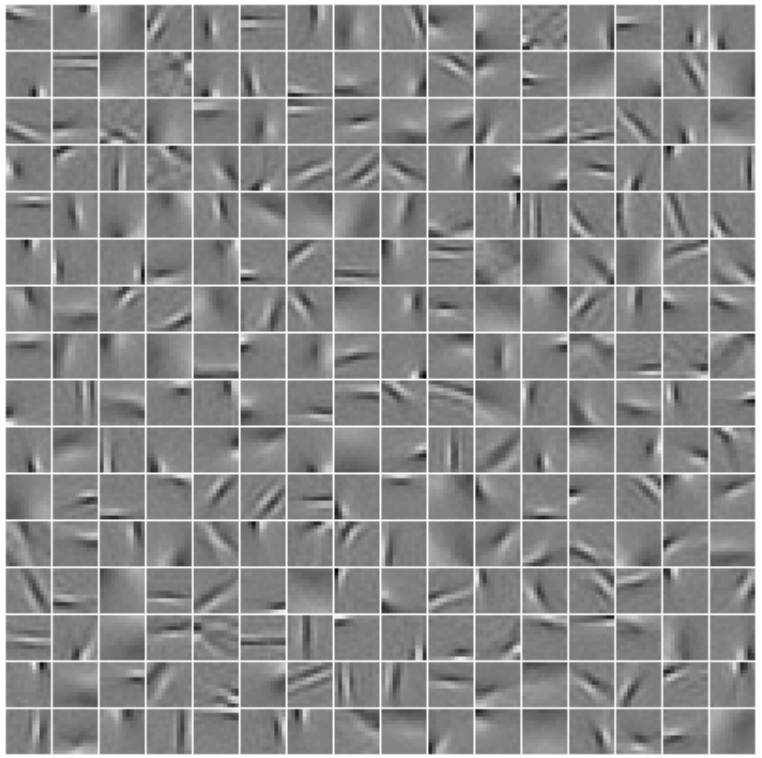
Weights:

Layer 1 filter kernels make sense,
Layer 2 do not (at least, directly)

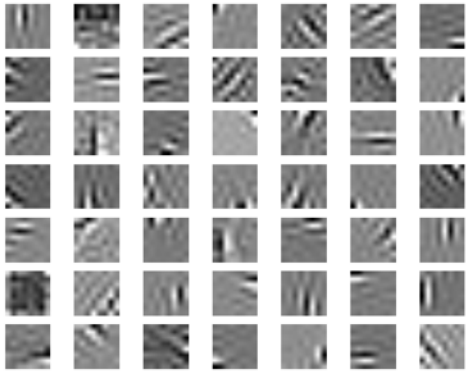
Weights:

Layer 1: 16 5x5x3 kernels
Layer 2: 20 5x5x16 kernels

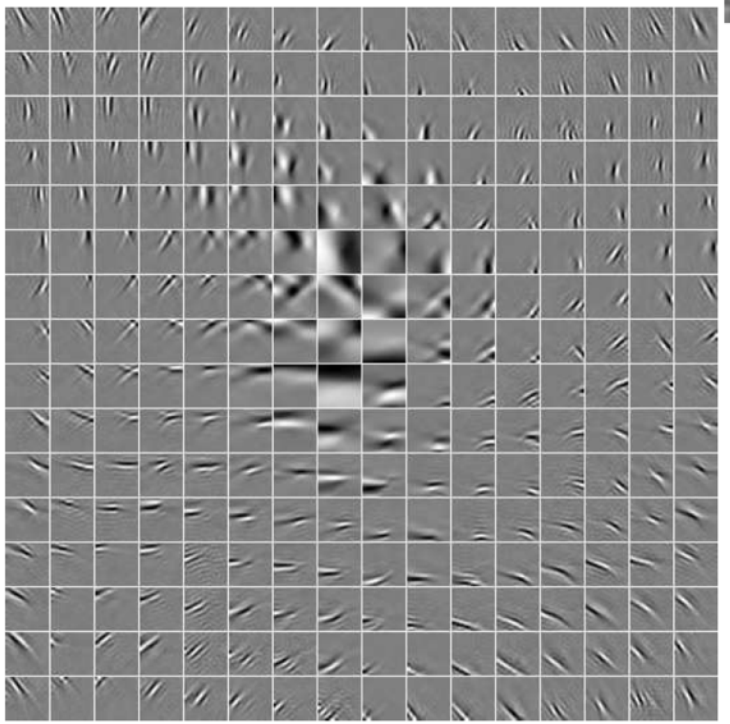
Gabor-like filters all over the place



Predictive Sparse Decomposition



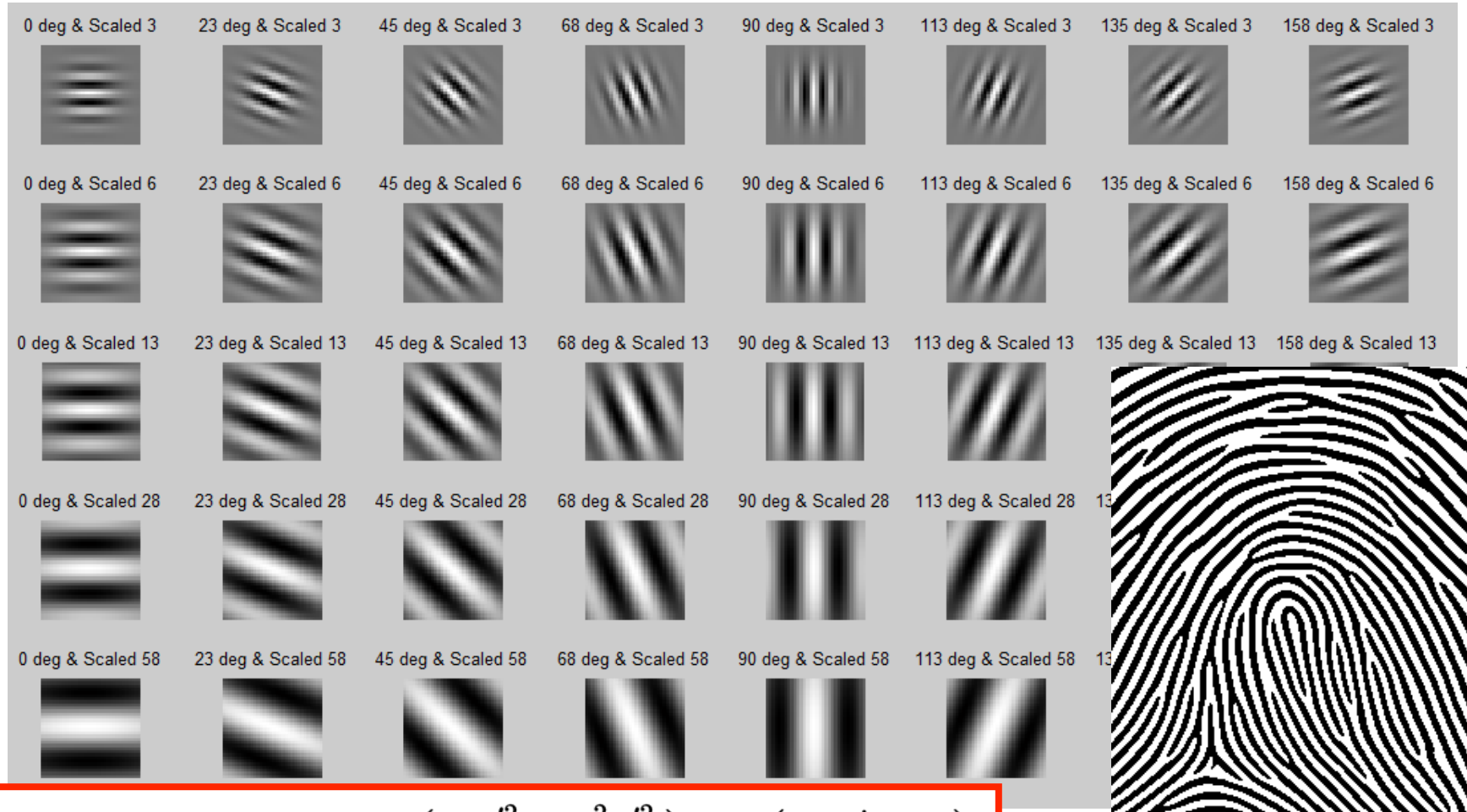
Gabor filters



Natural local features of an image (by Aapo Hyvärinen)

7

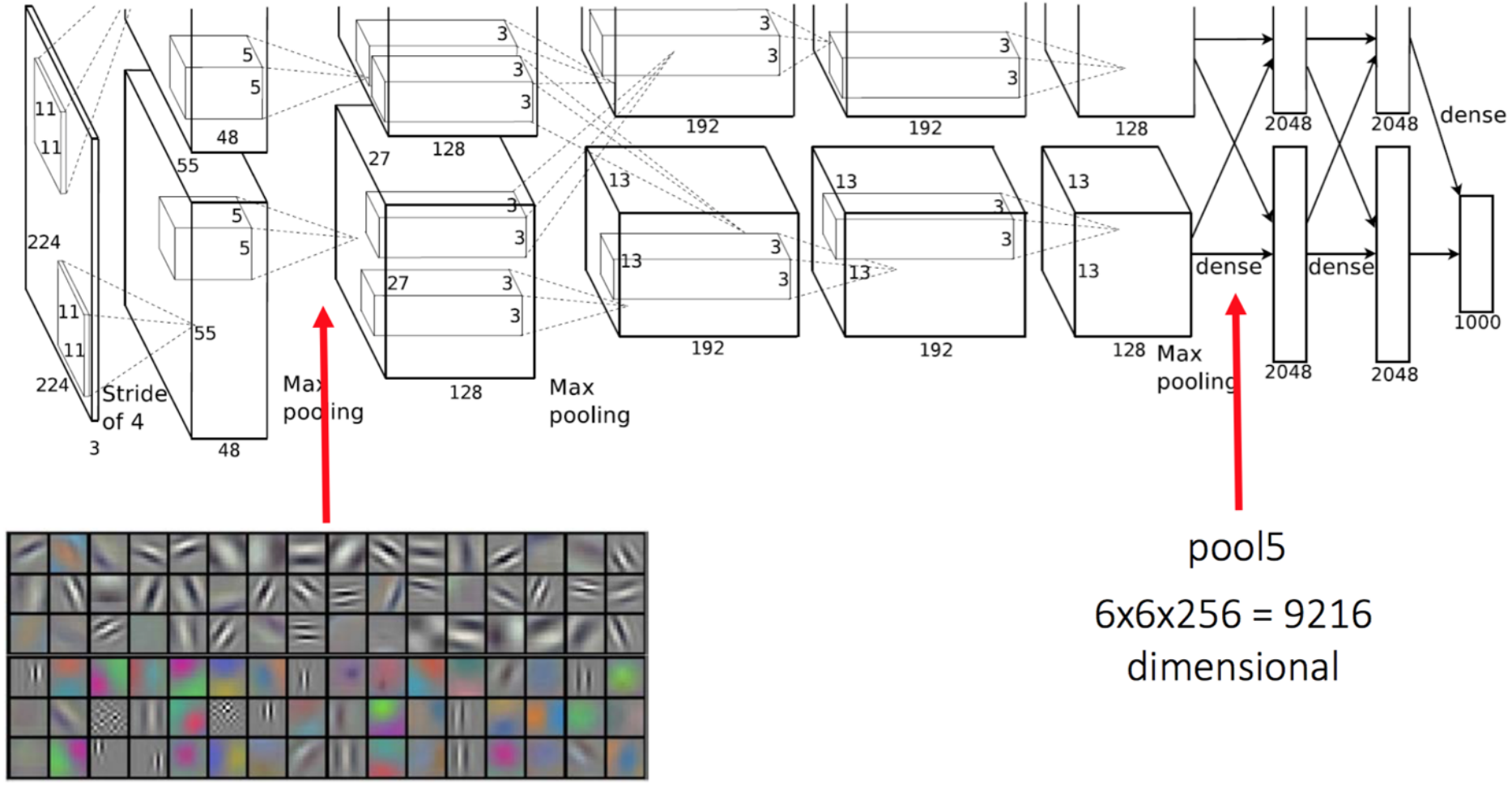
Gabor-like filters all over the place



$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cos\left(2\pi \frac{x'}{\lambda} + \psi\right)$$

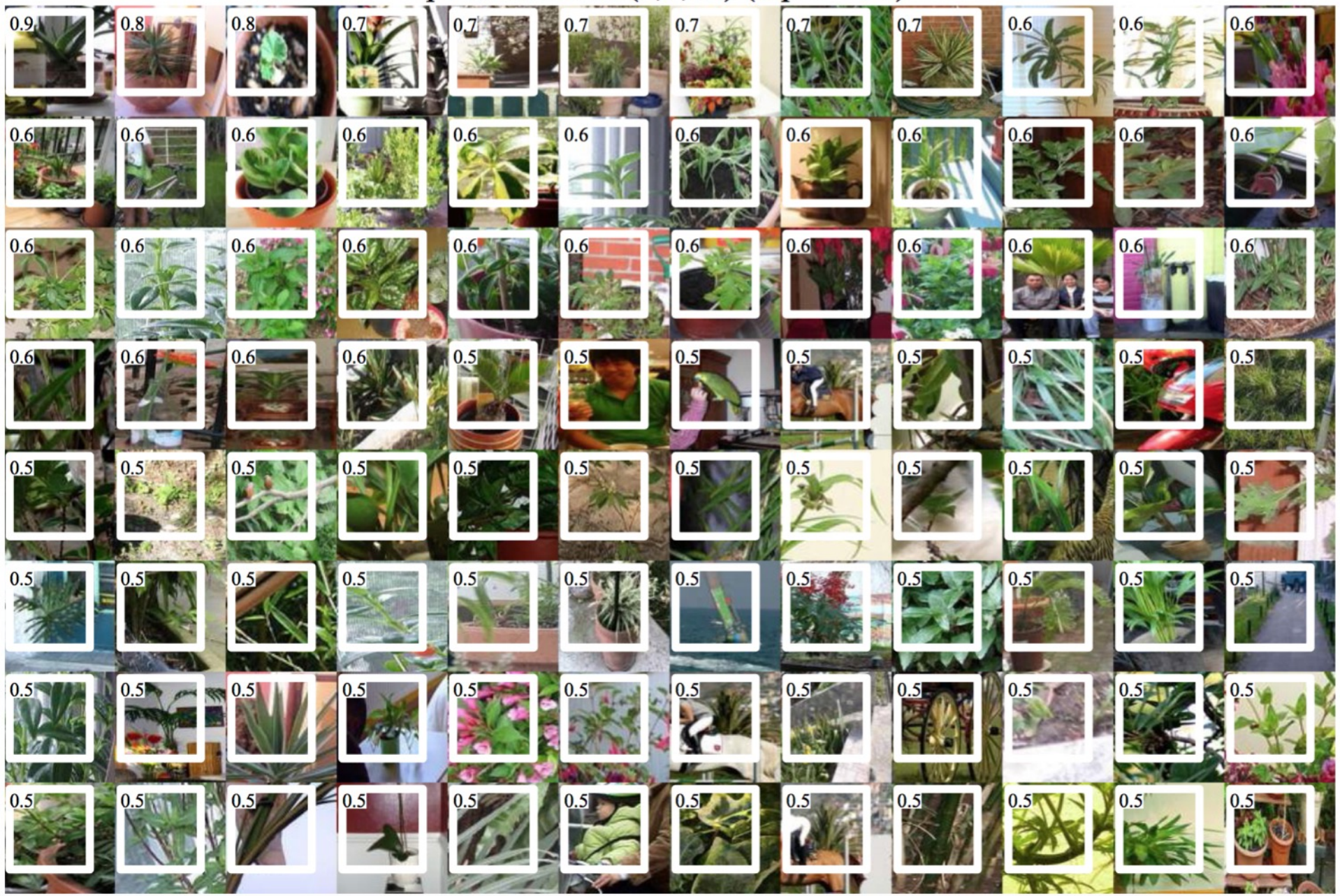
8

Visualize image patches that strongly activate neurons



9

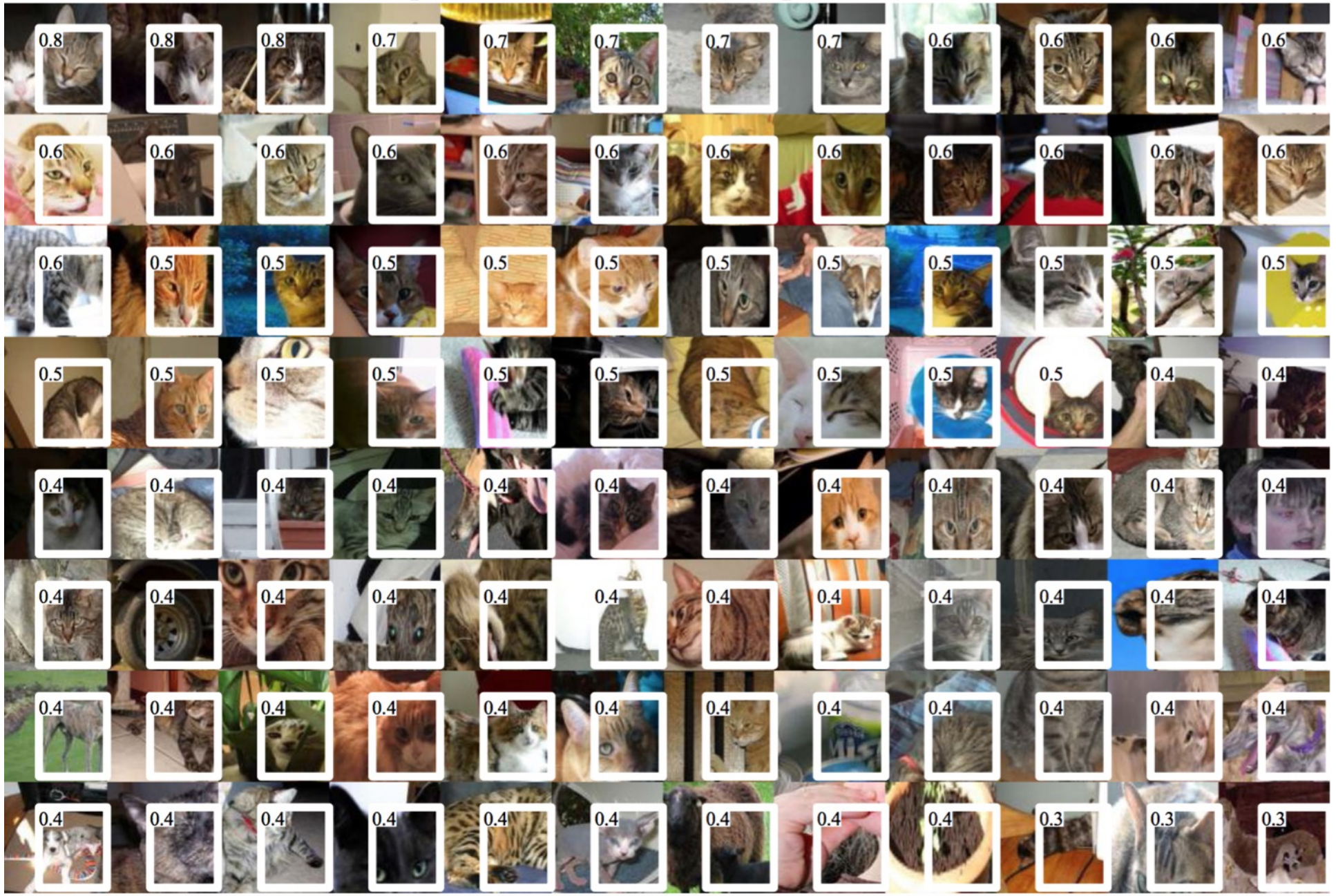
pool5 feature: (3,3,42) (top 1 – 96)



pool5 feature: (3,4,80) (top 1 − 96)

