SIGGRAPH ASIA 2018 TOKYO

CreativeAI: Deep Learning for Graphics

Feature Visualization

Niloy Mitra
UCL

Iasonas Kokkinos
UCL

Paul Guerrero
UCL

Nils Thuerey
TU Munich

Tobias Ritschel
UCL

UCL

Technische Universität München

Timetable

			Niloy	Paul	Nils
Theory and Basics	Introduction	2:15 pm	X	X	X
	Machine Learning Basics	~ 2:25 pm	X		
	Neural Network Basics	~ 2:55 pm			X
	Feature Visualization	~ 3:25 pm		X	
	Alternatives to Direct Supervision	~ 3:35 pm		X	

15 min. break

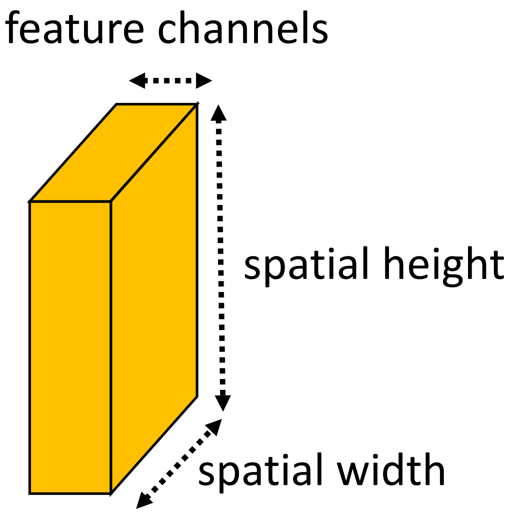
			Niloy	Paul	Nils
State of the Art	Image Domains	4:15 pm		X	
	3D Domains	~ 4:45 pm	X		
	Motion and Physics	~ 5:15 pm			X
	Discussion	~ 5:45 pm	X	X	X

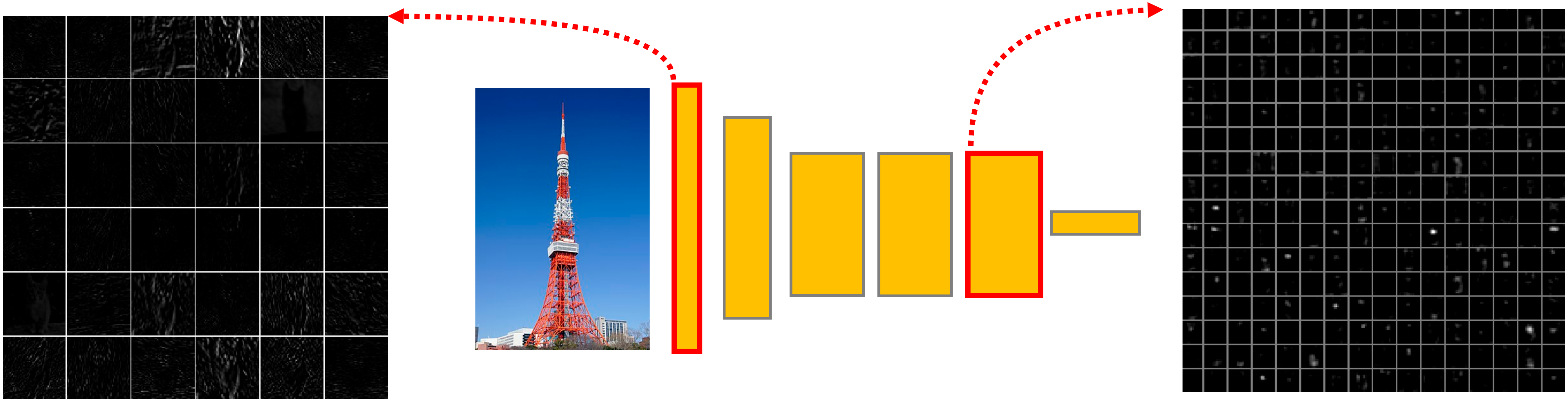
What to Visualize

- Features (activations)
- Weights (filter kernels in a CNN)
- Attribution: input parts that contribute to a given activation
- Inputs that maximally activate some class probabilities or features
- Inputs that maximize the error (adversarial examples)

Feature Samples

- In good training, features are usually sparse
- Can find "dead" features that never activate



