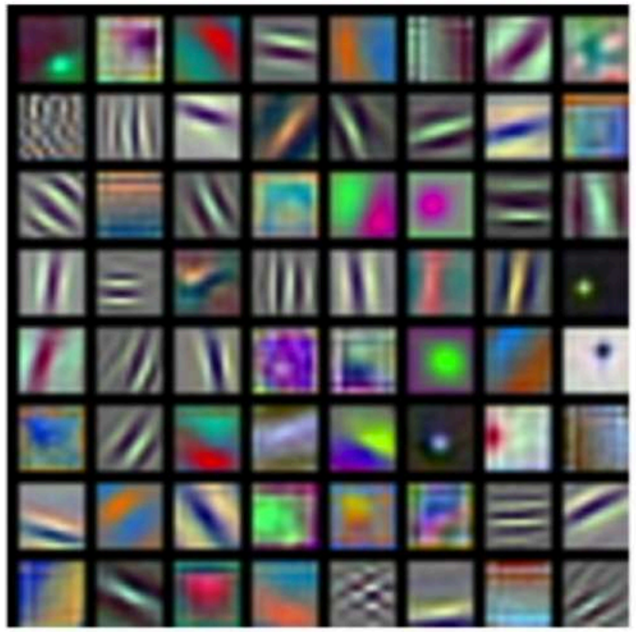
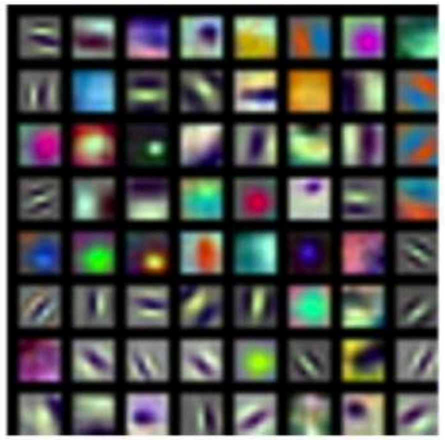
Visualize CNN