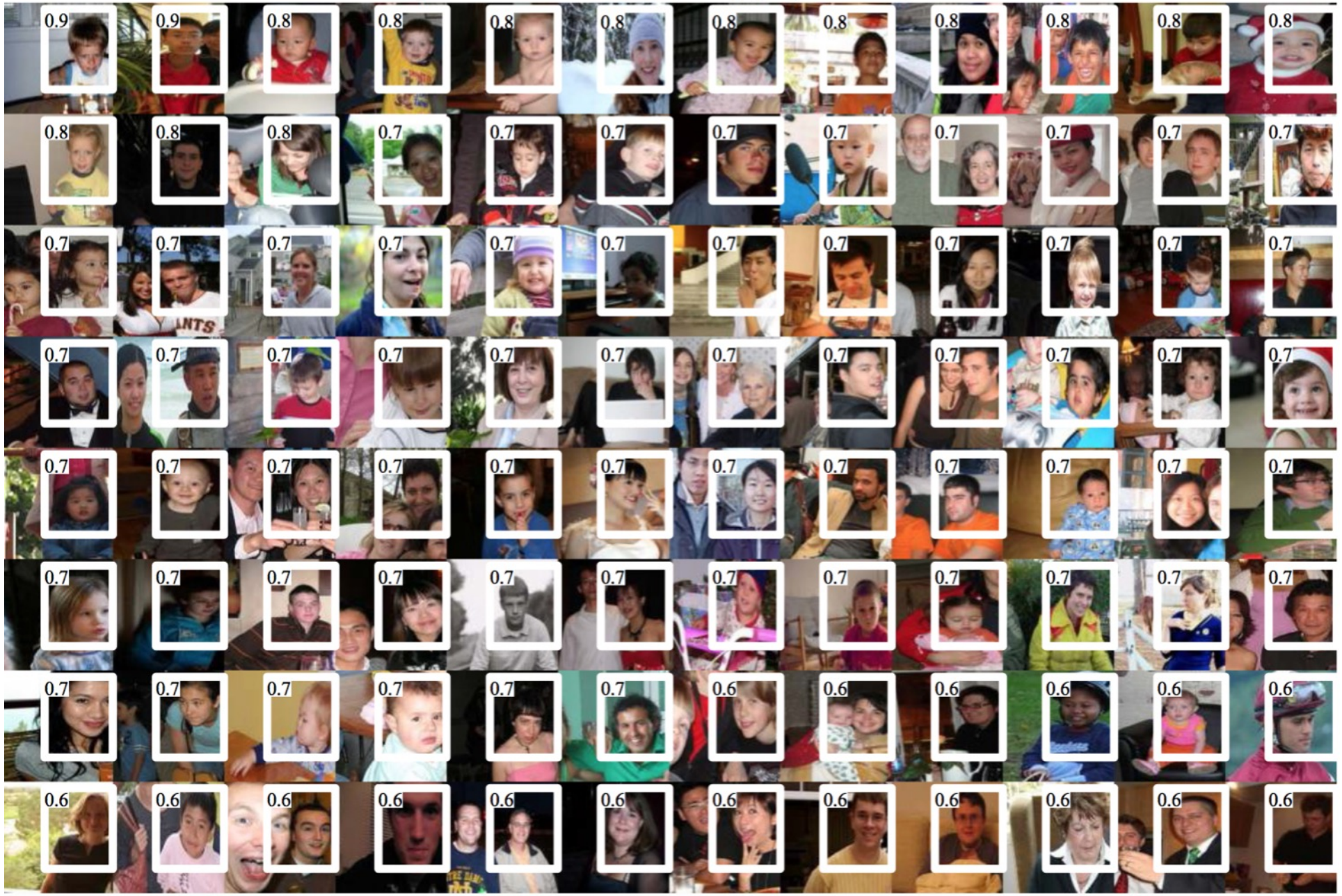
1

pool5 feature: (4,5,110) (top 1 − 96)



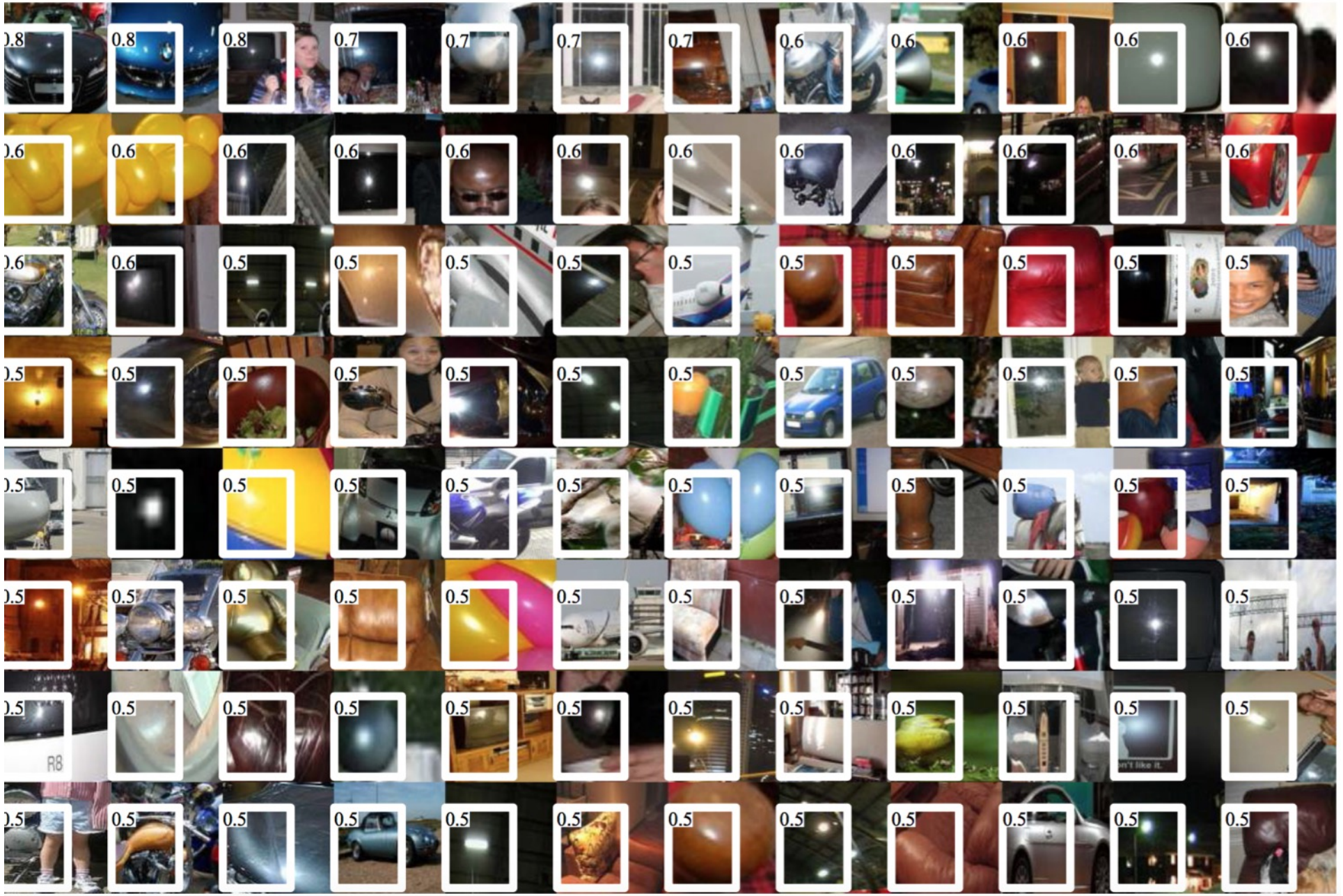
1

pool5 feature: (3,5,129) (top 1 − 96)



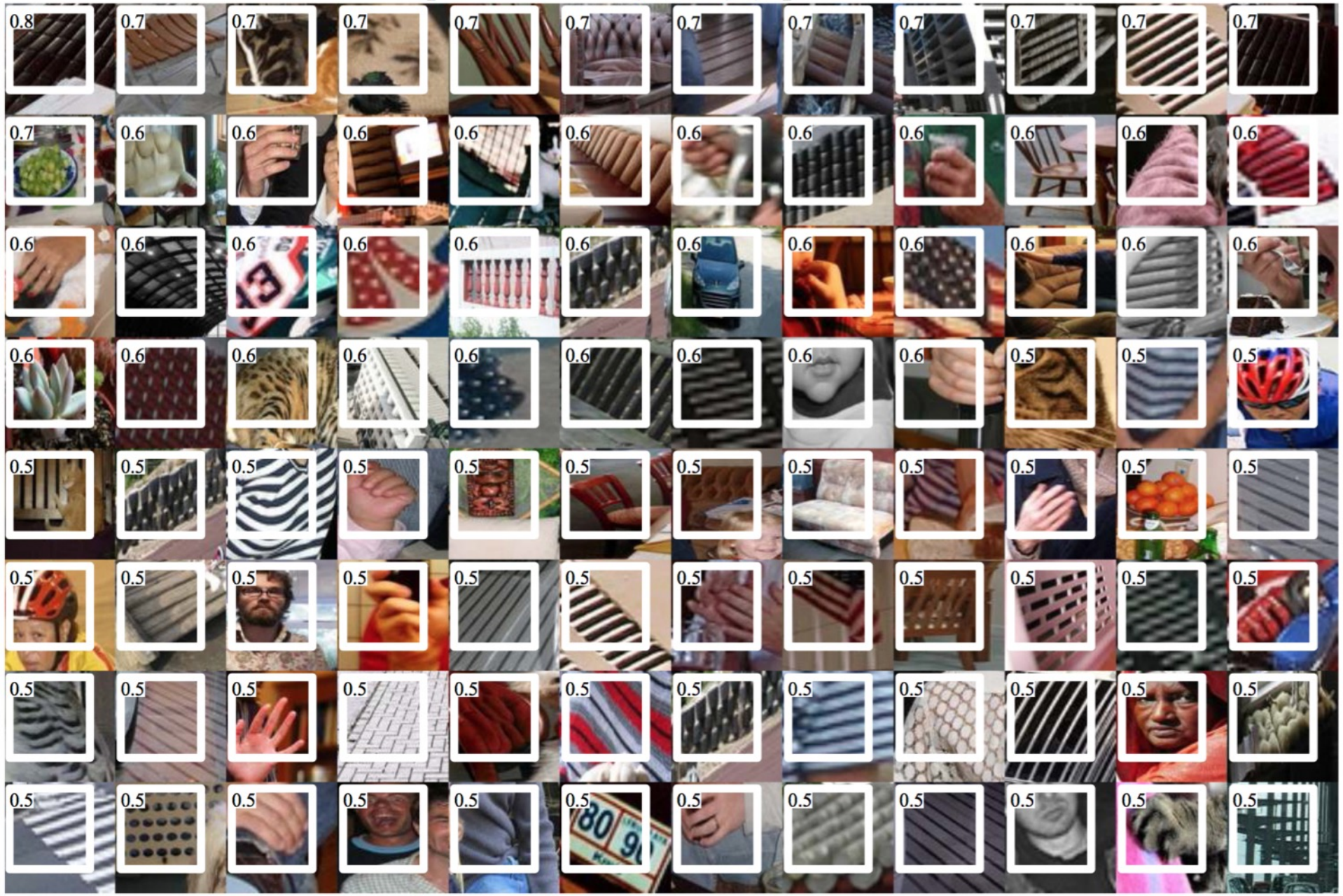
1

pool5 feature: (4,2,26) (top 1 − 96)



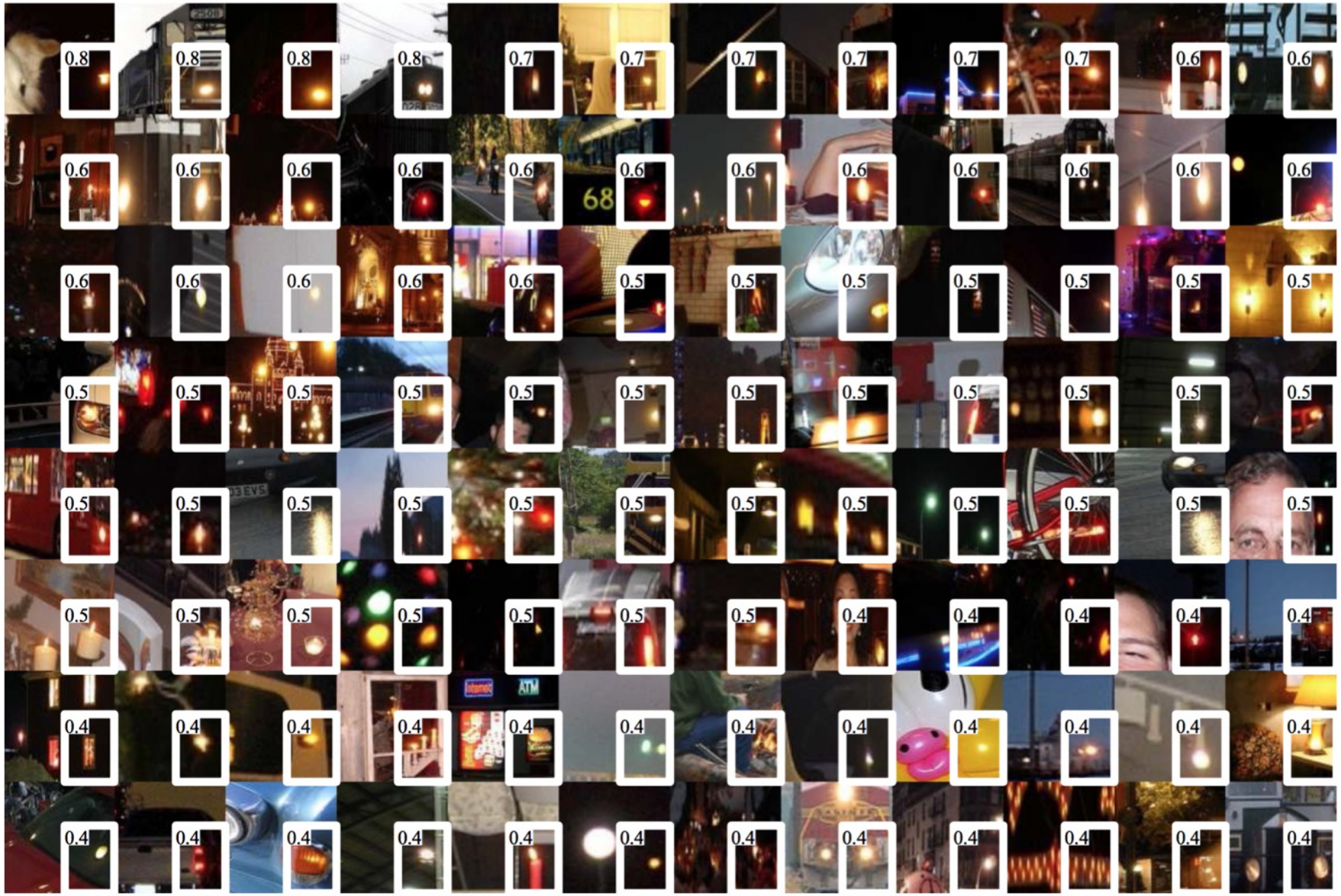
1

pool5 feature: (3,3,39) (top 1 − 96)



1

pool5 feature: (5,6,53) (top 1 − 96)

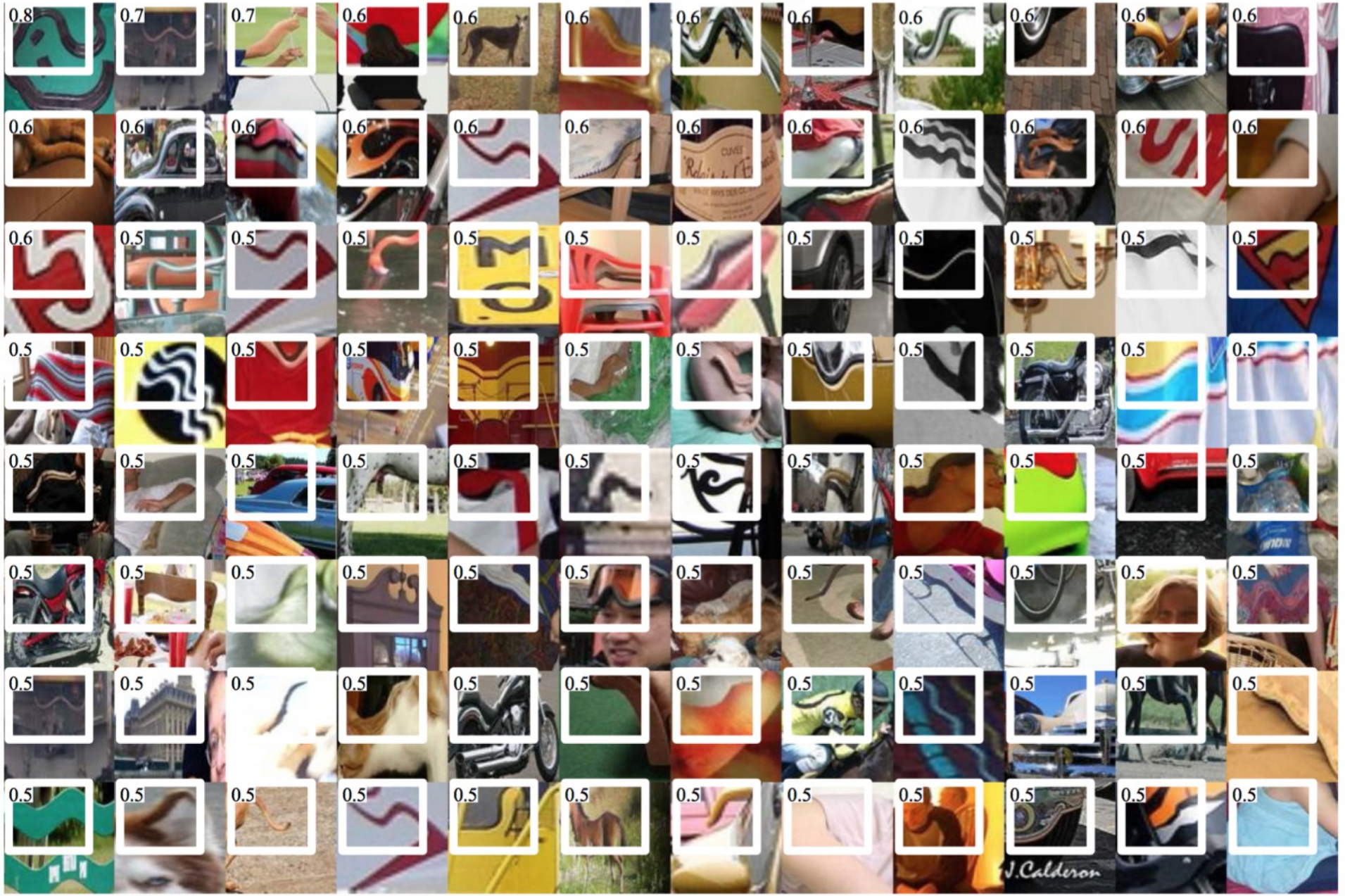


1

pool5 feature: (1,4,138) (top 1 – 96)