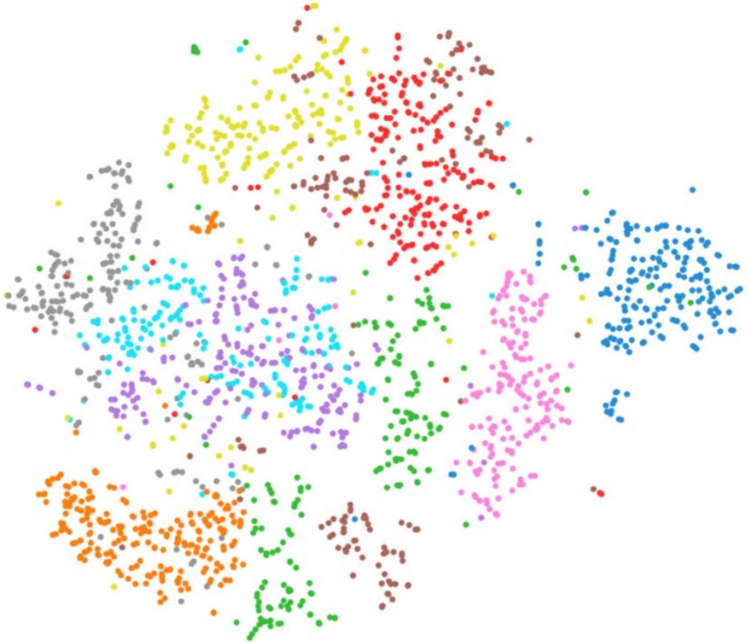


Images from: http://cs231n.github.io/understanding-cnn/

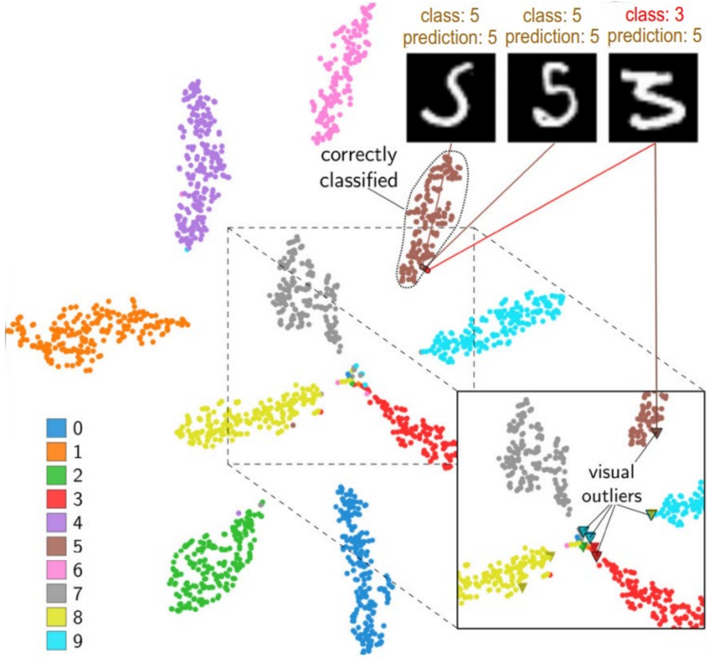
Feature Distribution using t-SNE

- Low-dimensional embedding of the features for visualization

t-SNE embedding of image features in a CNN layer



before training



after training

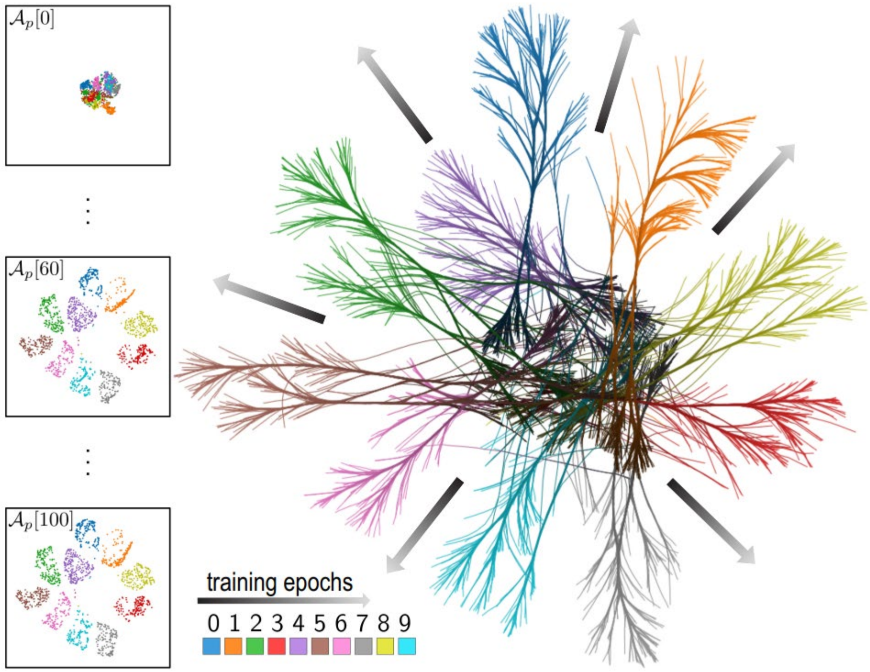
t-SNE embedding of MNIST (images of digits) features in a CNN layer, colored by class