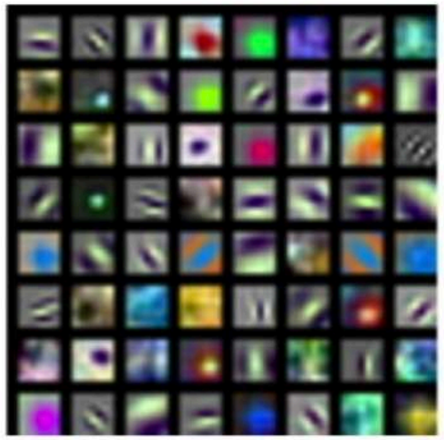
First Layer: Visualize Filters



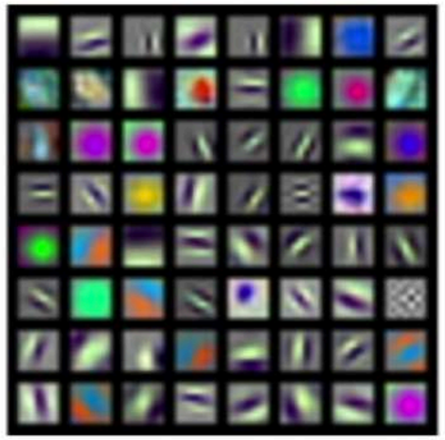
AlexNet:
64 x 3 x 11 x 11



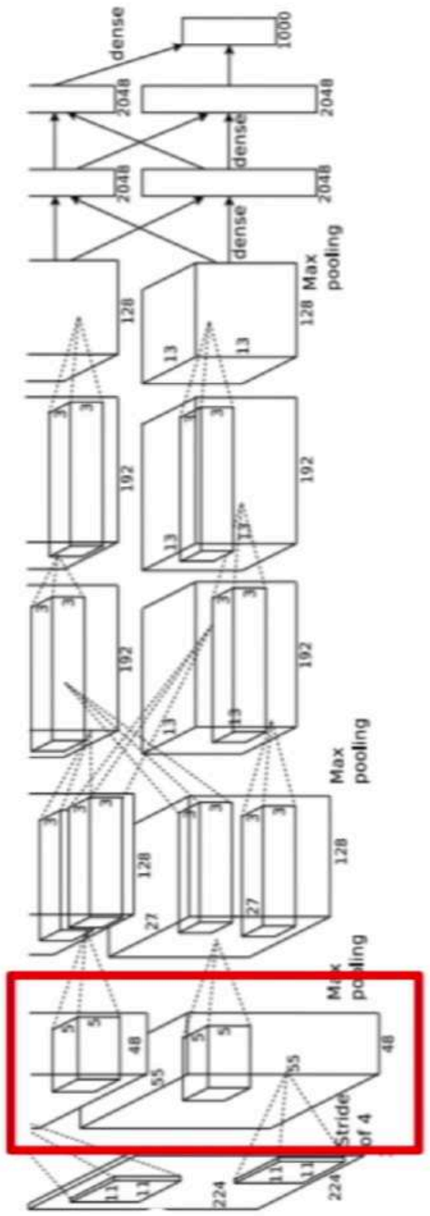
ResNet-18:
64 x 3 x 7 x 7



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

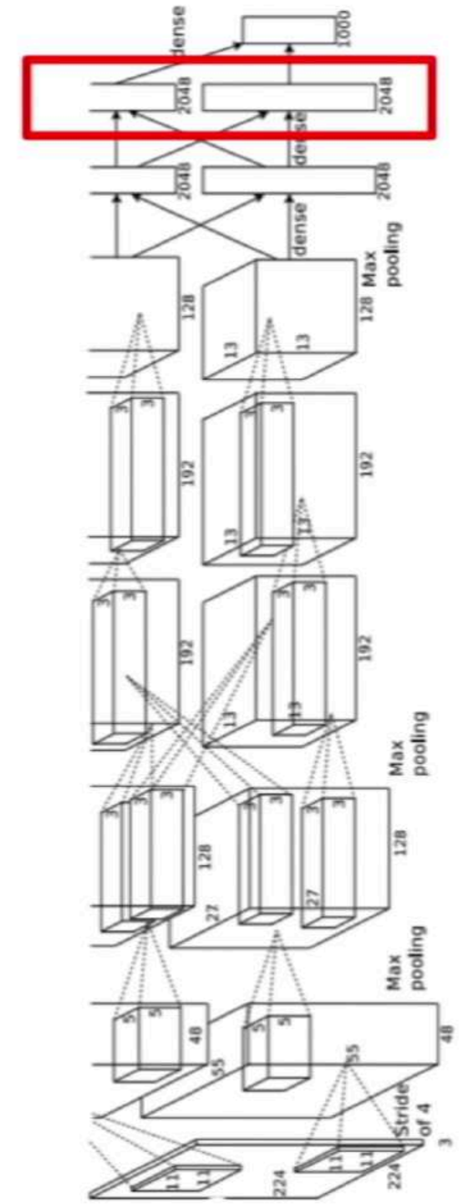
Open the Black Box

- Model explainability
  - Trustworthy
  - Accountability
  - Knowledge transfer

- When model don’t work (well enough)
  - Something wrong with the data?
  - Something wrong with the model architecture?
  - Something wrong with the code?