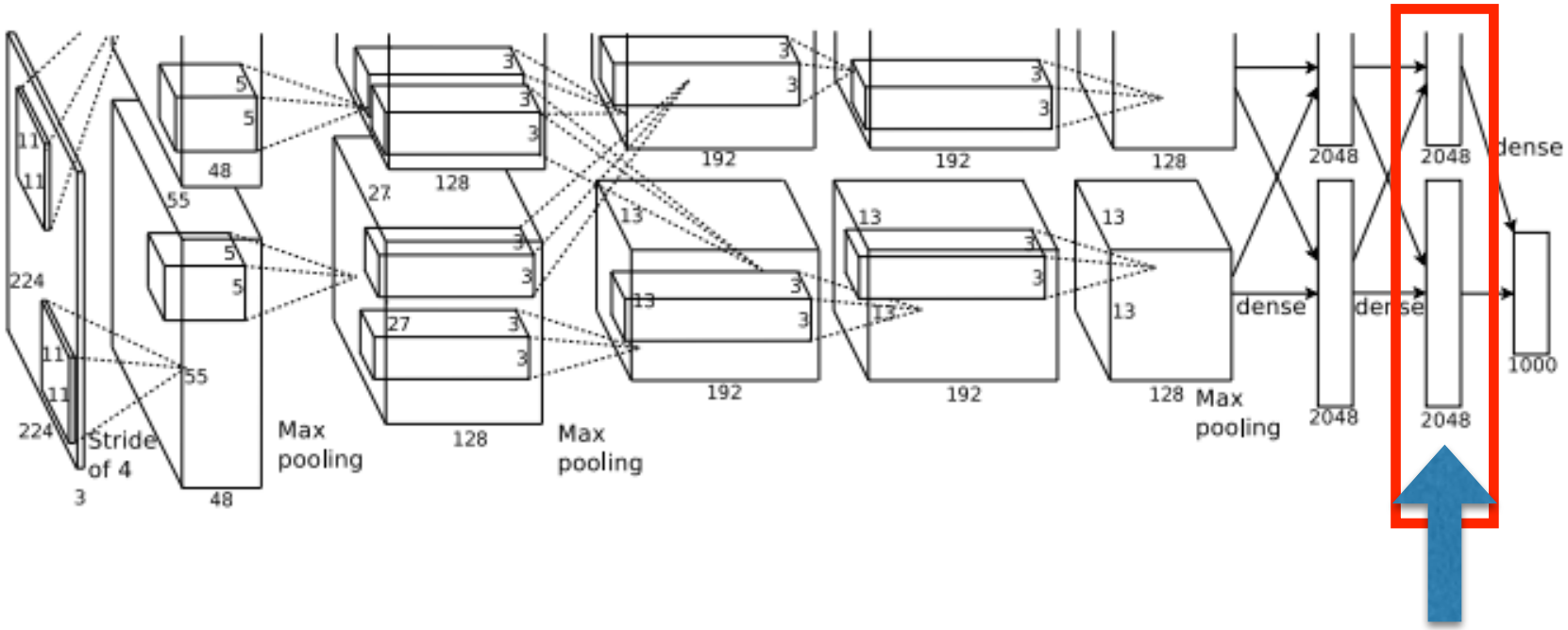
1

pool5 feature: (2,3,210) (top 1 − 96)



1

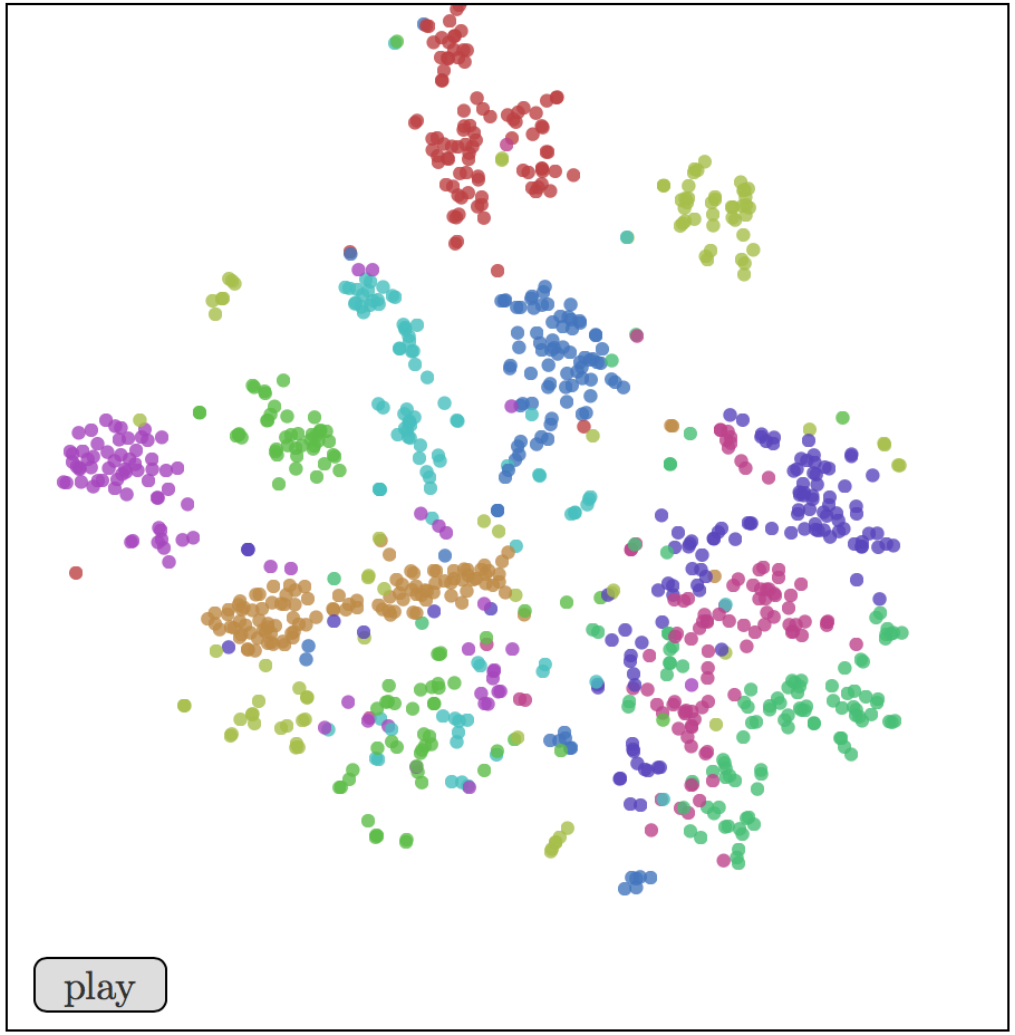
Visualize projected descriptors



Layer 7 outputs (immediately before the classifier) encode image summary

Visualize projected descriptors

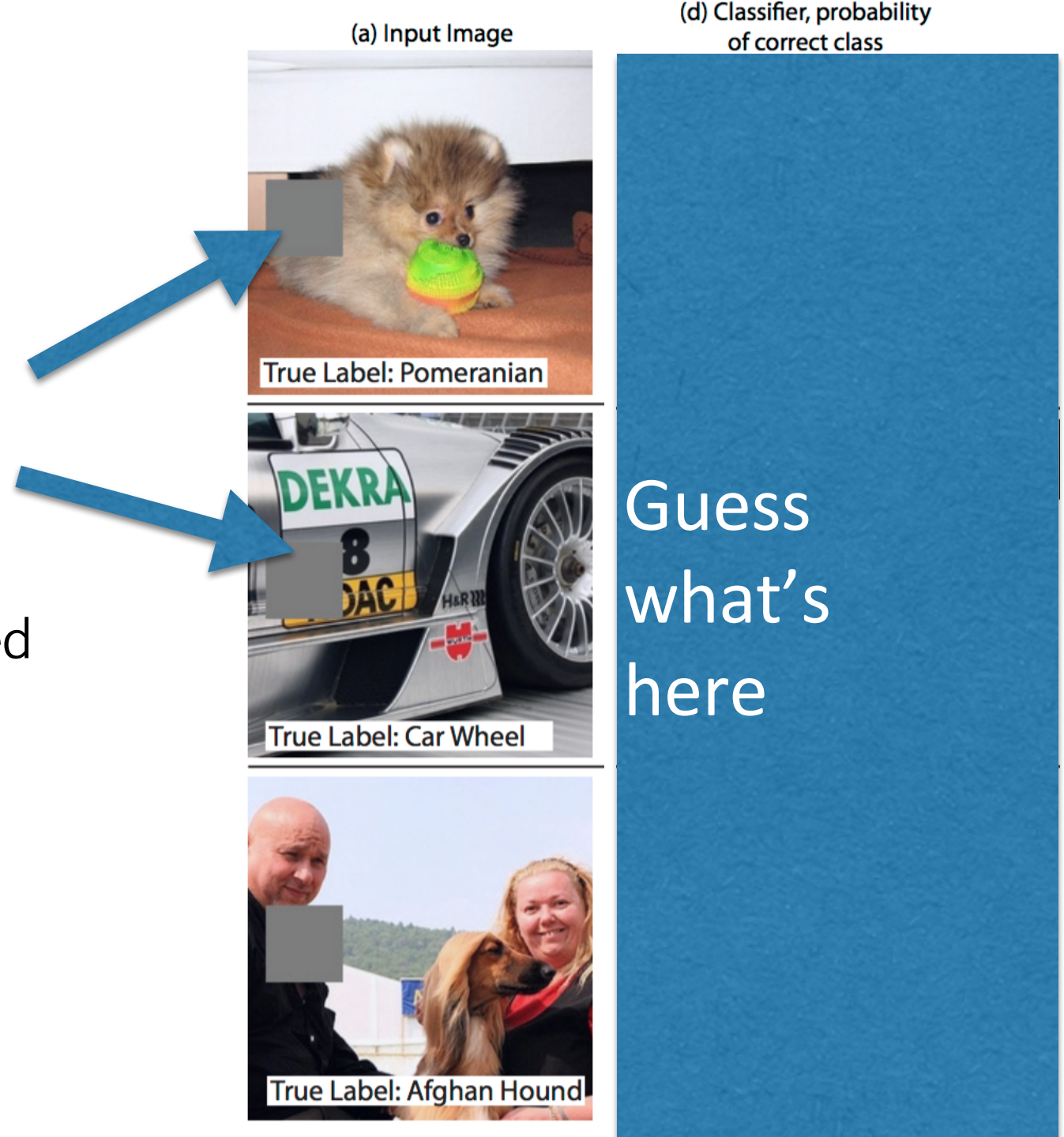
- Take the pre-trained CNN
- Extract a large number (50K) high-dimensional features from its last layer
- Use some structure-preserving dimension reduction technique to compute a low-dimensional embedding (similar things end up in similar places)
- t-SNE works fine
- (Open the image in Preview)



Visualizing MNIST with t-SNE

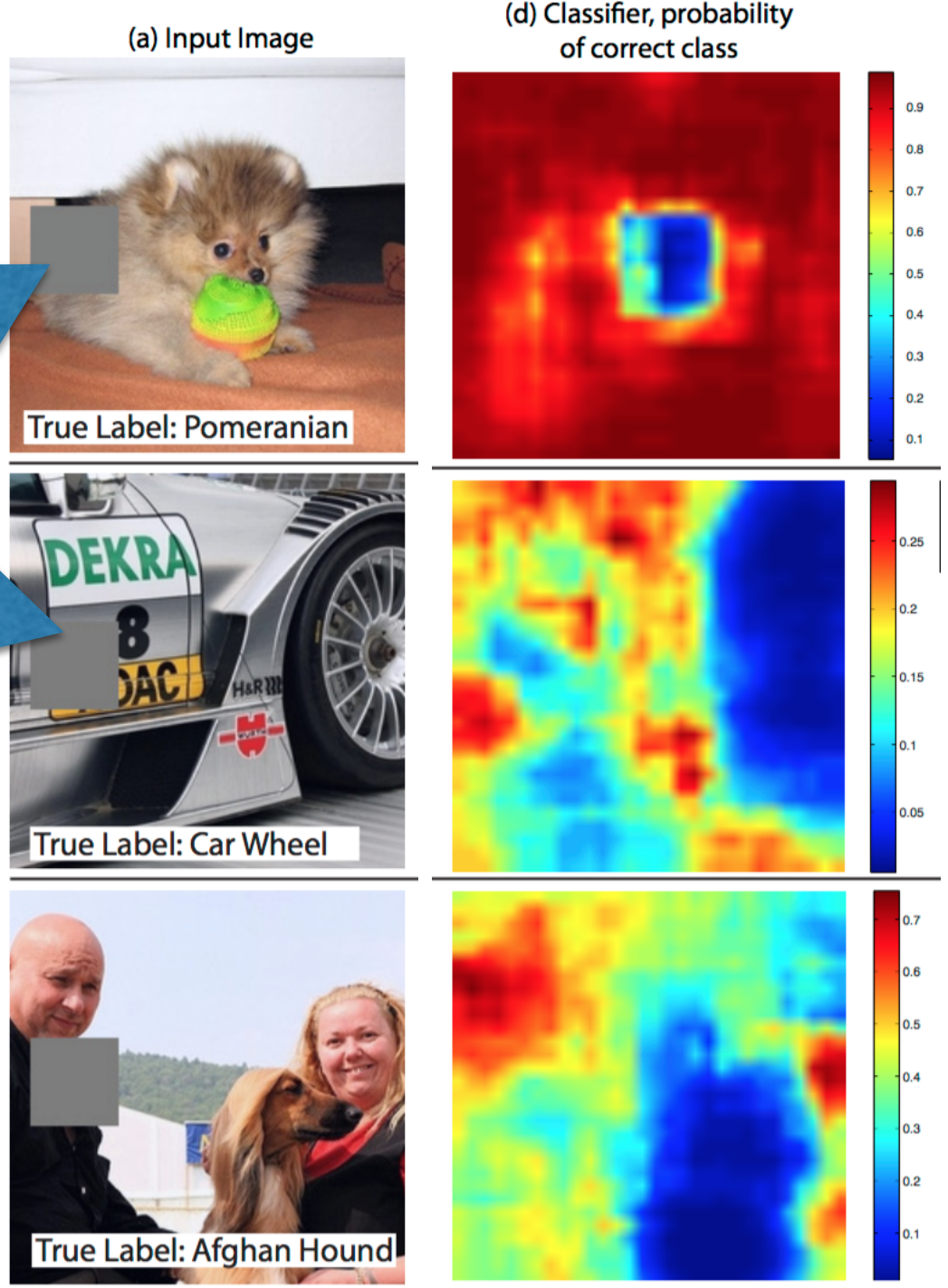
Occlusion experiments

- Slide a little square filled with zeros across the image, compute score for the true class



Occlusion experiments

- Slide a little square filled with zeros across the image, compute score for the true class

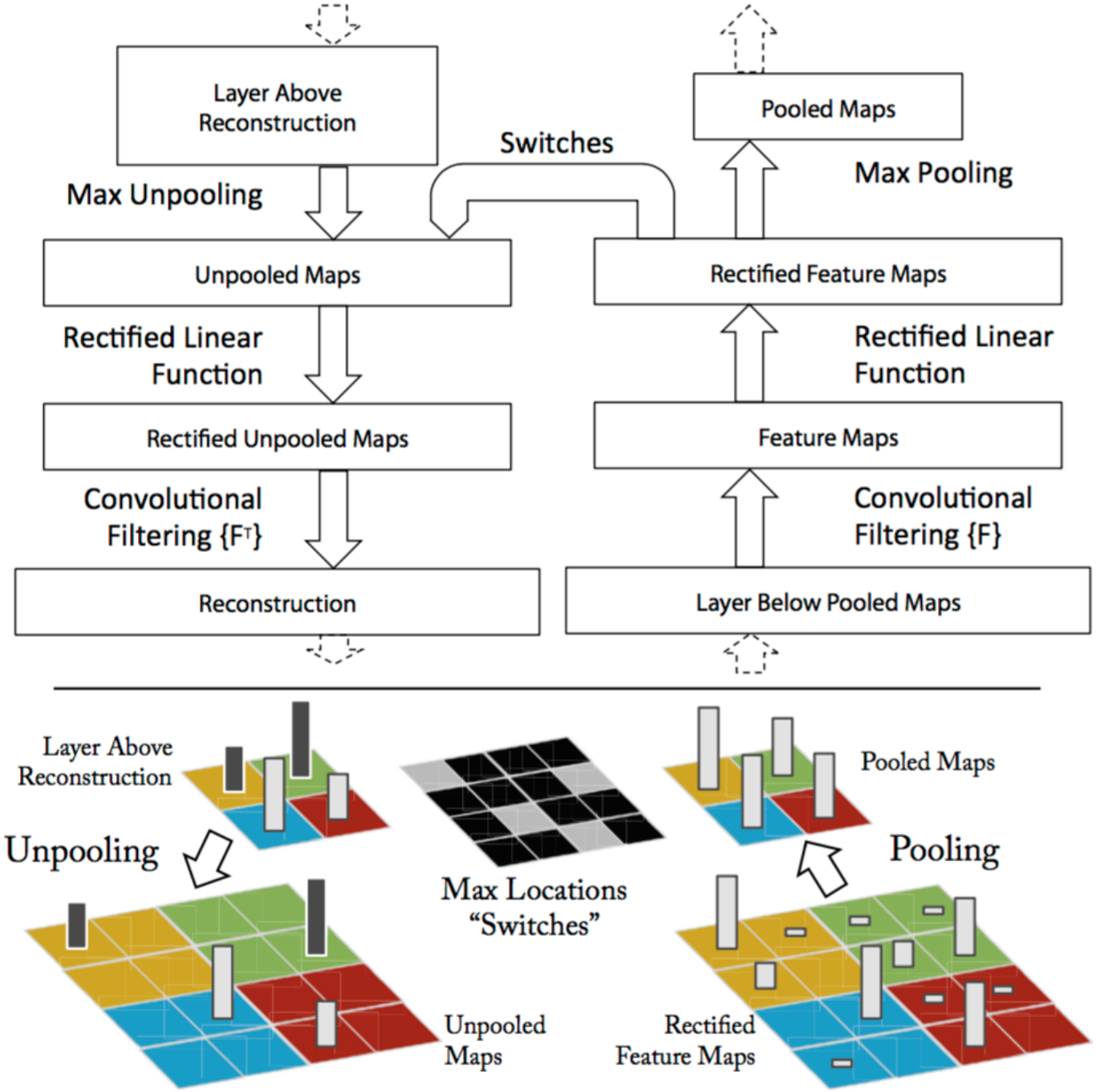


Deconv approach

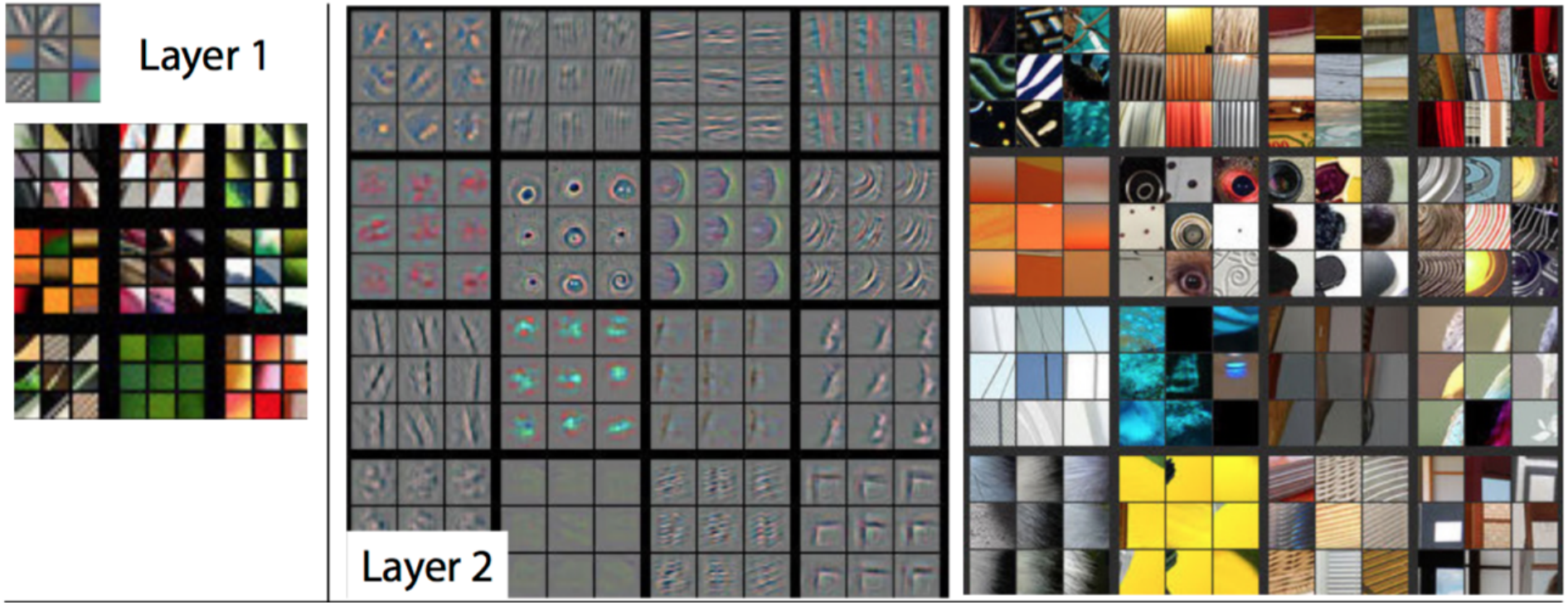
- Attach a deconvnet to each layer
- Pass the image though the network
- Set all but one activations to zero, pass the feature maps to the attached deconvnet
- (i) unpool, (ii) rectify, (iii) filter to reconstruct activity
- Repeat until pixel space is reached