Images from: https://cs.stanford.edu/people/karpathy/cnnembed/ and
Rauber et al. *Visualizing the Hidden Activity of Artificial Neural Networks*. TVCG 2017

Feature Distribution using t-SNE

- Low-dimensional embedding of the features for visualization

t-SNE embedding of image features in a CNN layer



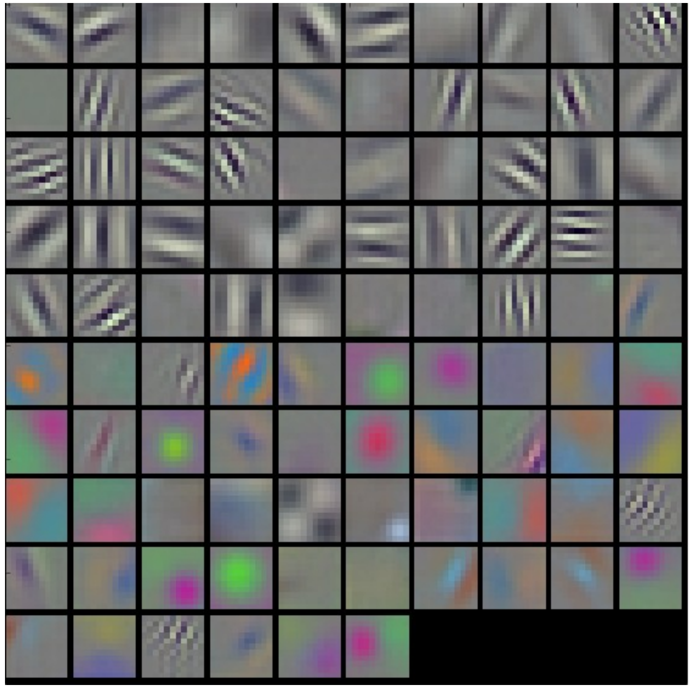
evolution during training

t-SNE embedding of MNIST (images of digits) features in a CNN layer, colored by class

Images from: https://cs.stanford.edu/people/karpathy/cnnembed/ and
Rauber et al. *Visualizing the Hidden Activity of Artificial Neural Networks*. TVCG 2017

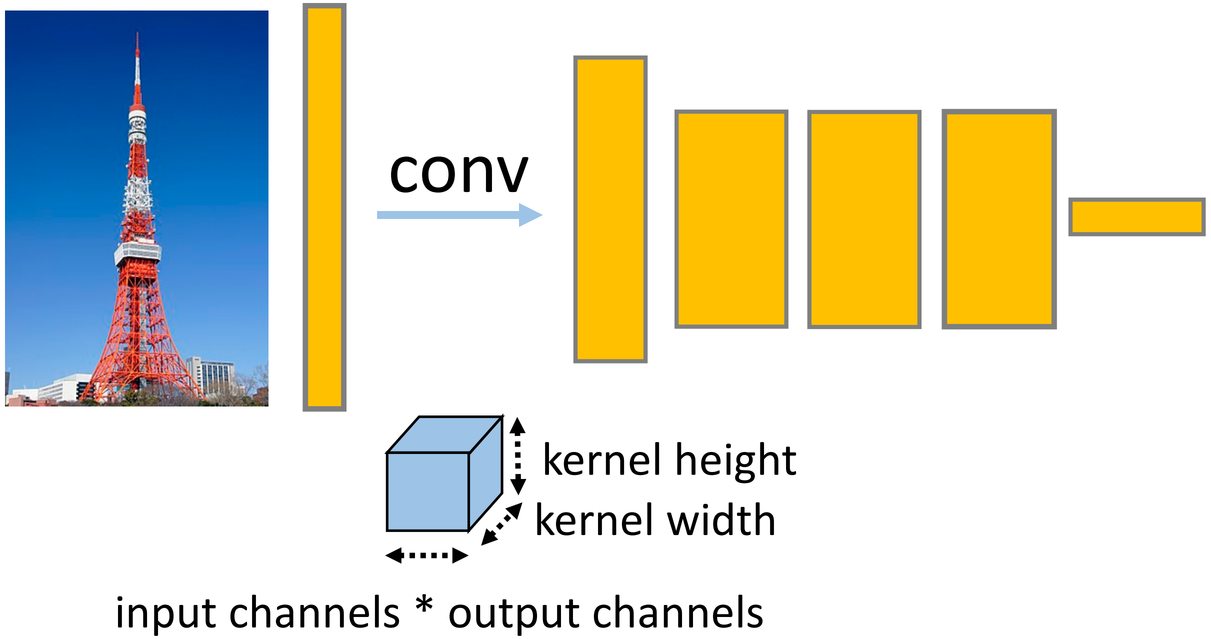
Weights (Filter Kernels)