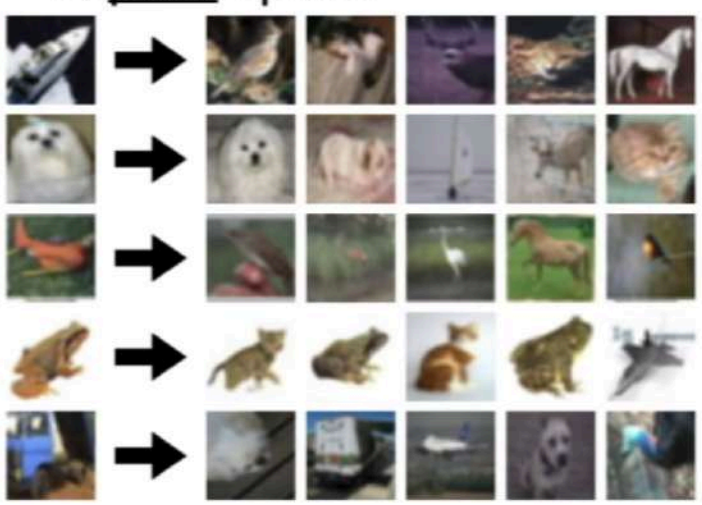
Last Layer: Nearest Neighbors

4096-dim vector



Test image   L2 Nearest neighbors in <u>feature</u> space

Recall: Nearest neighbors in <u>pixel</u> space



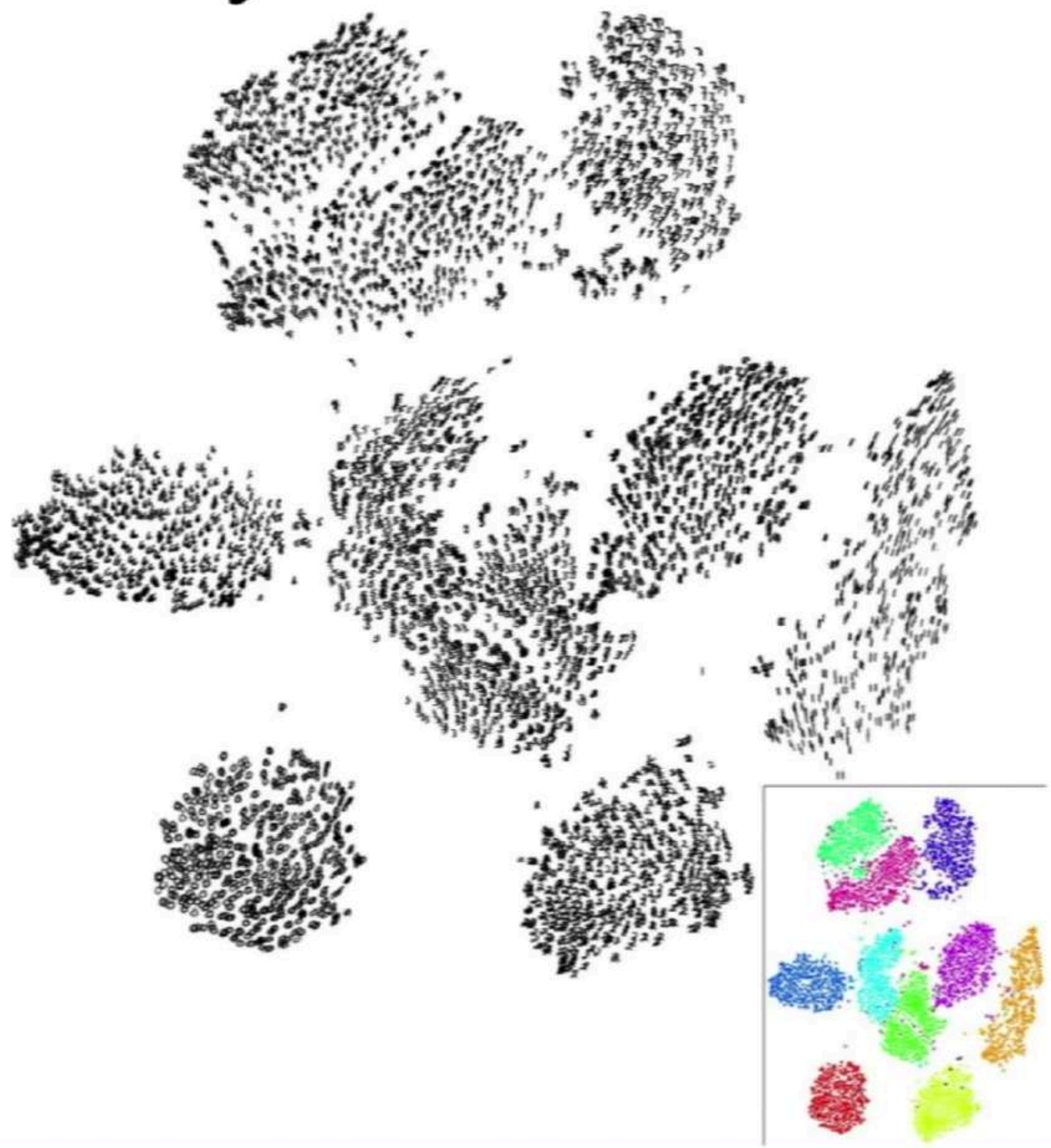
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figures reproduced with permission.