Deconv approach

- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]

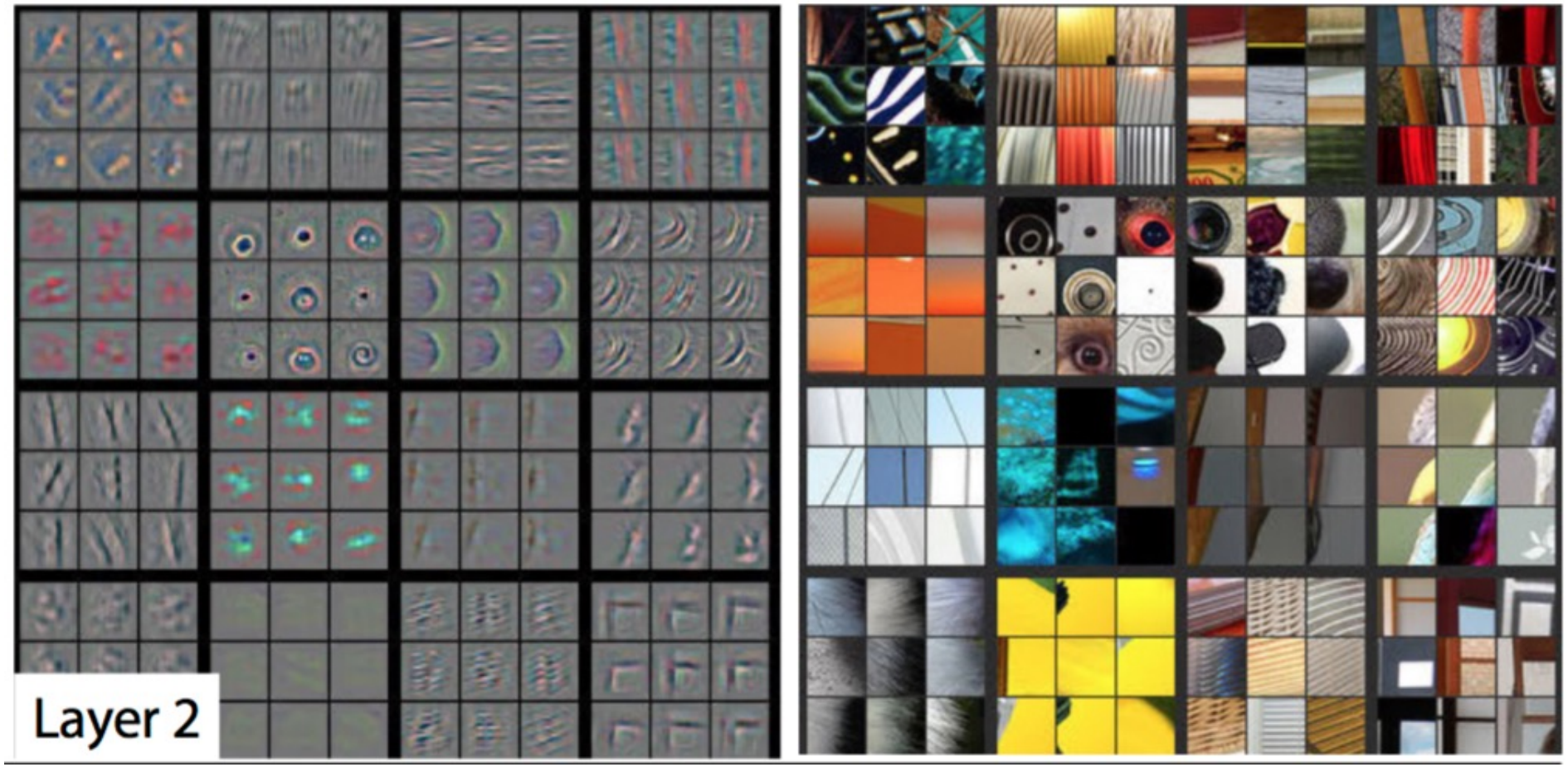


Deconv results



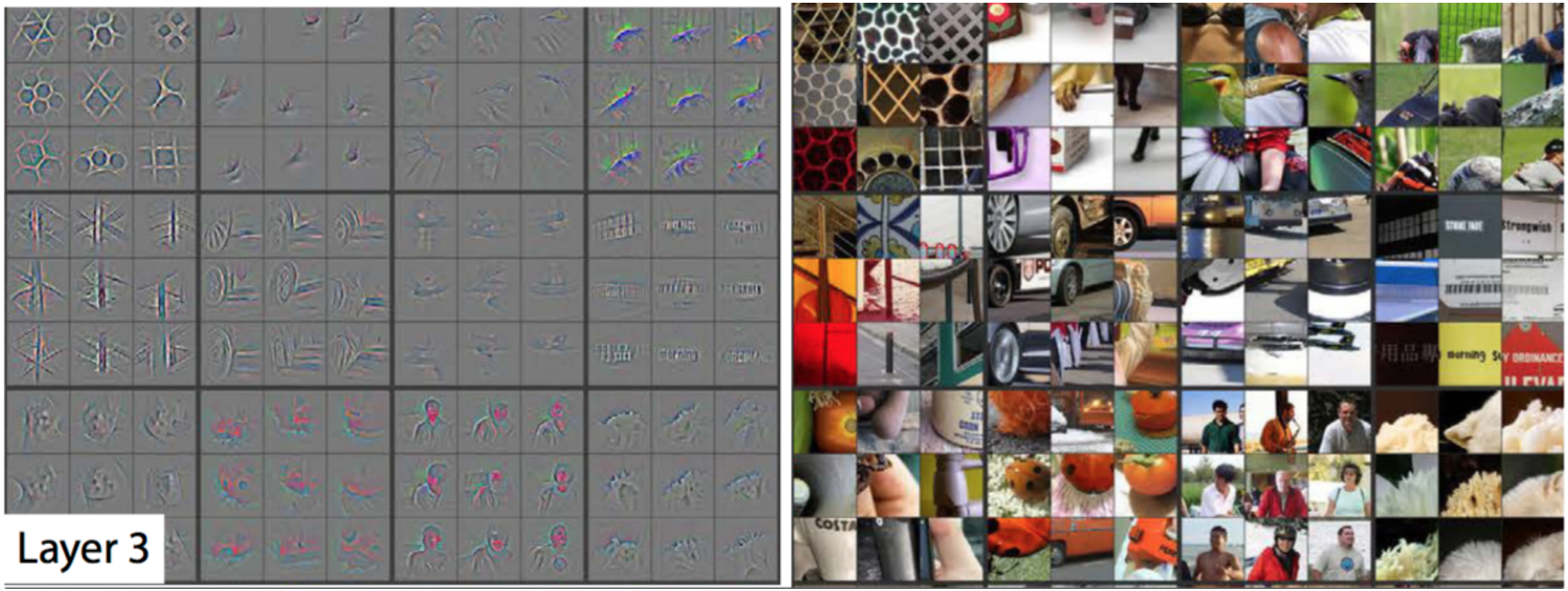
- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]

Deconv results



- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]

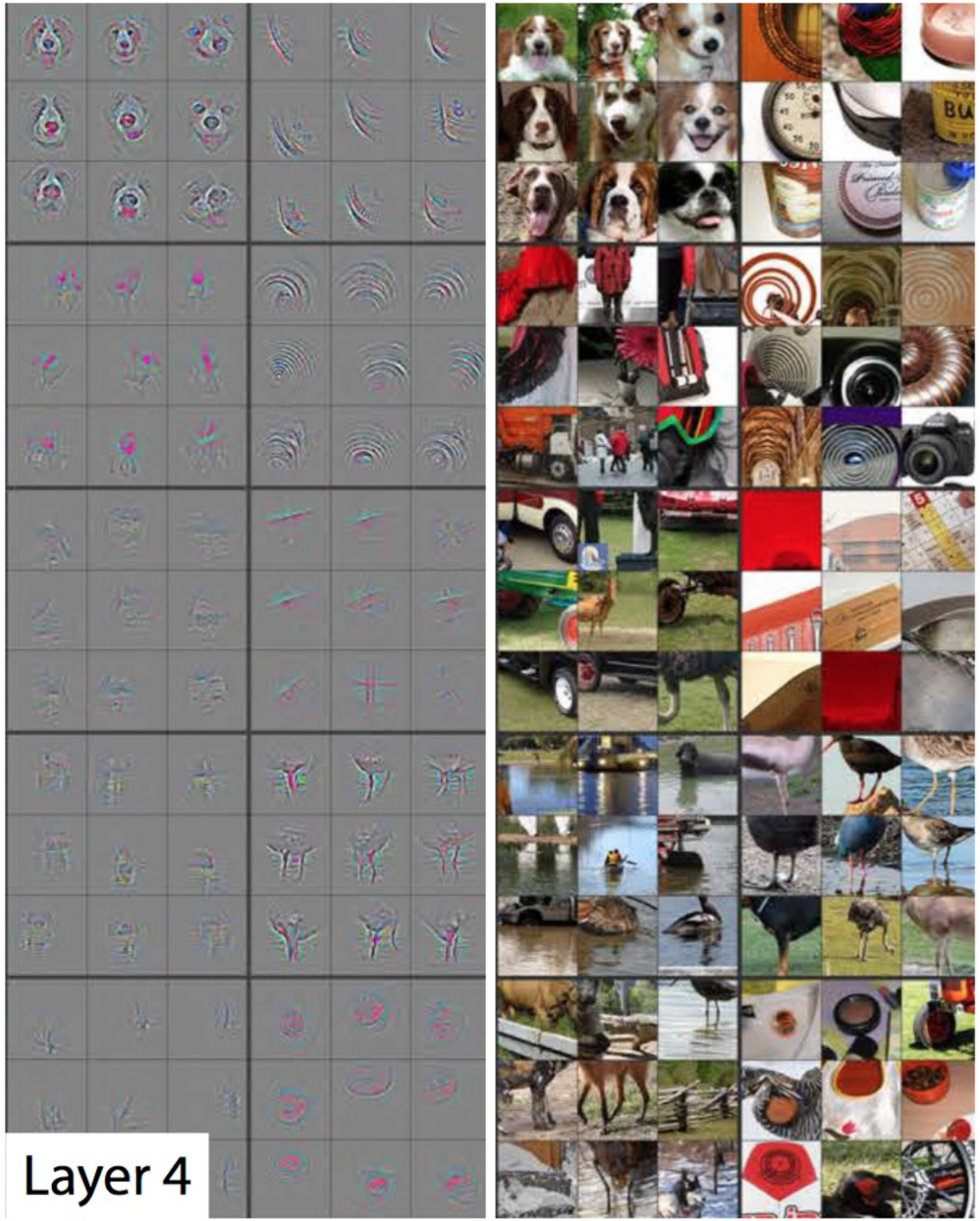
Deconv results



- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]

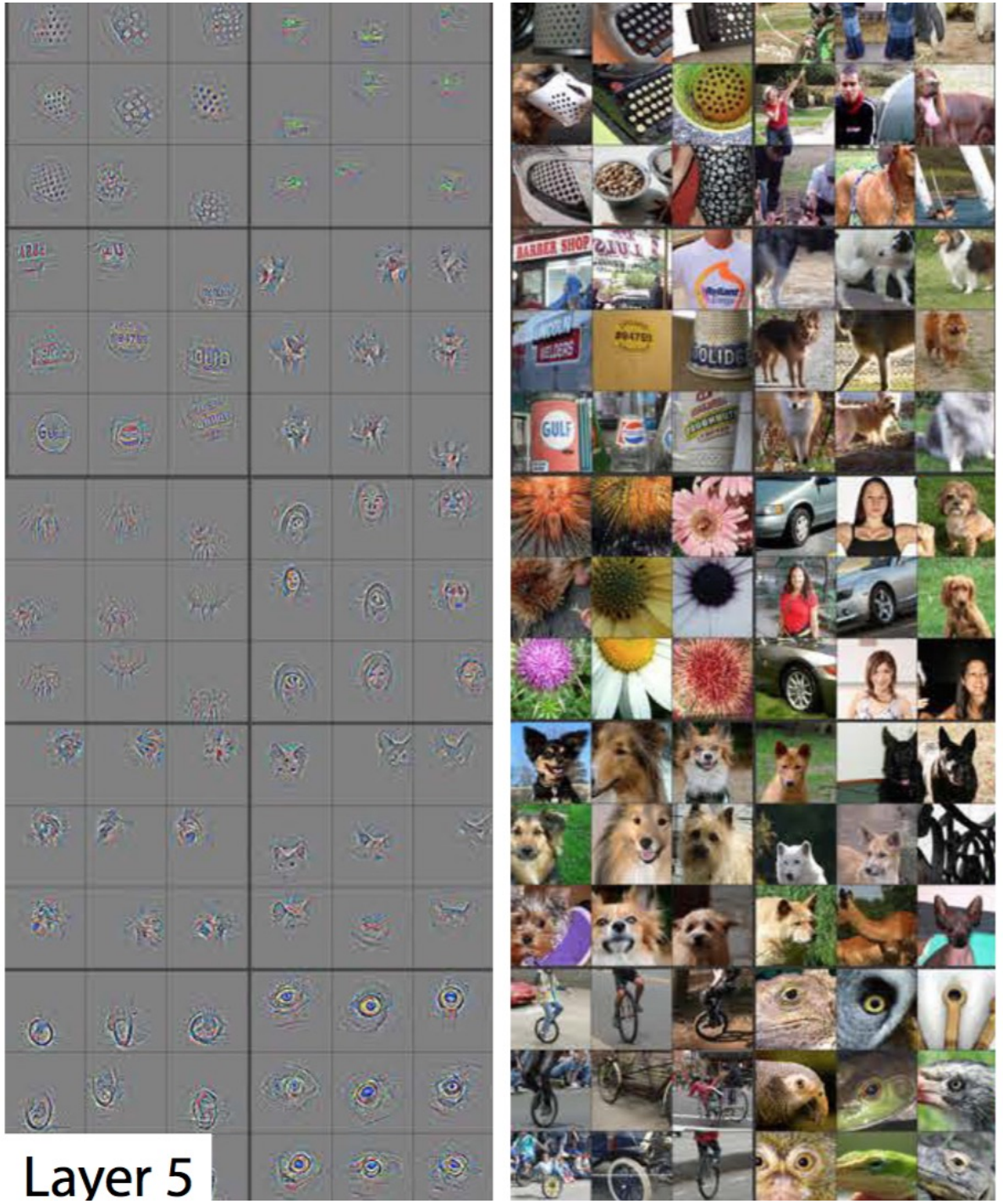
Deconv results

- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]