- Useful for CNN kernels, not useful for fully connected layers
- Kernels are typically smooth and diverse after a successful training



first layer filters of AlexNet

Images from: http://cs231n.github.io/understanding-cnn/



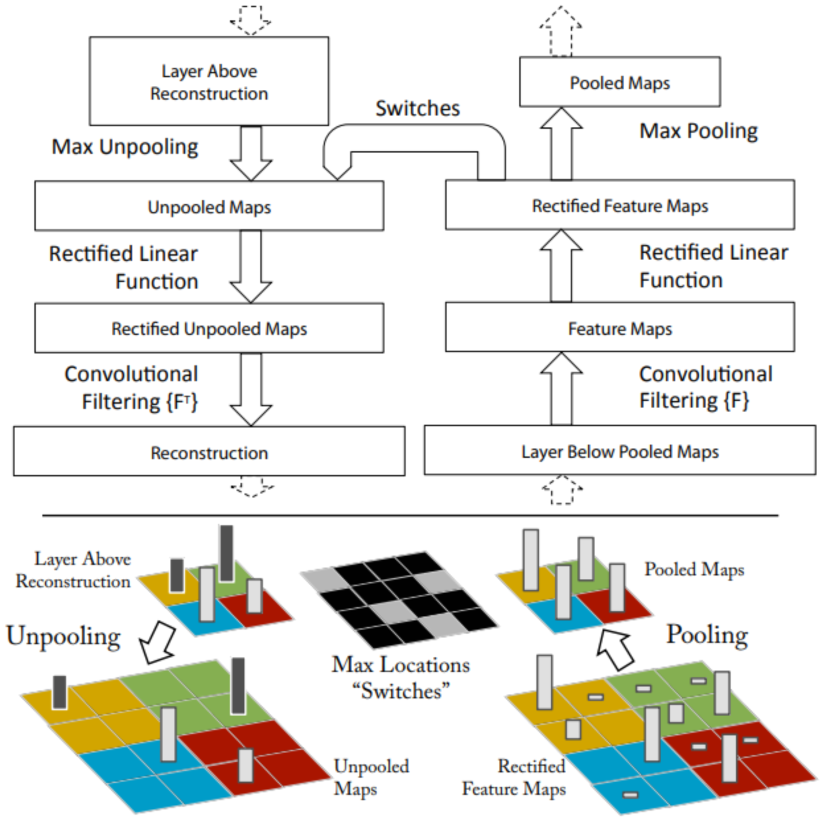
Code Examples

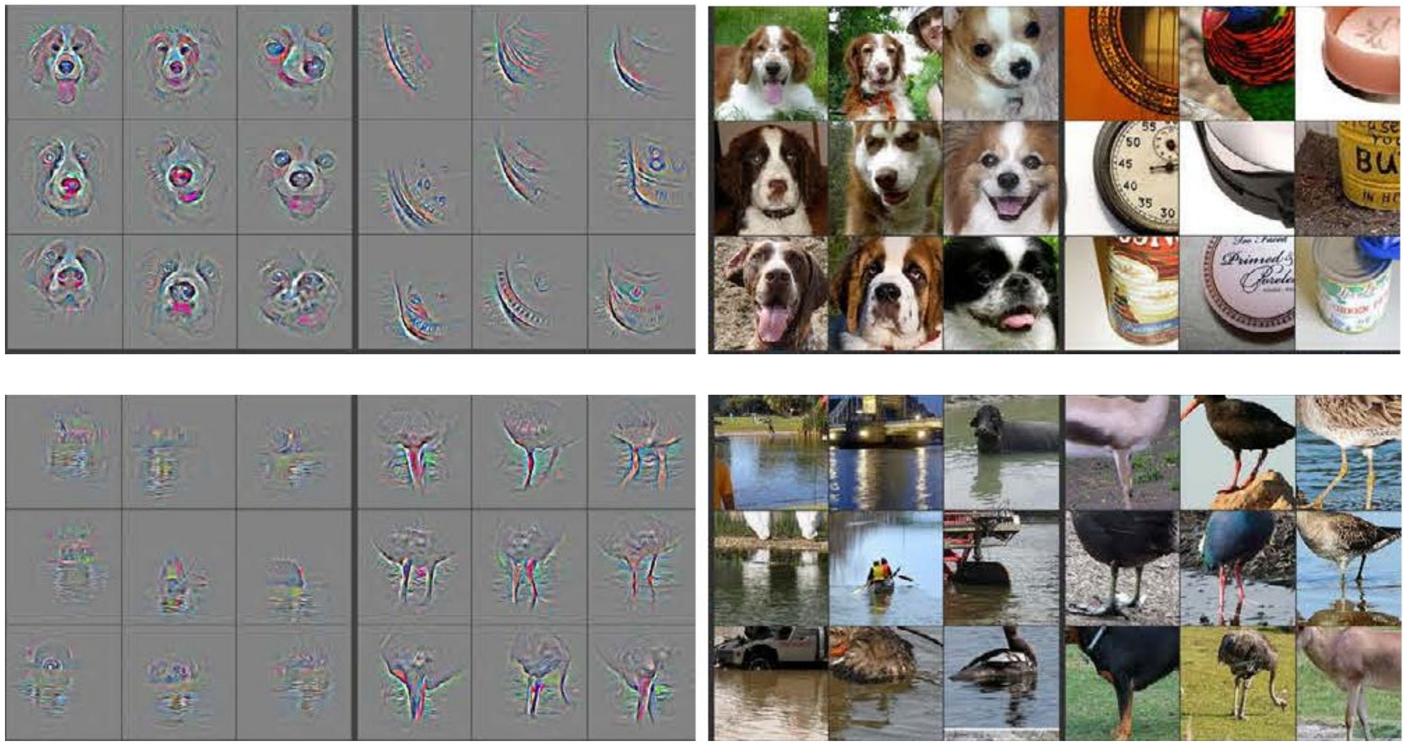
Filter Visualization

http://geometry.cs.ucl.ac.uk/creativeai

Attribution by Approximate Inversion

- Reconstruct Input from a given feature channel
- What information does the feature channel focus on?