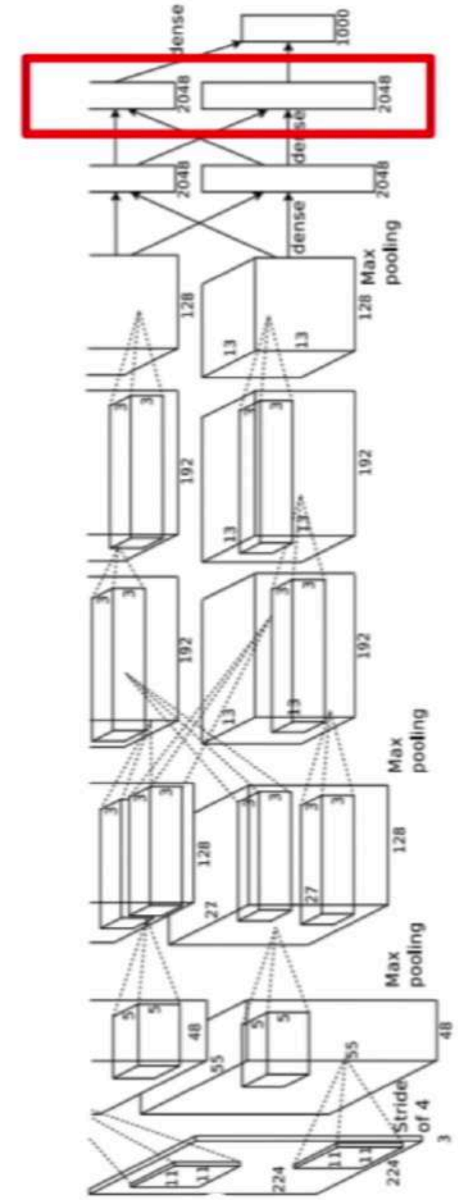
Last Layer: Dimensionality Reduction

Visualize the "space" of FC7 feature vectors by reducing dimensionality of vectors from 4096 to 2 dimensions

Simple algorithm: Principle Component Analysis (PCA)

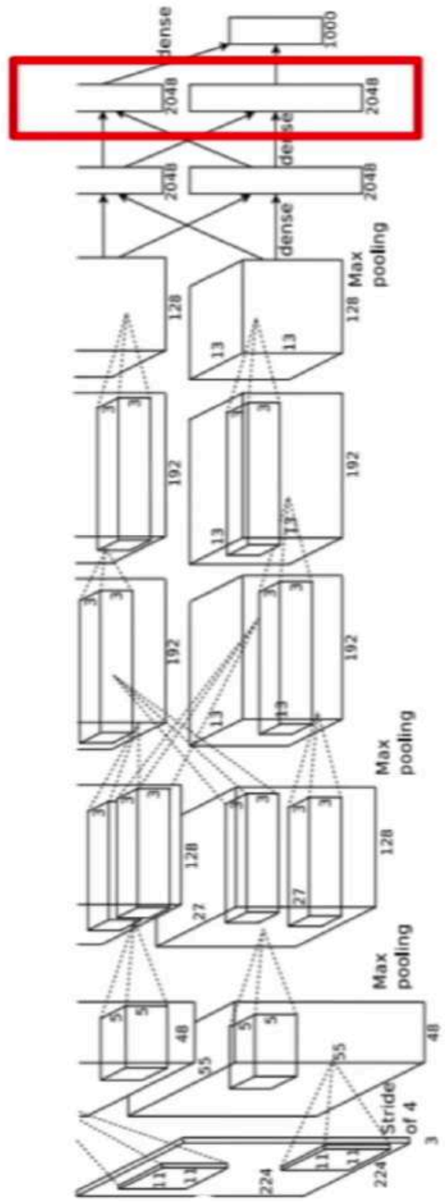
More complex: **t-SNE**





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

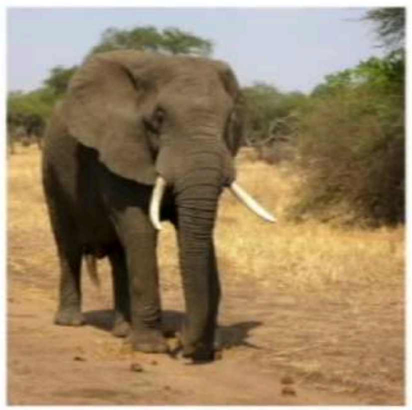
Last Layer: Dimensionality Reduction

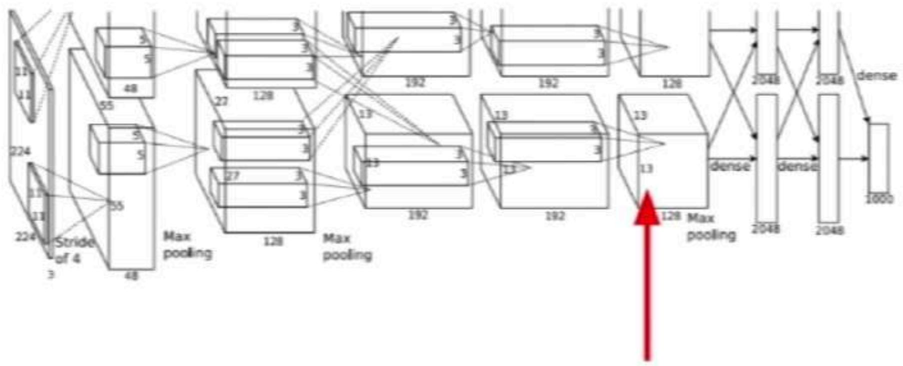


Van der Maaten and Hinton, "Visualizing Data using t-SNE", JMLR 2008
Krizhevsky et al, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS 2012.
Figure reproduced with permission.