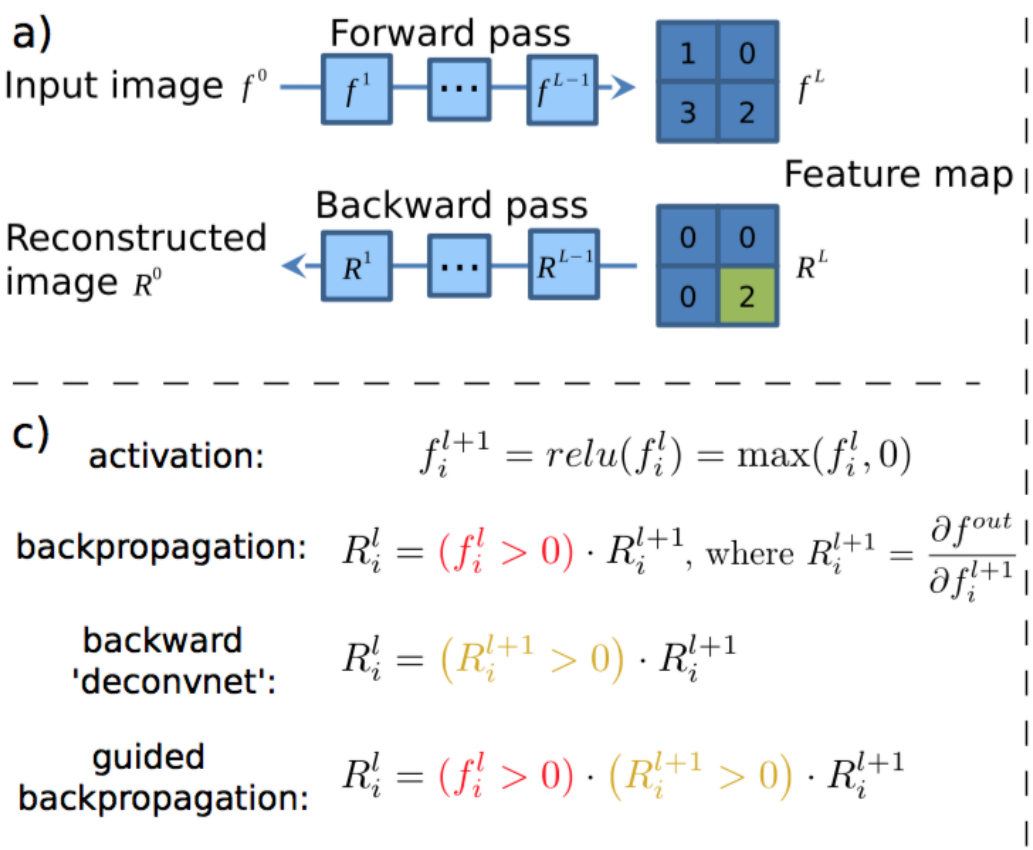
Deconv results

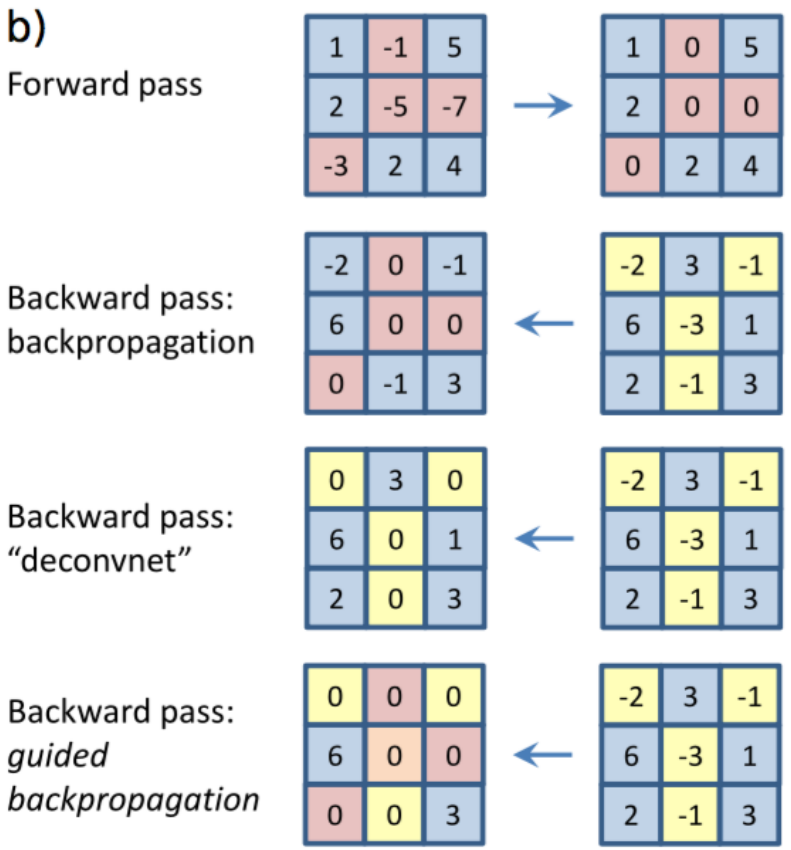
- [Visualizing and Understanding Convolutional Networks, Zeiler and Fergus 2013]



Deconv results: guided backprop

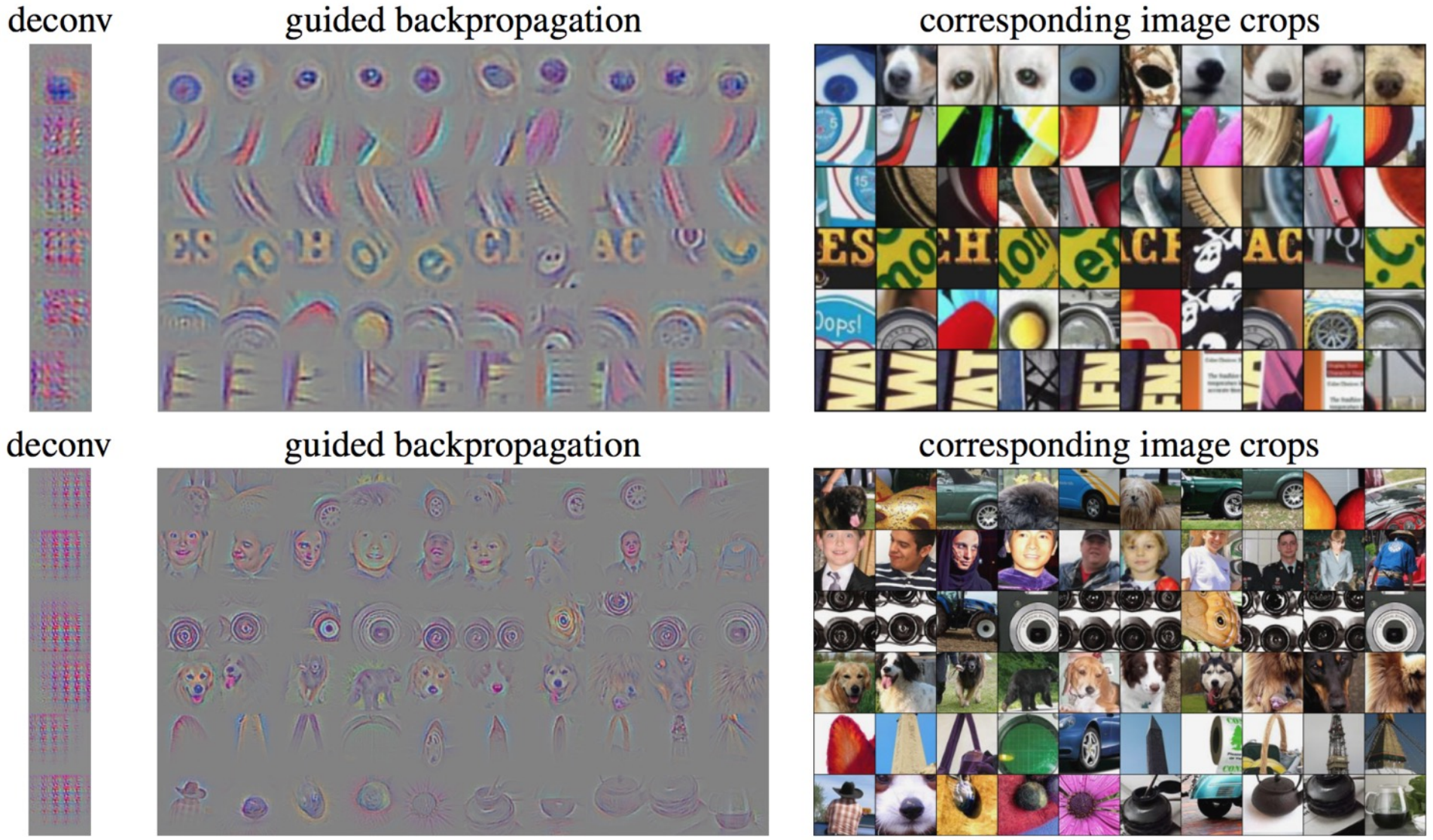
[Striving for Simplicity: The All Convolutional Net, Springenberg et al. 2015]





Deconv results: guided backprop

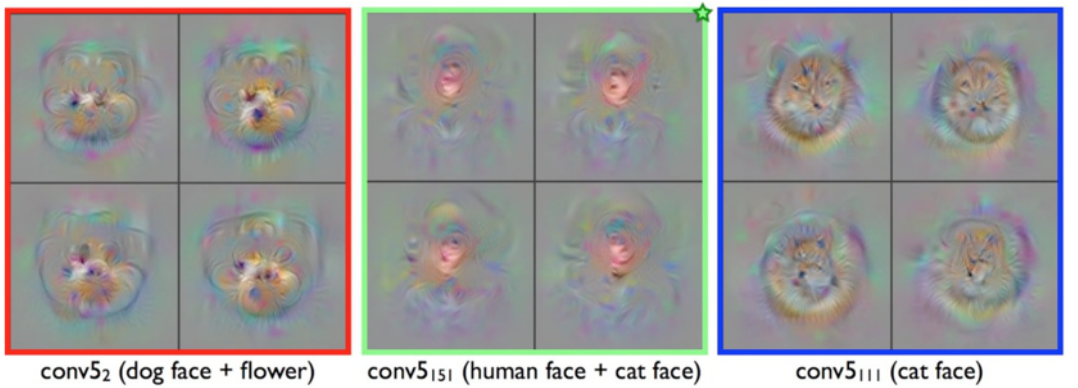
[Striving for Simplicity: The All Convolutional Net, Springenberg et al. 2015]



Visualizing activations

- http://yosinski.com/deepvis

- Check out this YouTube video https://www.youtube.com/watch?v=AgkfIQ4IGaM

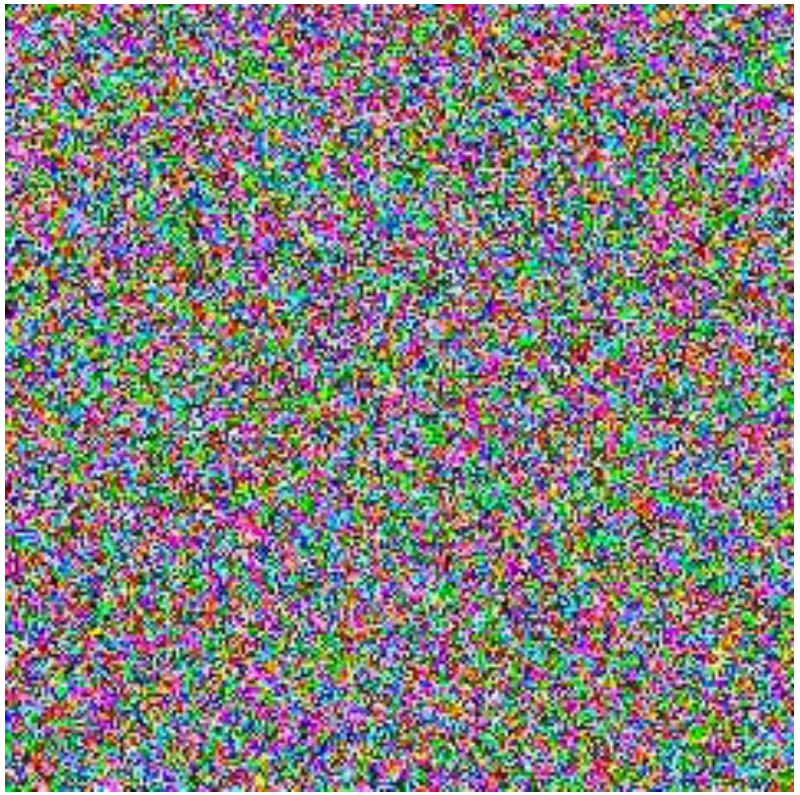


Visualizing neurons with image optimization

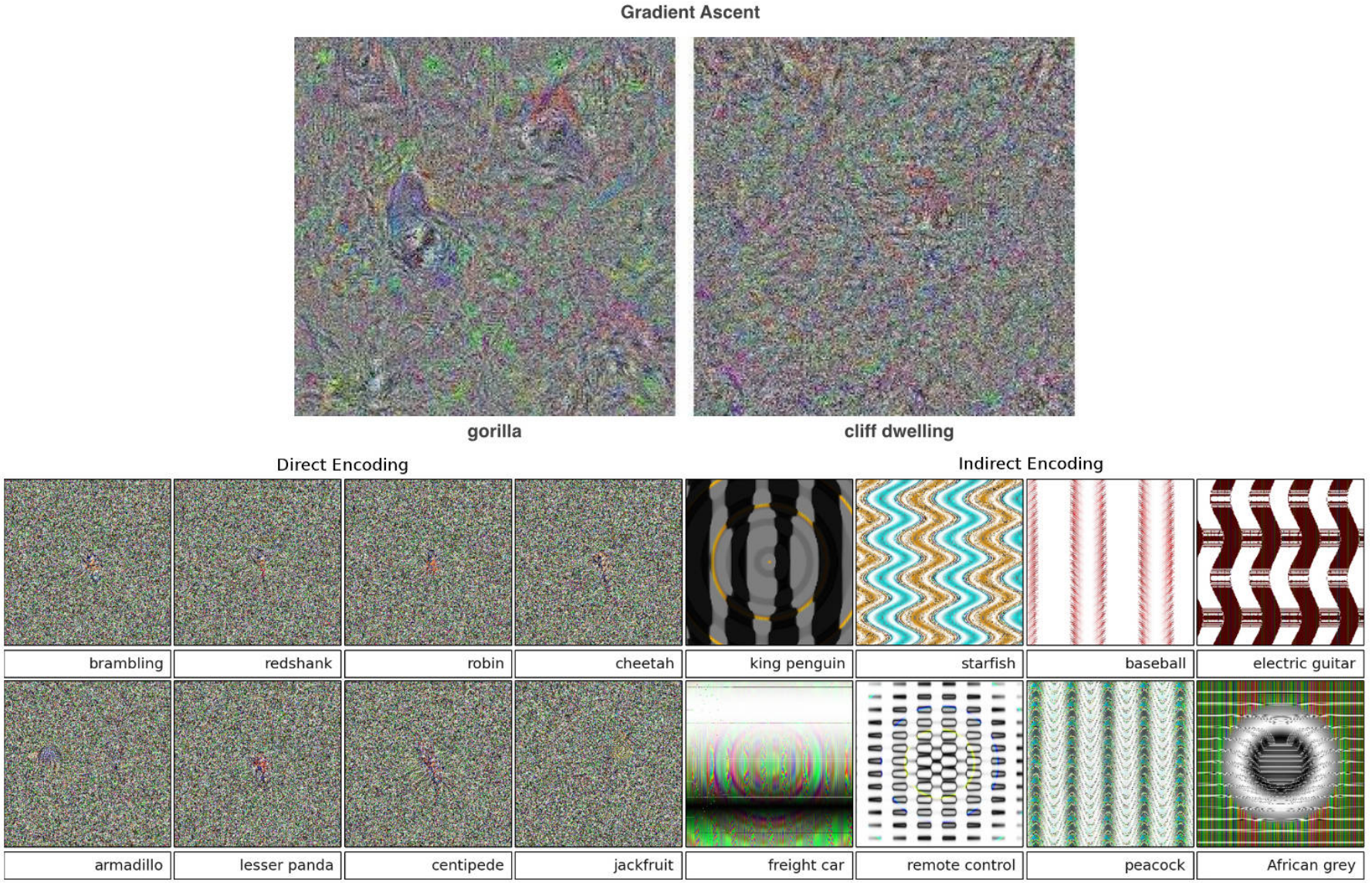
- Input: image x, network N
- Find image x that maximizes certain neuron's output (e.g. class score)
- Compute $x^* = arg\ max\ S_i(x)$

Visualizing neurons with image optimization

- Input: image x, network N
- Pick a neuron i (freeze others)
- Compute activation of neuron i by forwarding x though N: $a_i(x)$
- Compute gradient of $a_i(x)$ with respect to x: $da_i(x) / dx$
- Change x to increase activation in $a_i(x)$: $x \leftarrow x + \alpha\, da_i(x) / dx$
- Iterate 2—4…
- Look at x



Visualizing neurons with image optimization



Visualizing neurons with backprop and priors

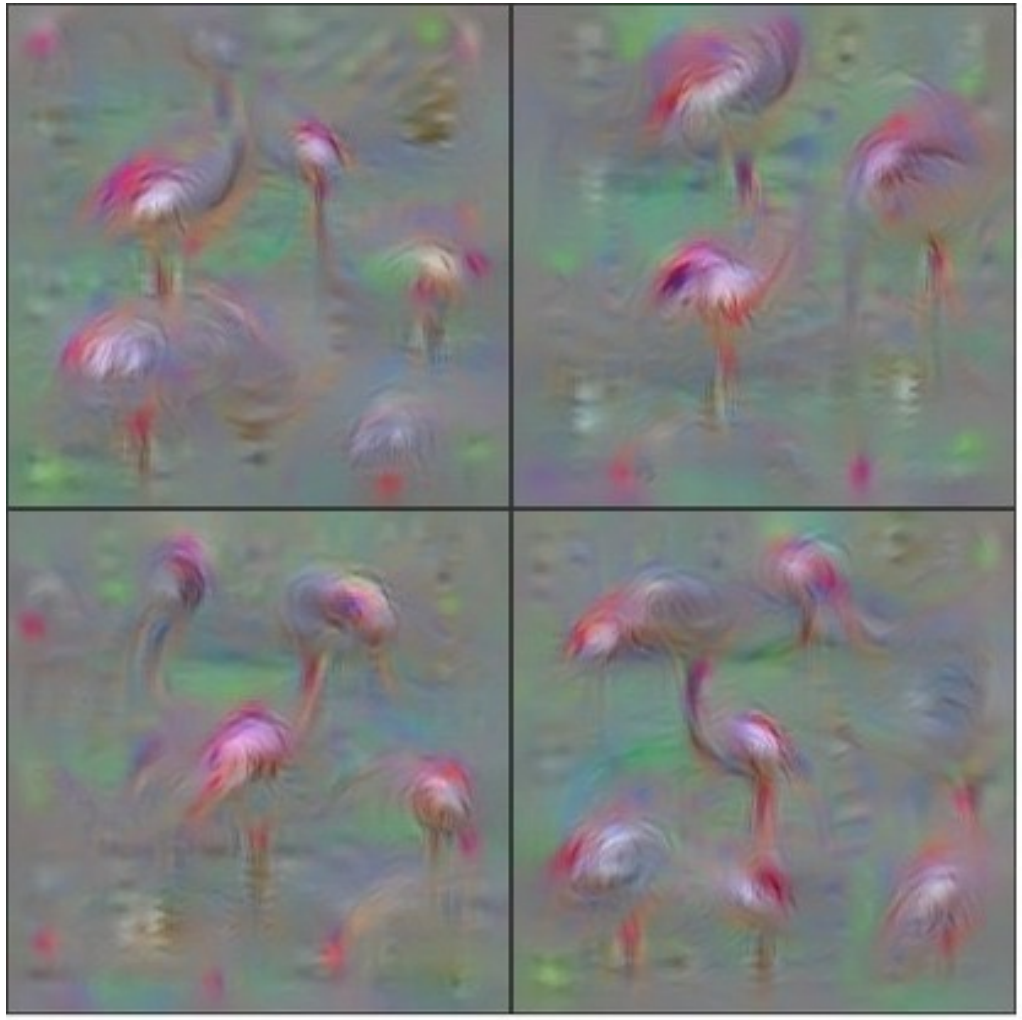
goose

ostrich

- [Deep Inside Convolutional Networks, Simonyan et at., 2014]: add L2-regularization: $x^* = arg\ max\ \{S_i(x) - \lambda \|x\|2\}$