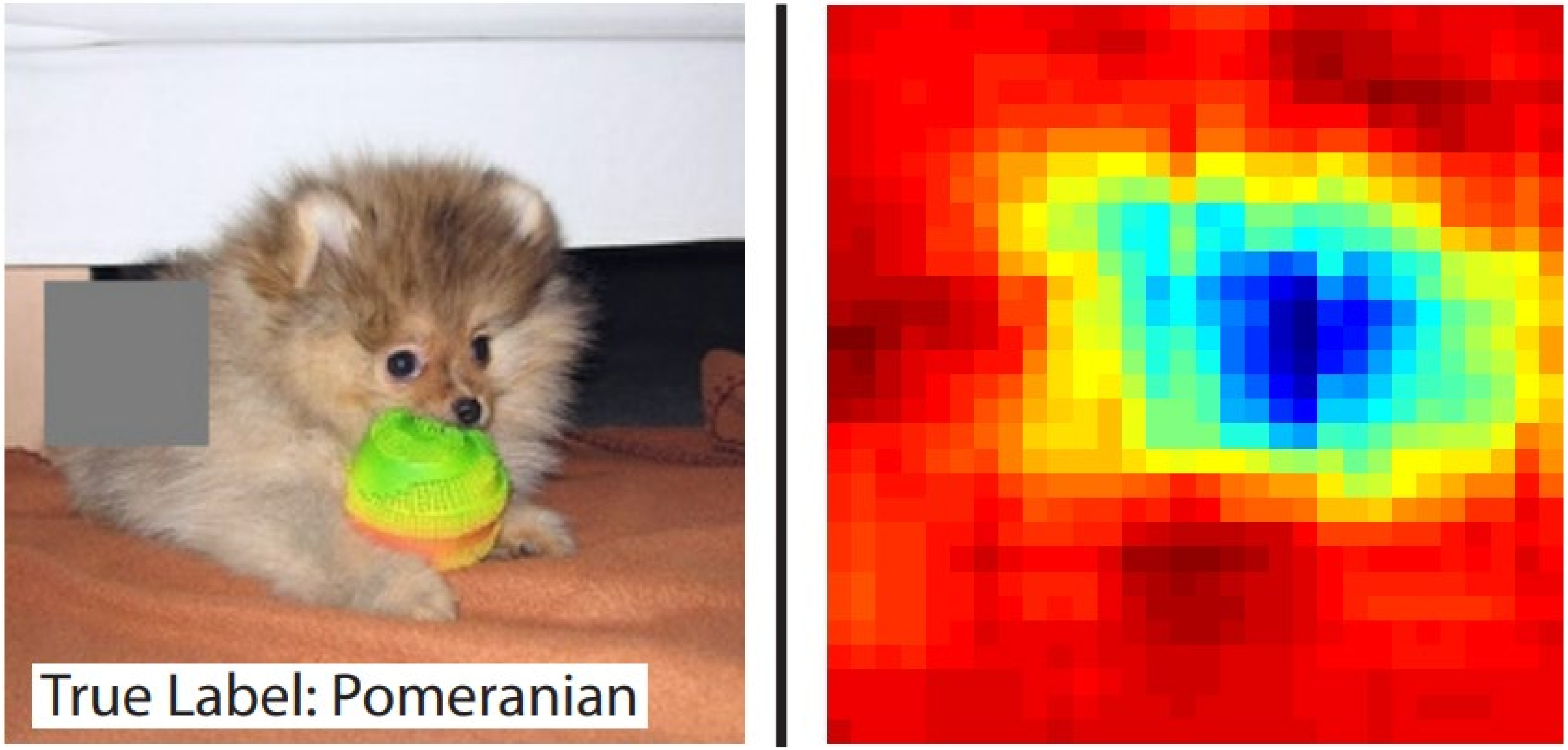




Zeiler and Fergus, *Visualizing and Understanding Convolutional Networks*, ECCV 2014

Perturbation-based Attribution

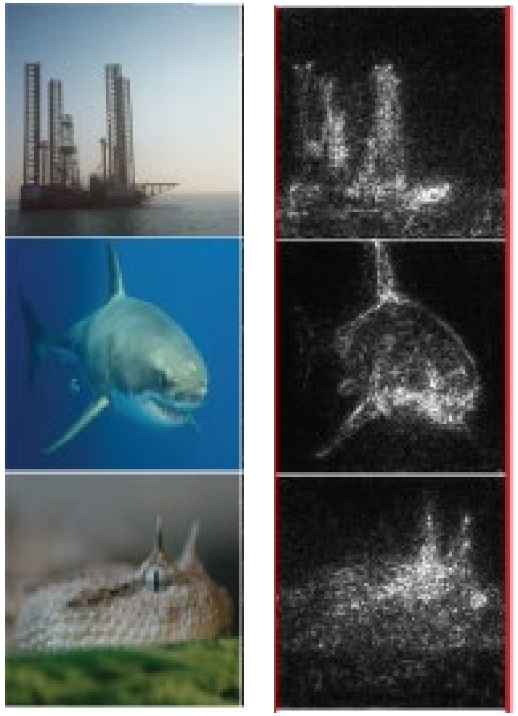
Probability for correct classification when centering the box at each pixel.