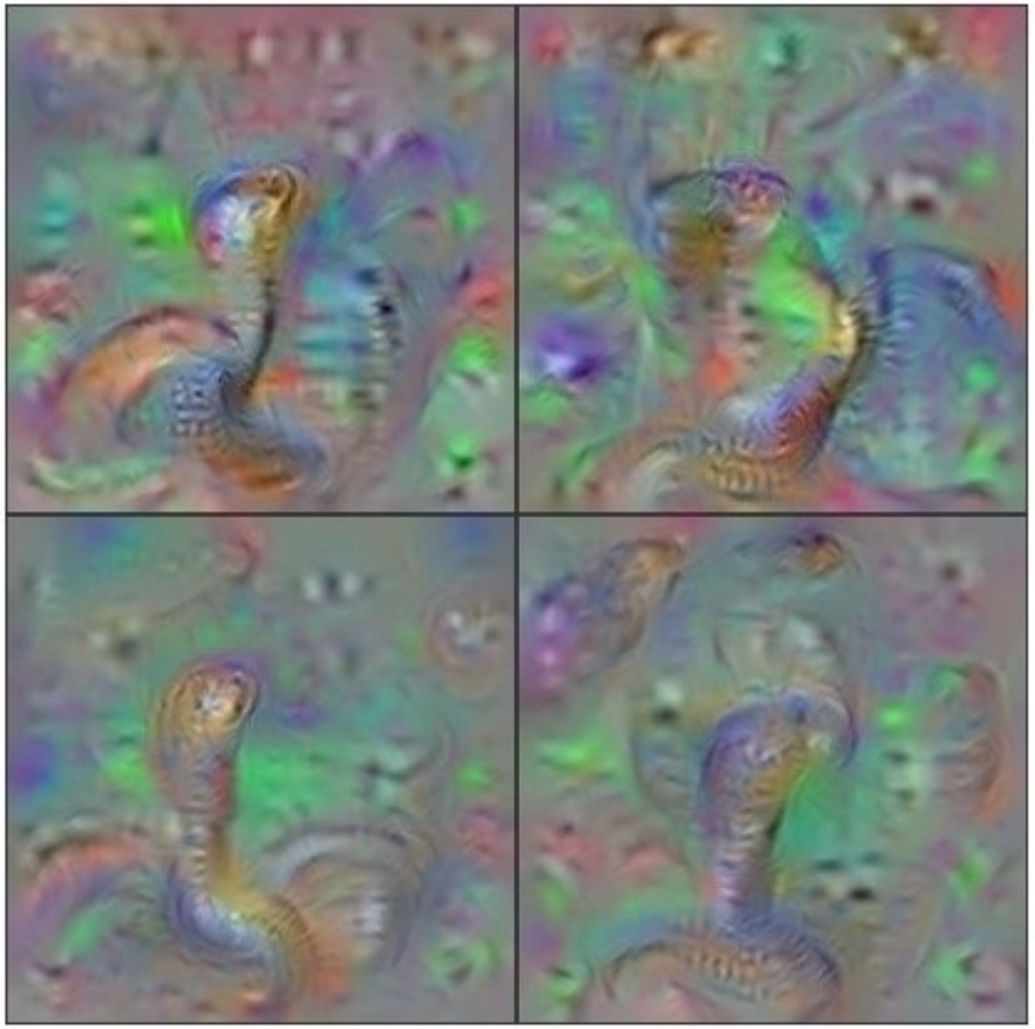
Visualizing neurons with regularized image optimization

- Understanding Neural Networks Through Deep Visualization:
  replace L2 with other stuff
  (gaussian blur etc.)
  DO check out
  http://yosinski.com/deepvis-fc8



Visualizing neurons with regularized image optimization

- Understanding Neural Networks Through Deep Visualization:
  replace L2 with other stuff
  (gaussian blur etc.)
  DO check out
  http://yosinski.com/deepvis-fc8

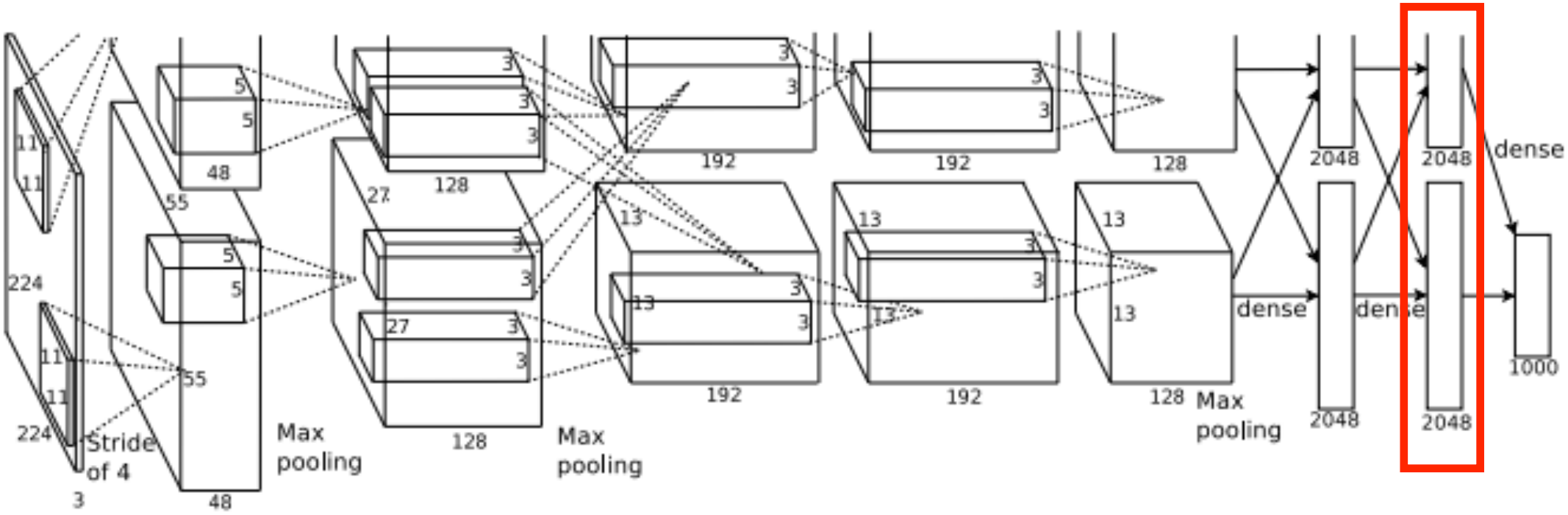


Why does this matter?

- Important features such as face detectors and text detectors are learned
  - Bowties are paired with faces
  - Bookshelves are paired with books with text on them
- Some neurons can be interpreted as they convey local information
- Networks learn a lot about the global structure of the image
  - e.g. outline of the starfish and the fact that it has five legs

3

Question: given the CNN descriptor, is it possible to reconstruct the entire input image?



Question: given the CNN descriptor, is it possible to reconstruct the entire input image?

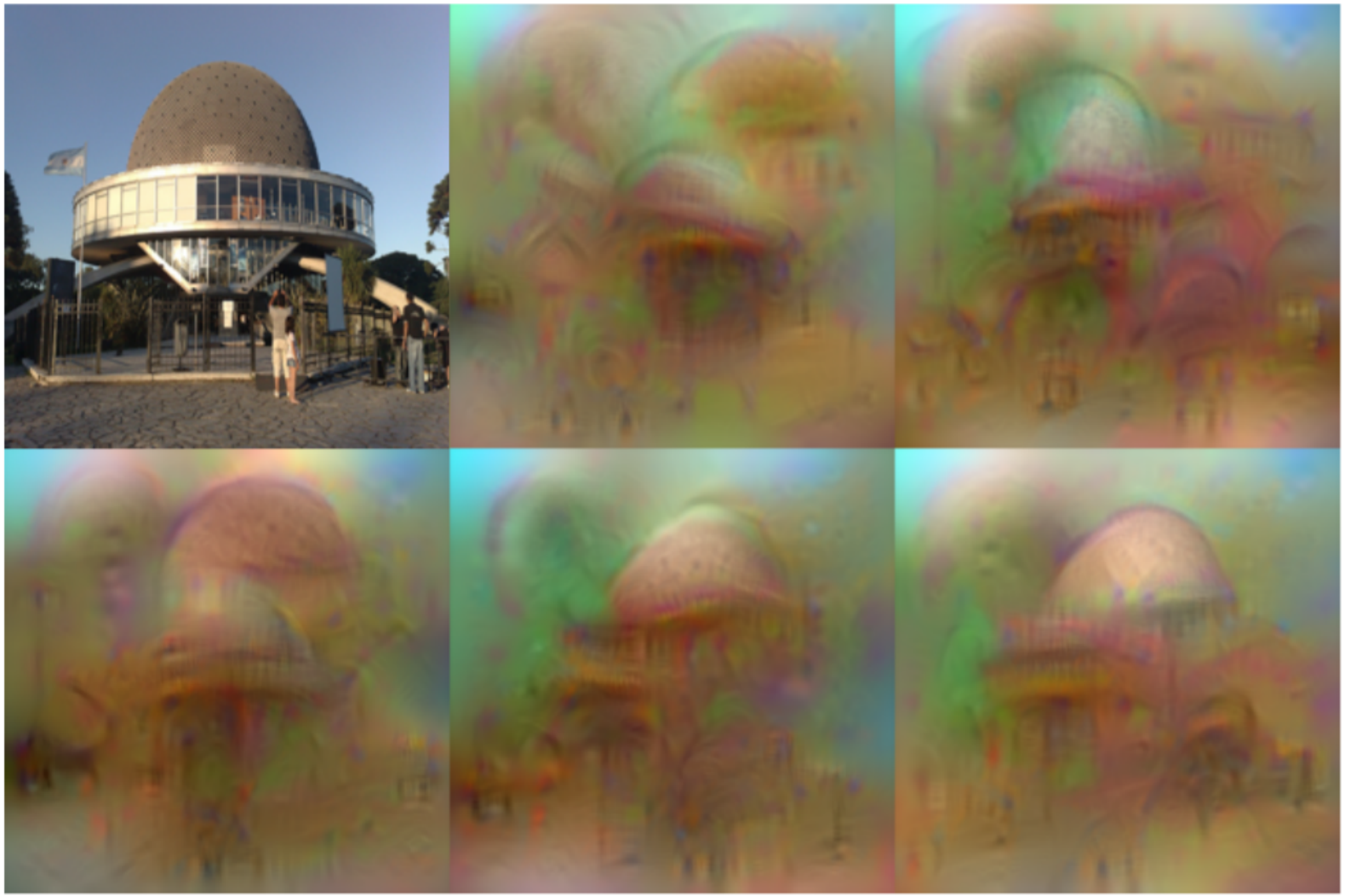
$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^{H \times W \times C}}{\operatorname{argmin}} \ell(\Phi(\mathbf{x}), \Phi_0) + \lambda \mathcal{R}(\mathbf{x})$$

$$\ell(\Phi(\mathbf{x}), \Phi_0) = \|\Phi(\mathbf{x}) - \Phi_0\|^2$$

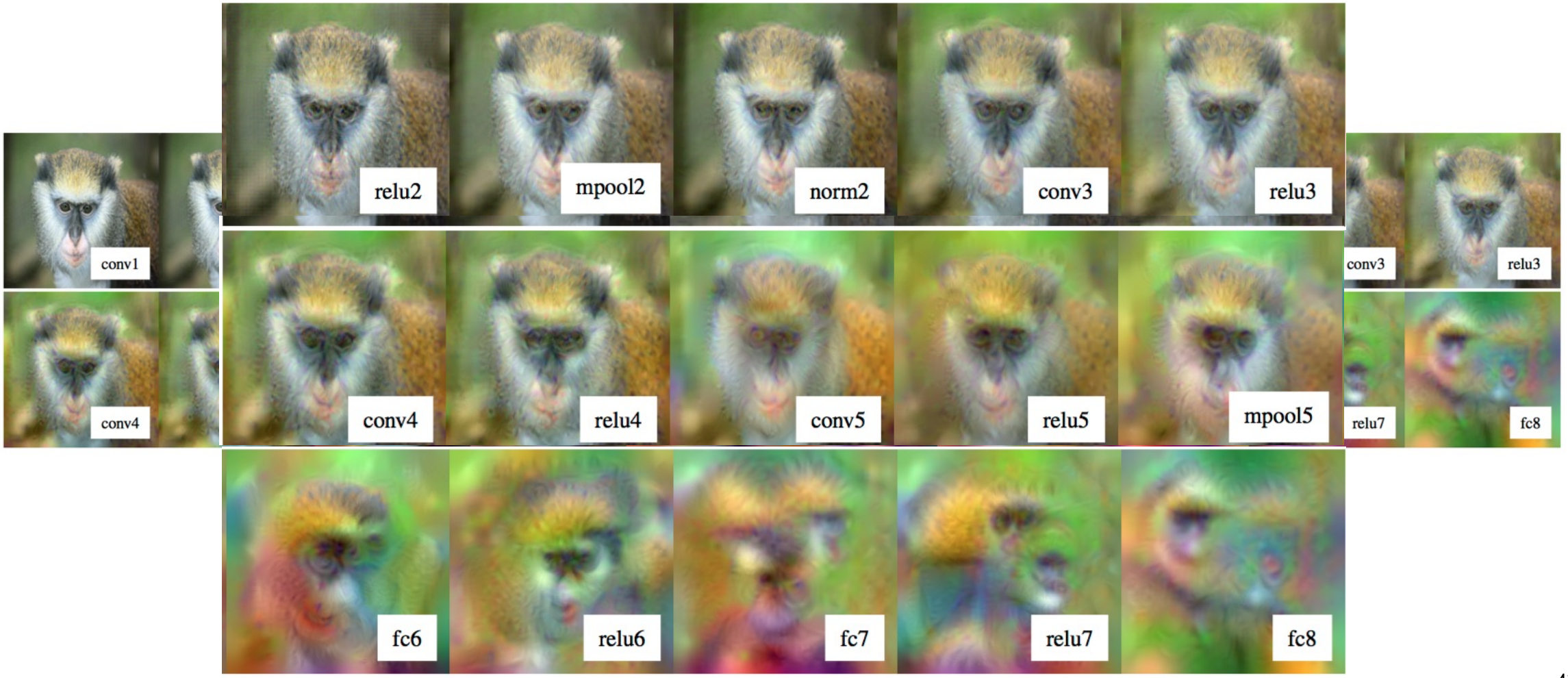
- [Understanding Deep Image Representations by Inverting Them, Mahendran & Vedaldi, 2014]

Reconstructions (5 different variants) are all equivalent from the models' viewpoint

- [Understanding Deep Image Representations by Inverting Them, Mahendran & Vedaldi, 2014]



Reconstructions (5 different variants) are all equivalent from the models' viewpoint



max-pool-5 reconstructions
(much spatial information is still present!)

[Understanding Deep Image Representations by Inverting Them, Mahendran & Vedaldi, 2014]

An intermediate summary

- Visualization may help you inspect what's 'inside' the CNN
- Deconvnets are an old (but reliable) approach to predicting the activation inducing patches
- Some of the inputs may even be reconstructed from the CNN output!

(Some) information theory of neural networks

Some Information Theory basics

- The KL-distribution divergence:

for any two distributions p(x) & q(x) over X:

$$D[p(x)\,\|\,q(x)] = \sum_x p(x)\log\frac{p(x)}{q(x)} \geq 0$$

- The Mutual Information:

for any two random variables, X , Y:

$$I(X;Y) = D[p(x,y)\,\|\,p(x)p(y)] = D[p(x\,|\,y)\,\|\,p(x)] = D[p(y\,|\,x)\,\|\,p(y)] = \mathrm{H(X)} - \mathrm{H(X\,|\,Y)}$$

- Data Processing Inequality (DPI) & Invariance:

for any Markov chain: $X \rightarrow Y \rightarrow Z$:

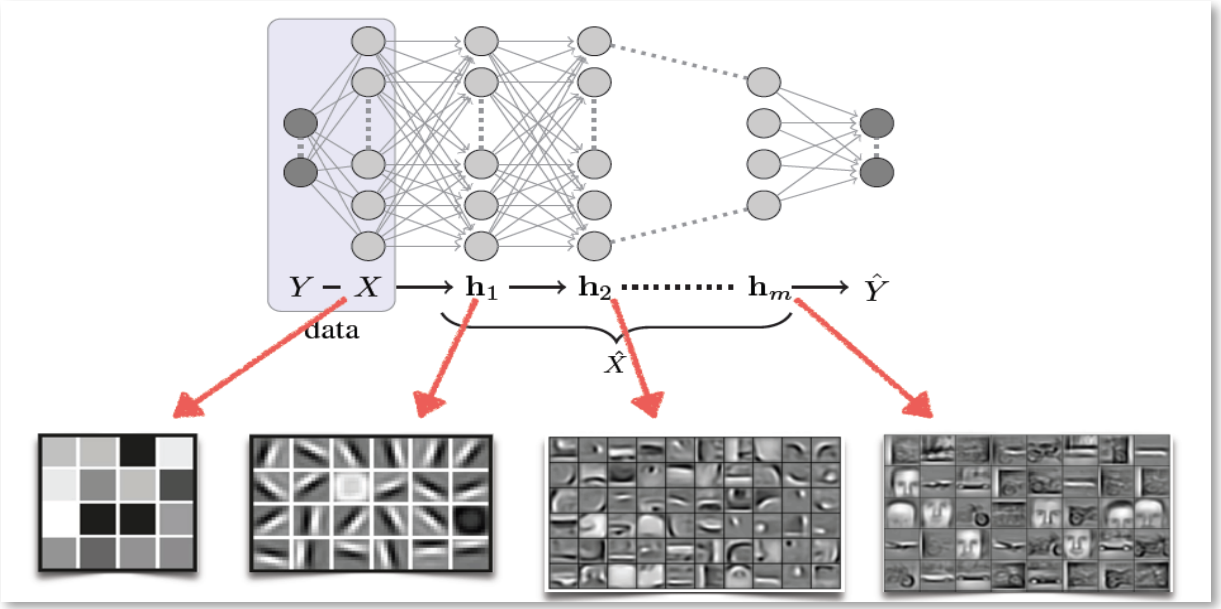
$$I(X;Y) \geq I(X;Z)$$

Reparametrization Invariance, for invertible ϕ, ψ:

$$I(X;Y) = I(\phi(X);\psi(Y))$$

Slide credit: Naftali Tishby

What do the DNN Layers represent?



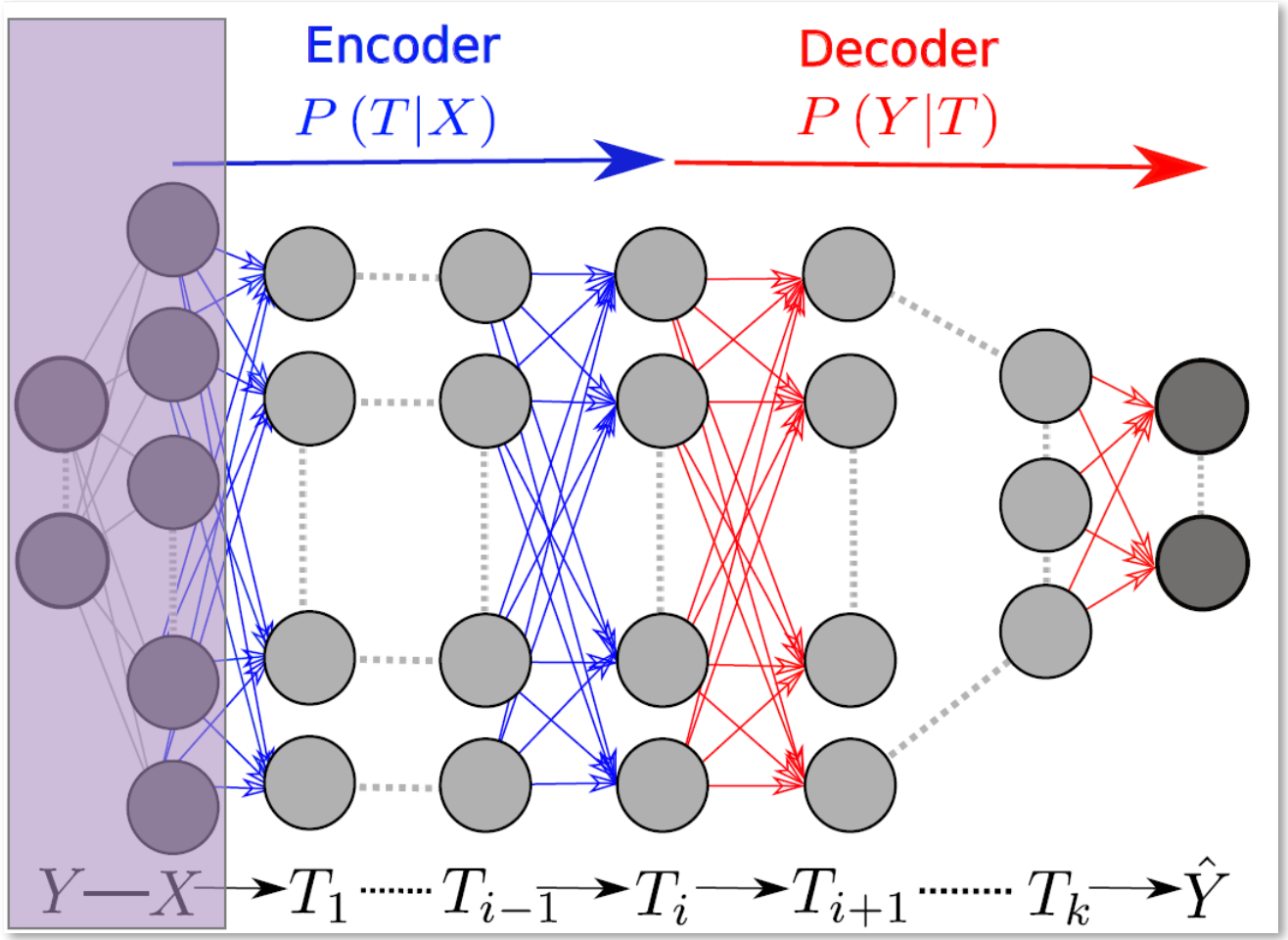
Data Processing Inequalities:

$$H(X) \geq I(X;h_i) \geq I(X;h_{i+1}) \geq I(X;h_{i+2}) \geq \ldots$$
$$I(X;Y) \geq I(h_i;Y) \geq I(h_{i+1};Y) \geq I(h_{i+2};Y) \geq \ldots$$

- A Markov chain of topologically distinct [soft] partitions of the input variable X.
- Successive Refinement of Relevant Information
- Individual neurons can be easily “scrambled” within each layer

Slide credit: Naftali Tishby

Each layer is characterized by its Encoder & Decoder Information



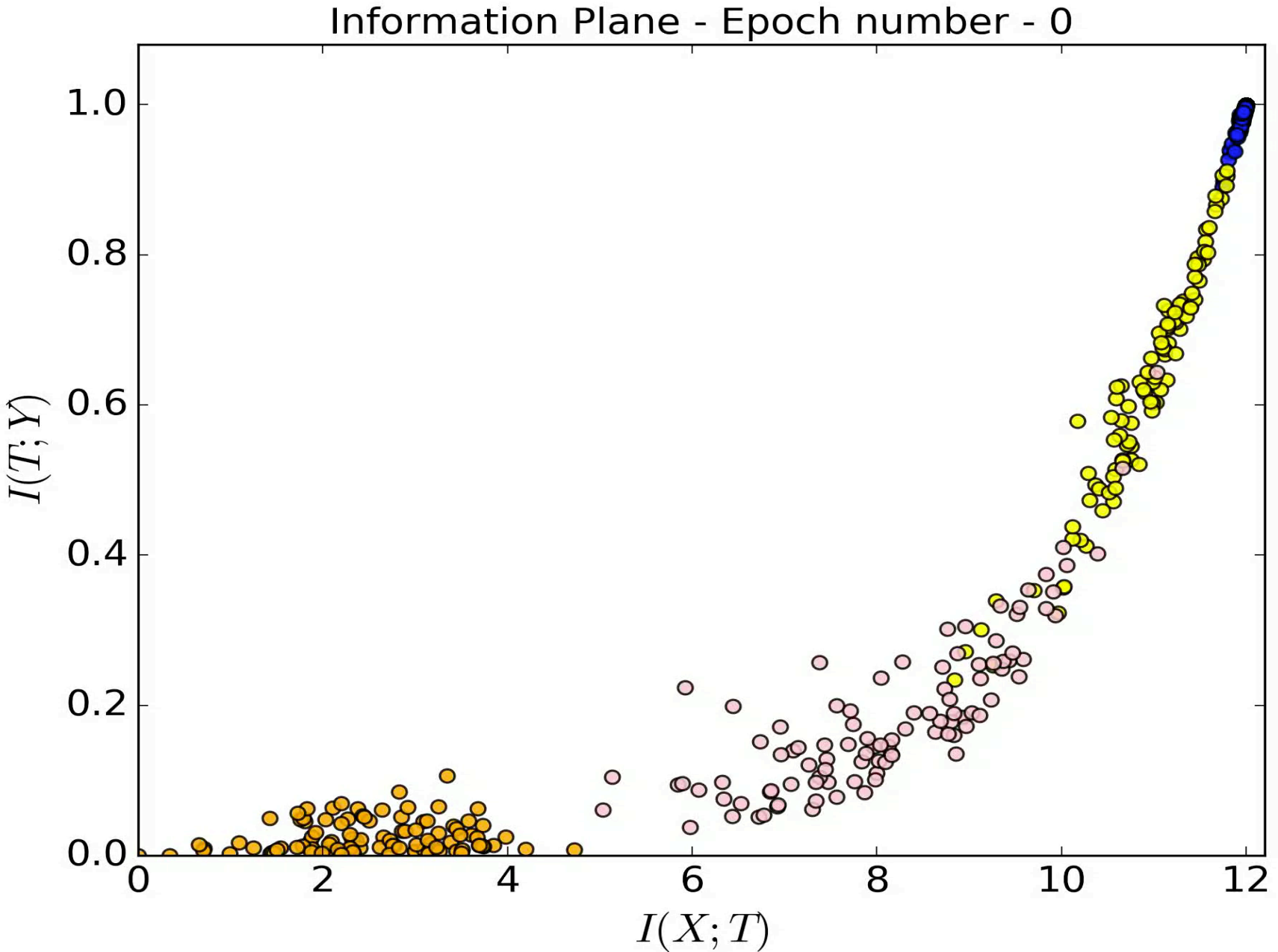
Theorem (Information Plane):

*For large typical **X**, the sample complexity of a DNN is completely determined by the encoder mutual information, **I(X;T)**, of the last hidden layer; the accuracy (generalization error) is determined by the decoder information, **I(T;Y)**, of the last hidden layer.*

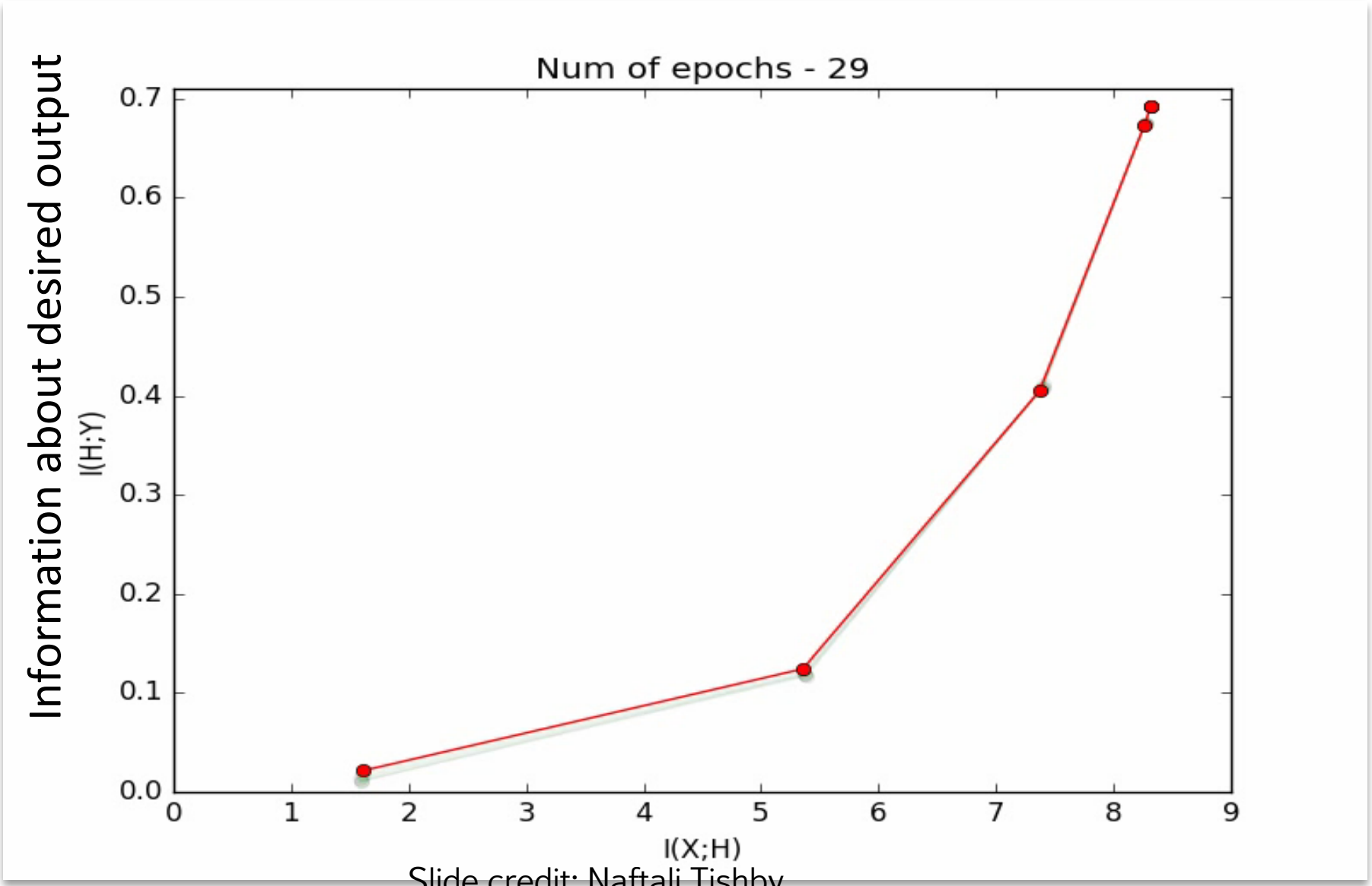
The complexity of the problem shifts from the decoder to the encoder, across the layers…

Slide credit: Naftali Tishby

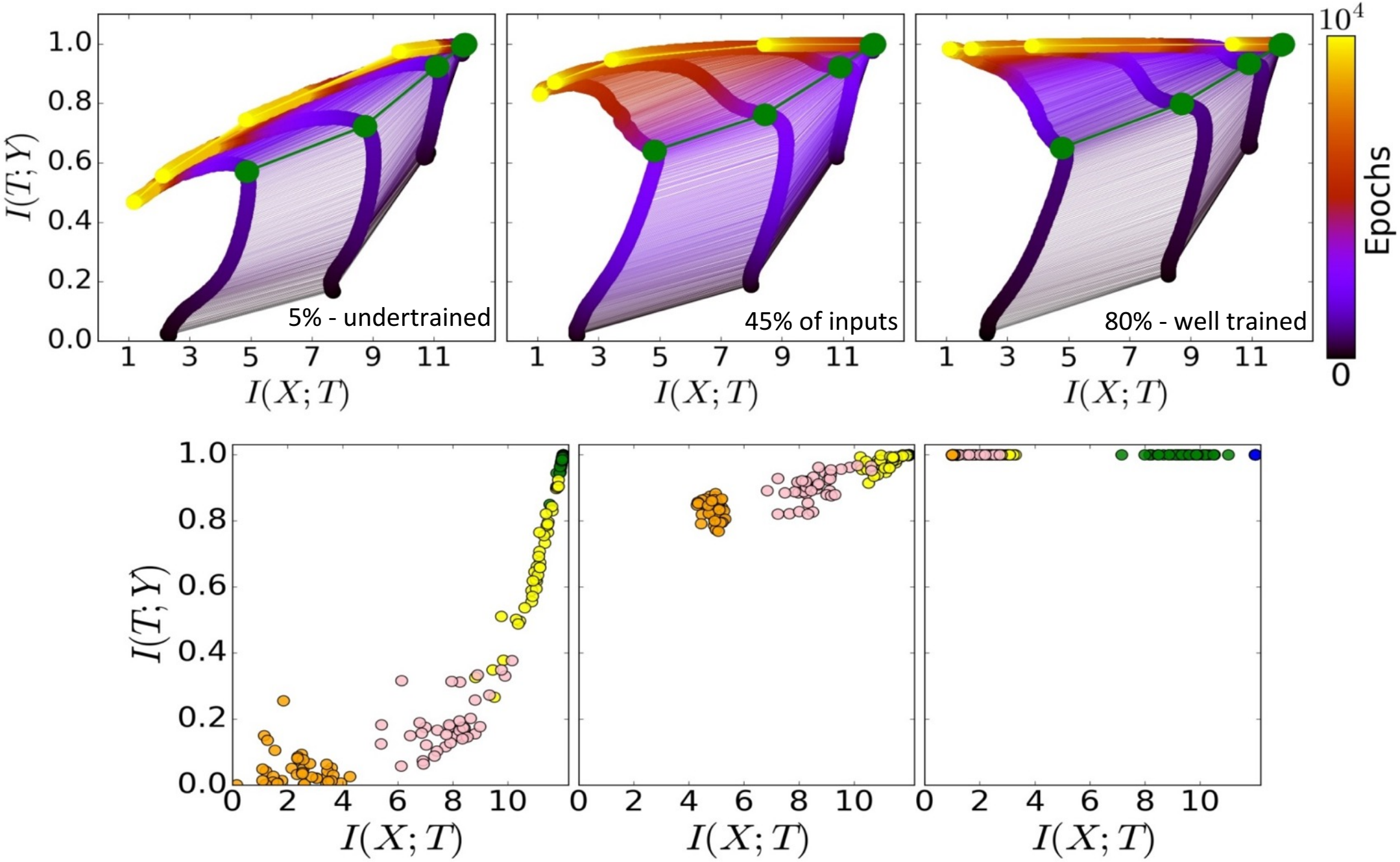
100 DNN Layers in Info-Plane without averaging