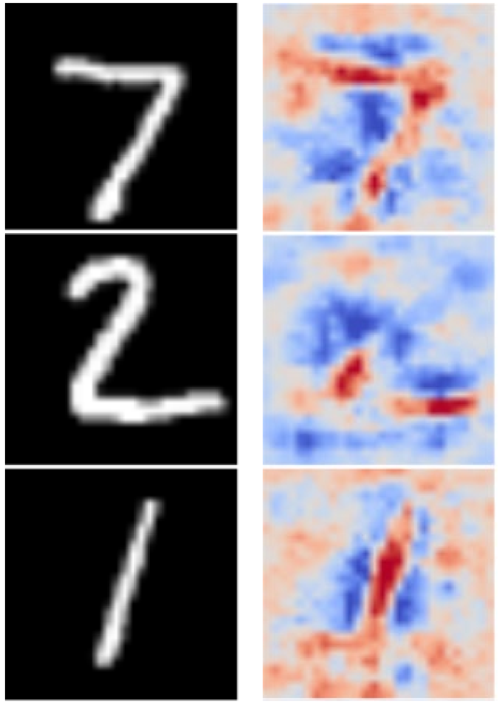


Zeiler and Fergus, *Visualizing and Understanding Convolutional Networks*, ECCV 2014

Gradient-based Attribution

- Derivative of class probability w.r.t input pixels
- Which parts of the input is the class probability sensitive to?