Averaged Layers in the Information-Plane

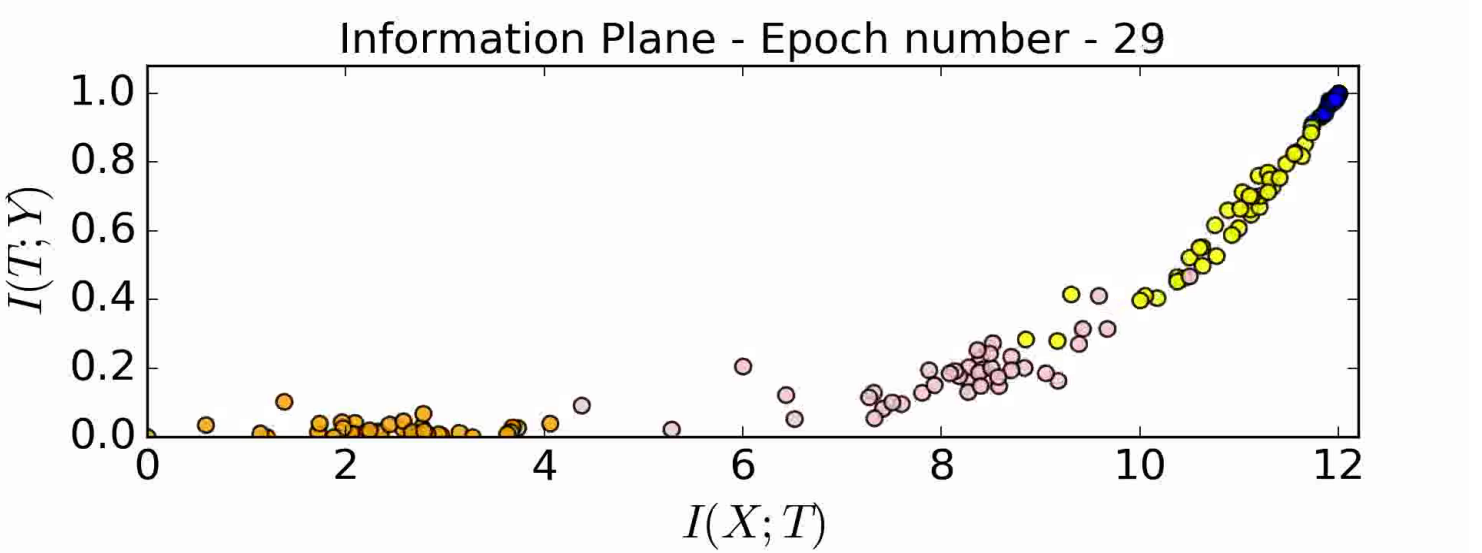


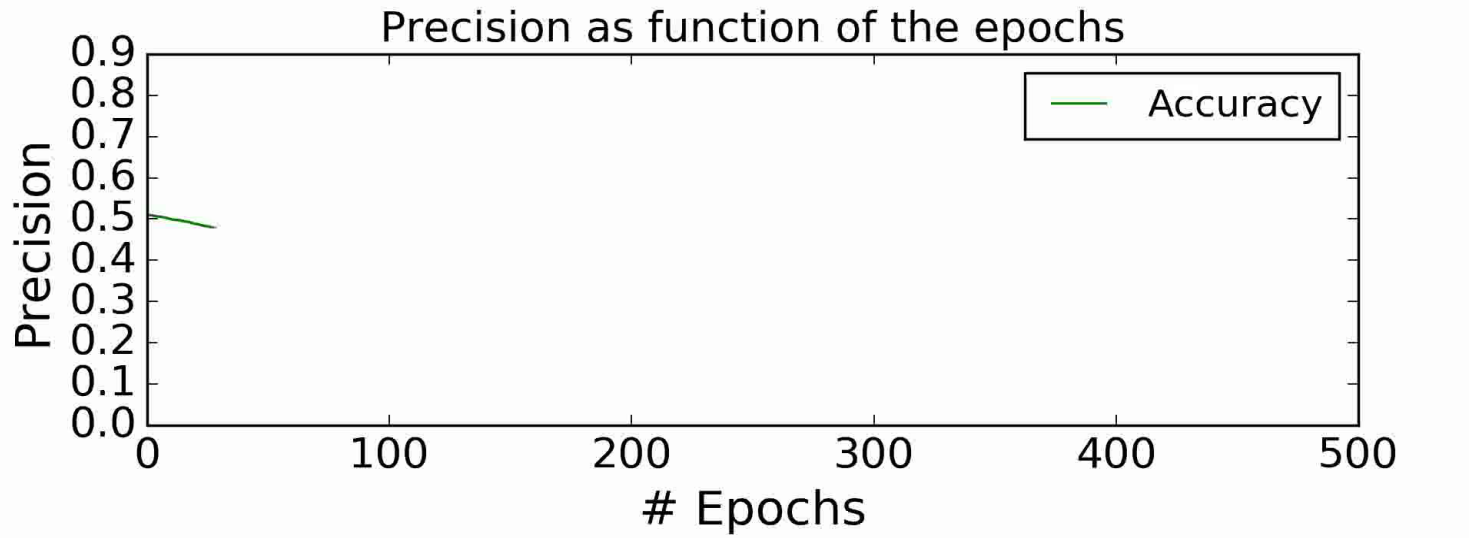
Slide credit: Naftali Tishby



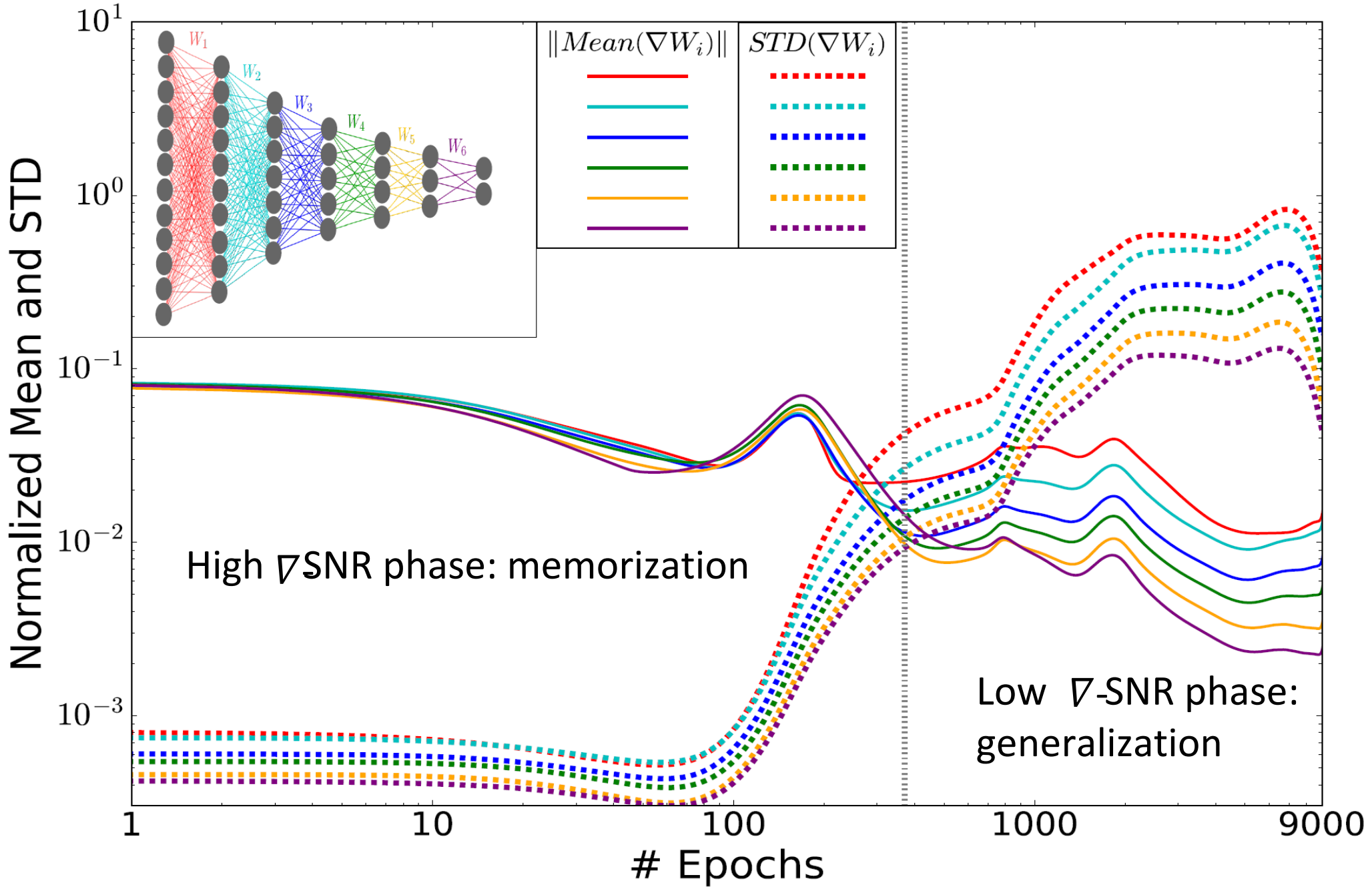
Slide credit: Naftali Tishby

Layers paths with generalization error - symmetric



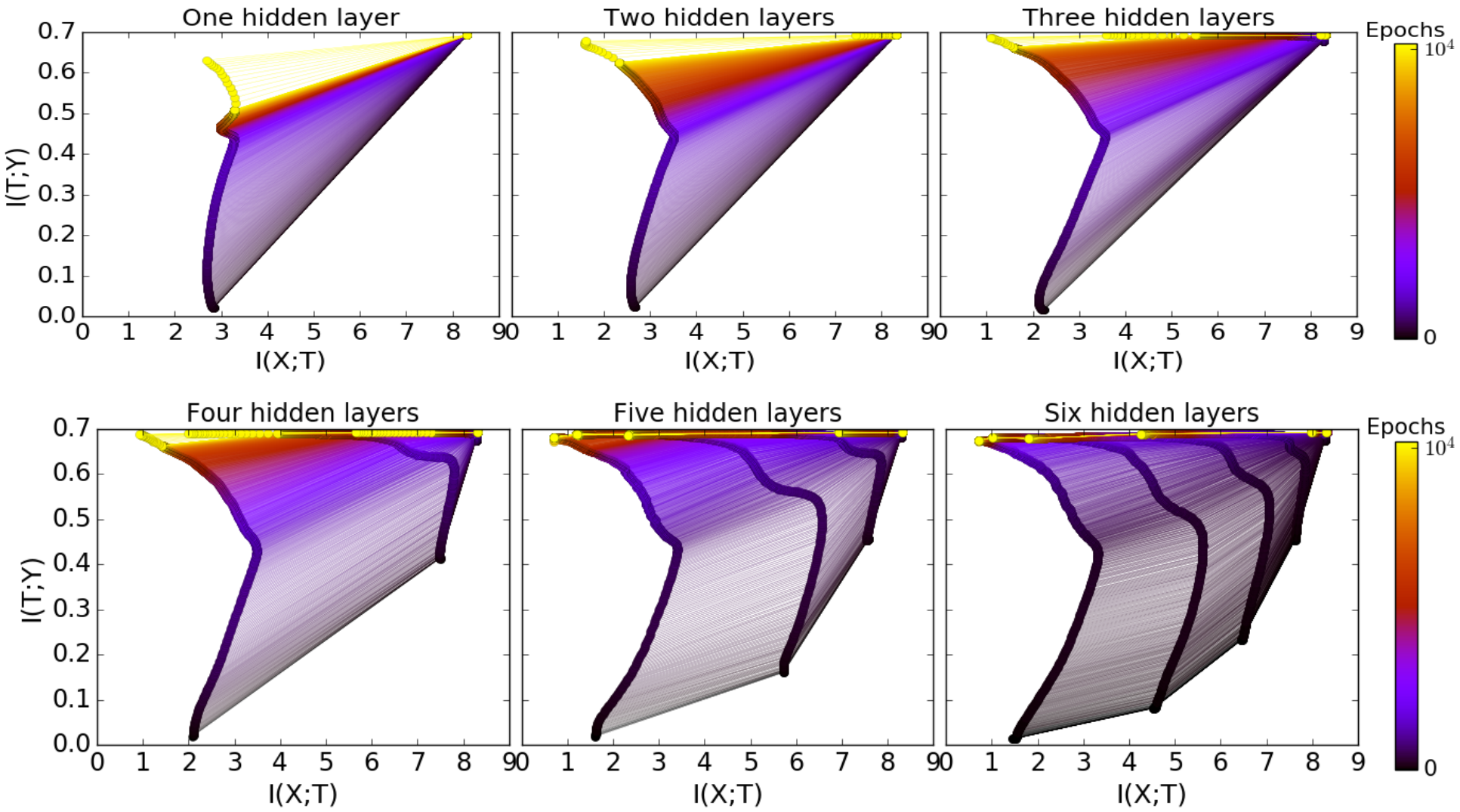


Slide credit: Naftali Tishby



Slide credit: Naftali Tishby

The benefit of the hidden layers



More layers take much FEWER training epochs for good generalization.

The optimization time depend super-linearly (exponentially?) on the compressed information, delta Ix, for each layer.

Slide credit: Naftali Tishby

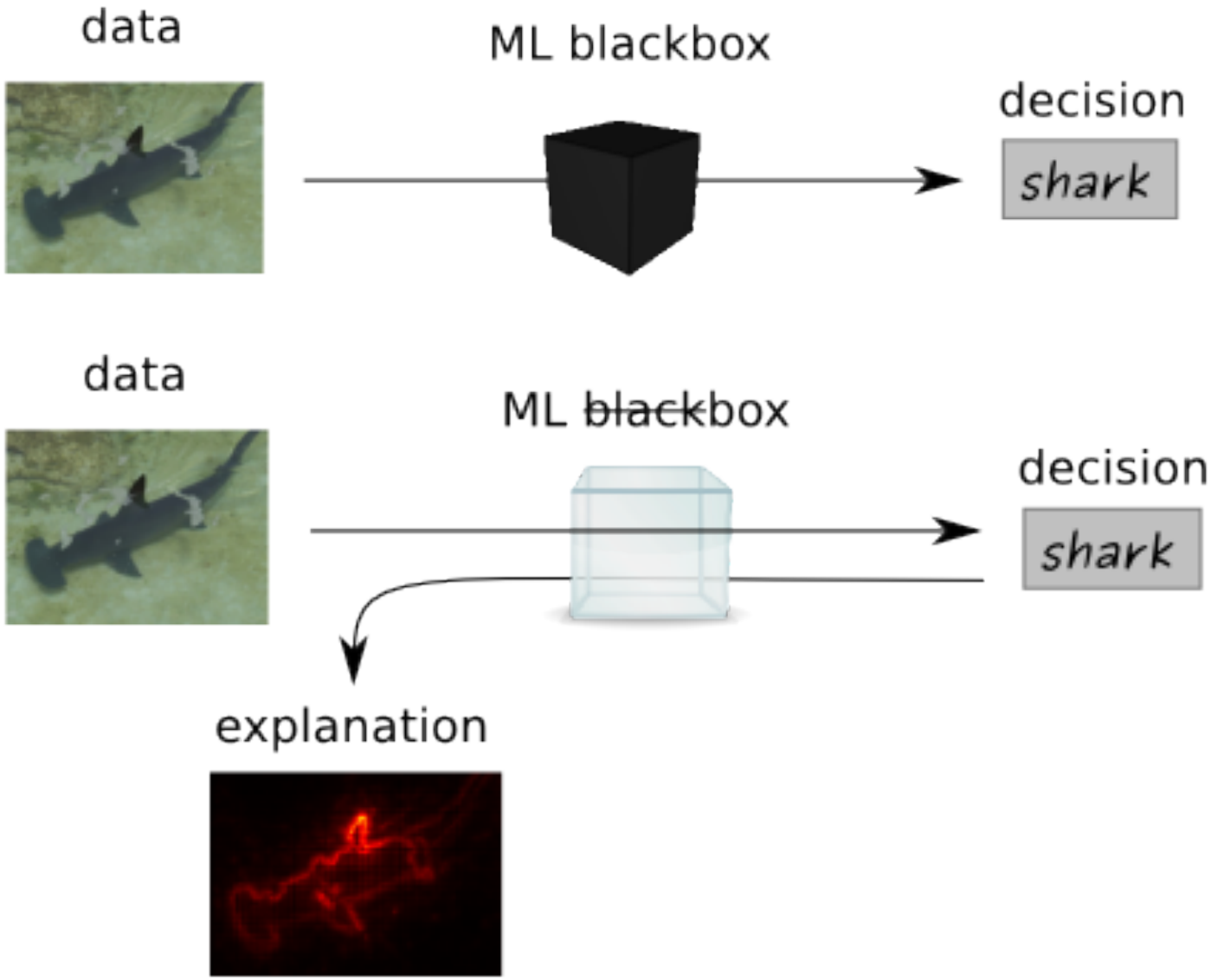
An intermediate summary

- Information theory may help establish important facts on the learning of deep models
- Deep networks train in two distinct phases: ***memorization*** and ***generalization***

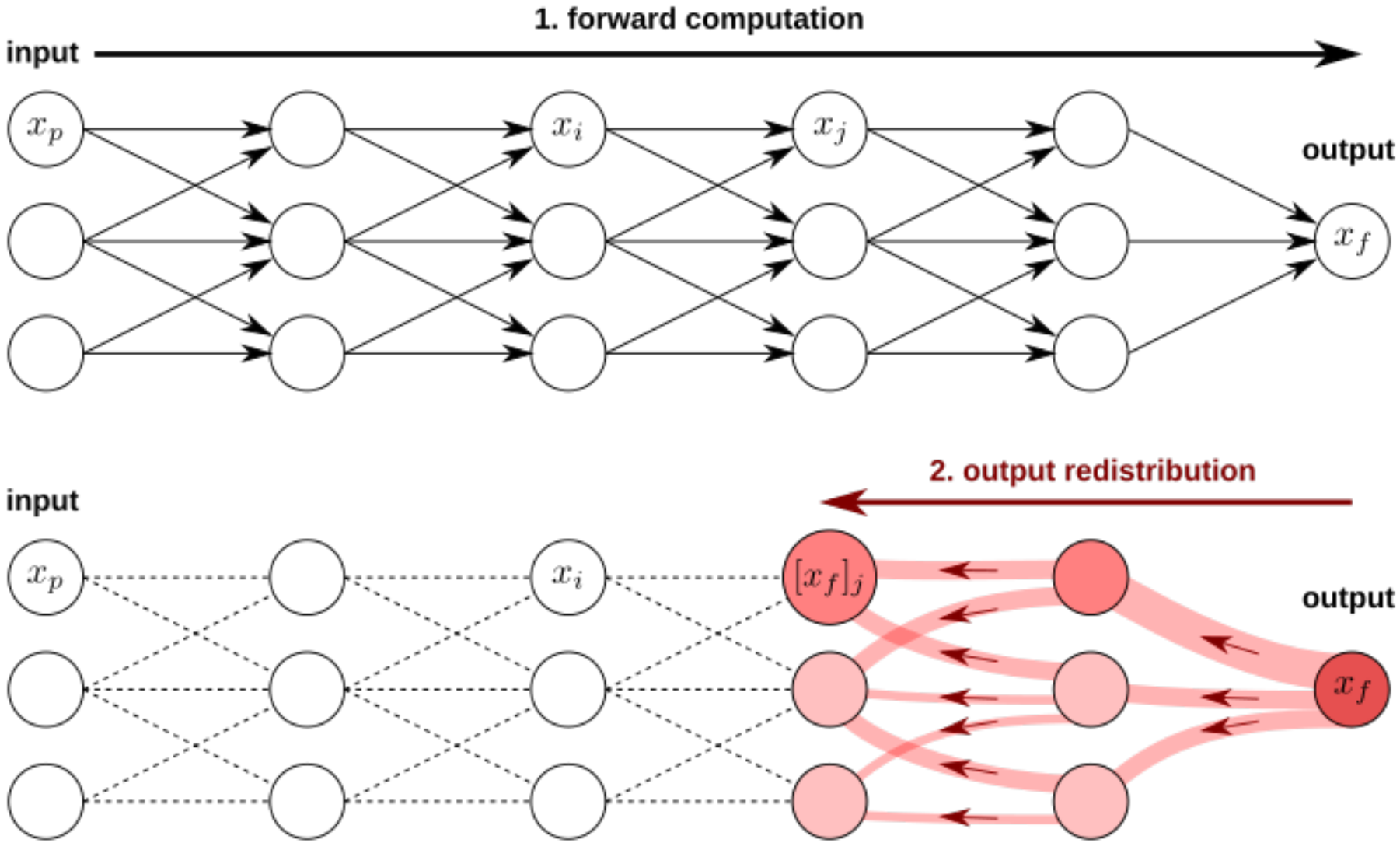
Deep Taylor decomposition

Deep Taylor: an idea

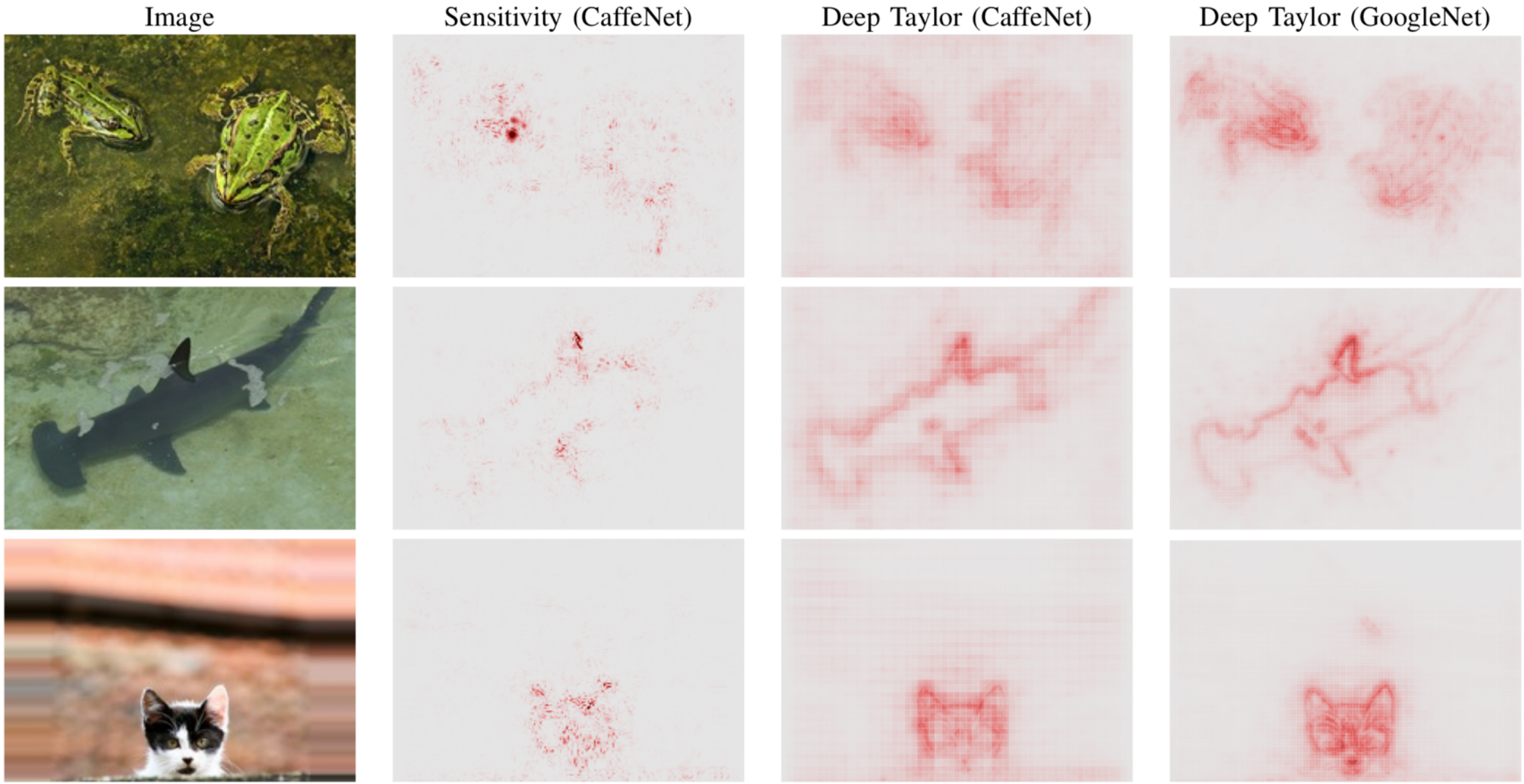
- Search for pixels in the input ***relevant*** for the prediction outcome



Deep Taylor: an idea



Deep Taylor visualizations



An intermediate summary

- [Explaining NonLinear Classification Decisions with Deep Taylor Decomposition, Montavon et al., 2015]
- Video available at https://www.youtube.com/watch?v=gy_Cb4Do_YE
- See the notebook at https://github.com/yandexdataschool/mlhep2018