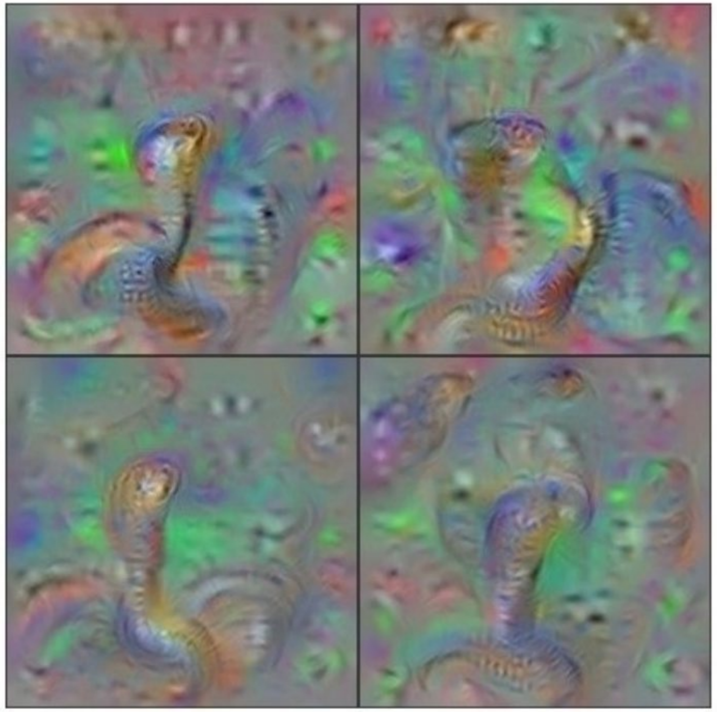


Smilkov et al., *SmoothGrad: removing noise by adding noise*, arXiv 2017

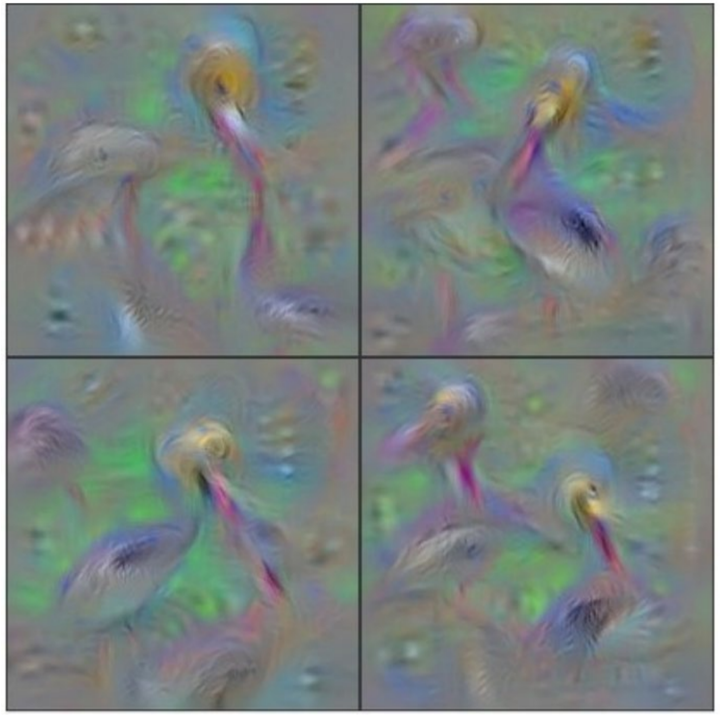
Inputs that Maximize Feature Response

Local maxima of the response for class:

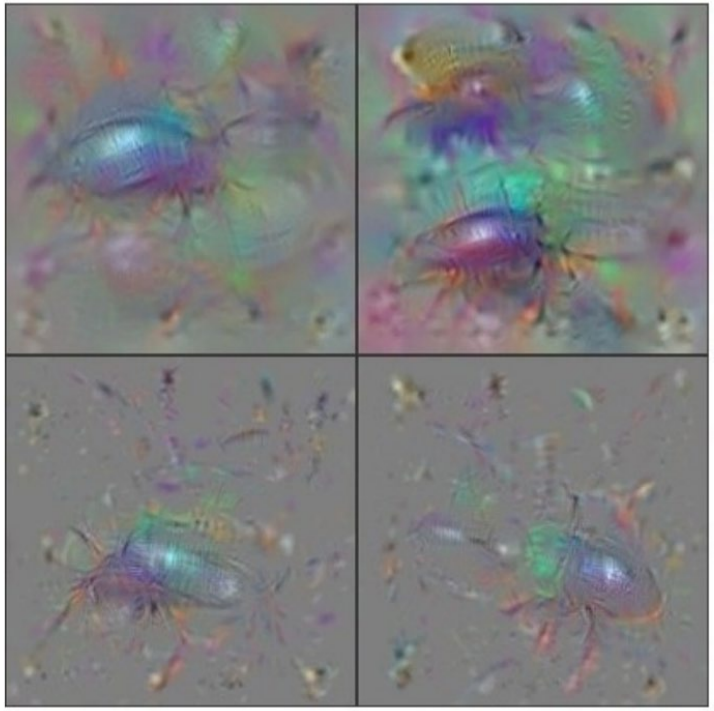
Indian Cobra



Pelican



Ground Beetle



Images from: Yosinski et al. *Understanding Neural Networks Through Deep Visualization*. ICML 2015

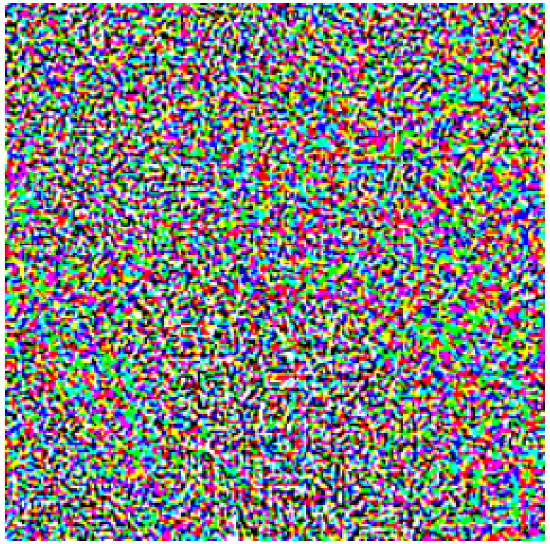
Inputs that Maximize the Error

$$\max_{\delta \in \Delta} \mathcal{L}(x + \delta, y; \theta) \qquad \Delta = \{\delta \in \mathbb{R}^d \mid \|\delta\|_p \leq \varepsilon\}$$

x

"Panda" 55.7% conf.

$+ .007 \times$



δ

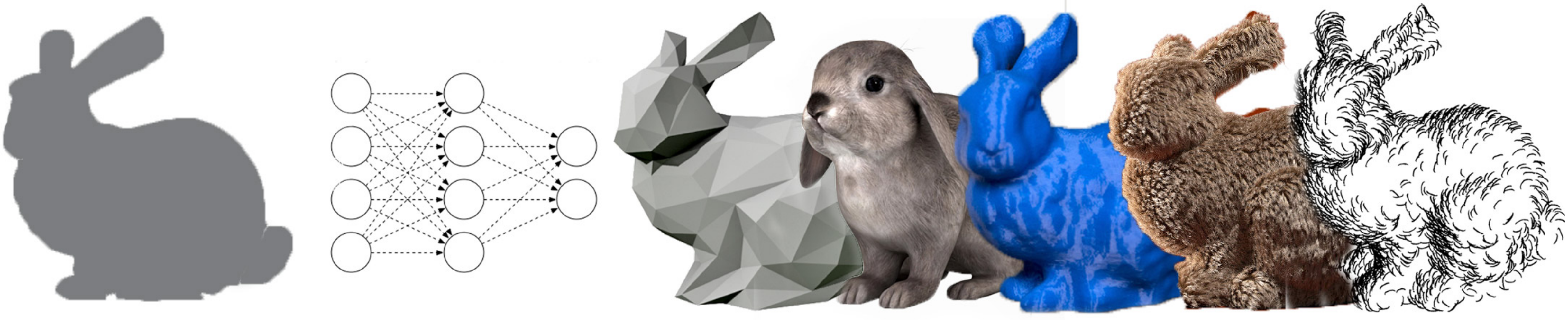
$=$

$x + \delta$

"Gibbon" 99.3% conf.

Images from: Goodfellow et al. *Explaining and Harnessing Adversarial Examples*. ICLR 2015

Course Information (slides/code/comments)



http://geometry.cs.ucl.ac.uk/creativeai/