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

Maximally Activating Patches





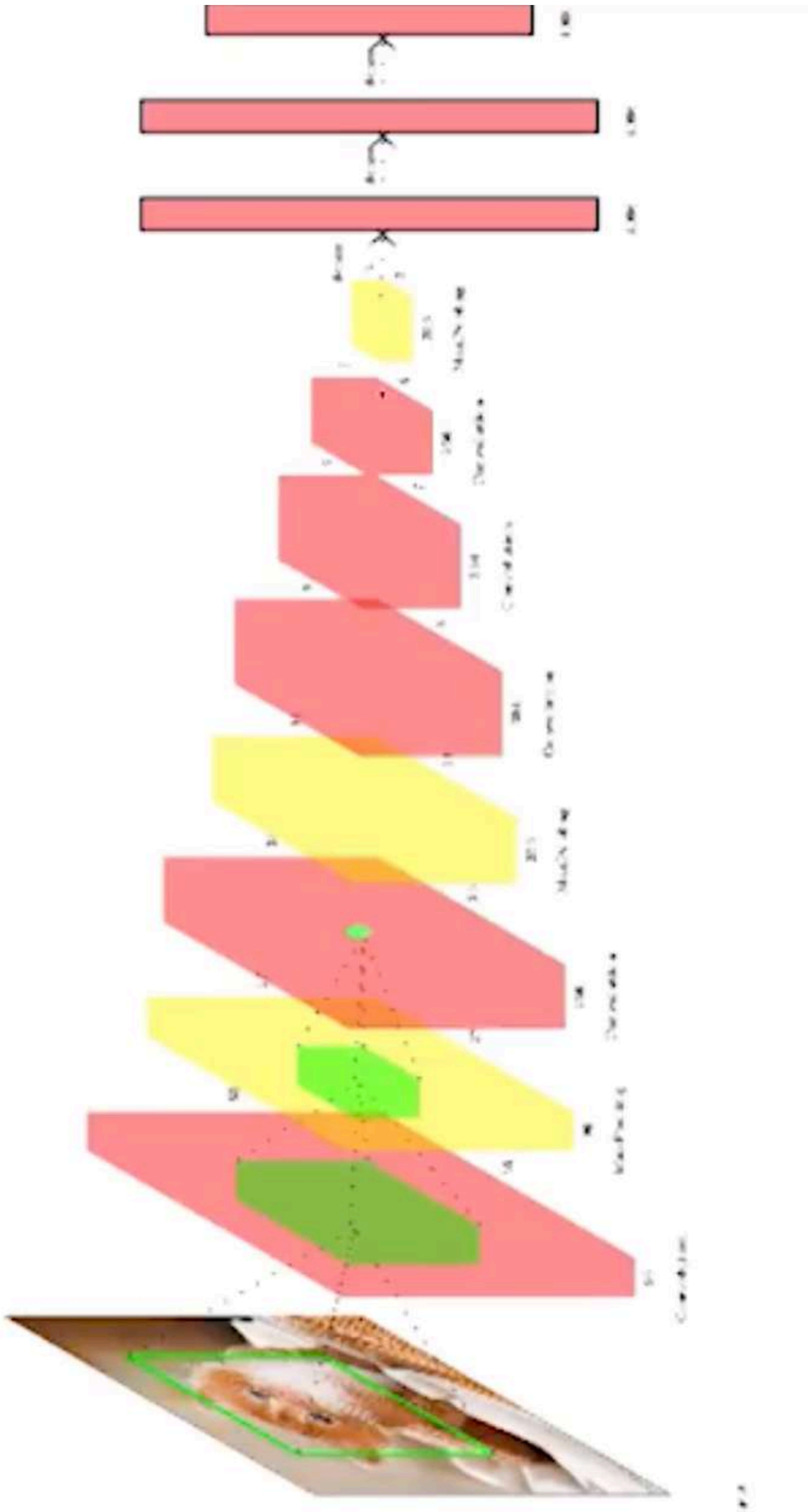
Pick a layer and a channel; e.g. conv5 is 128 x 13 x 13, pick channel 17/128

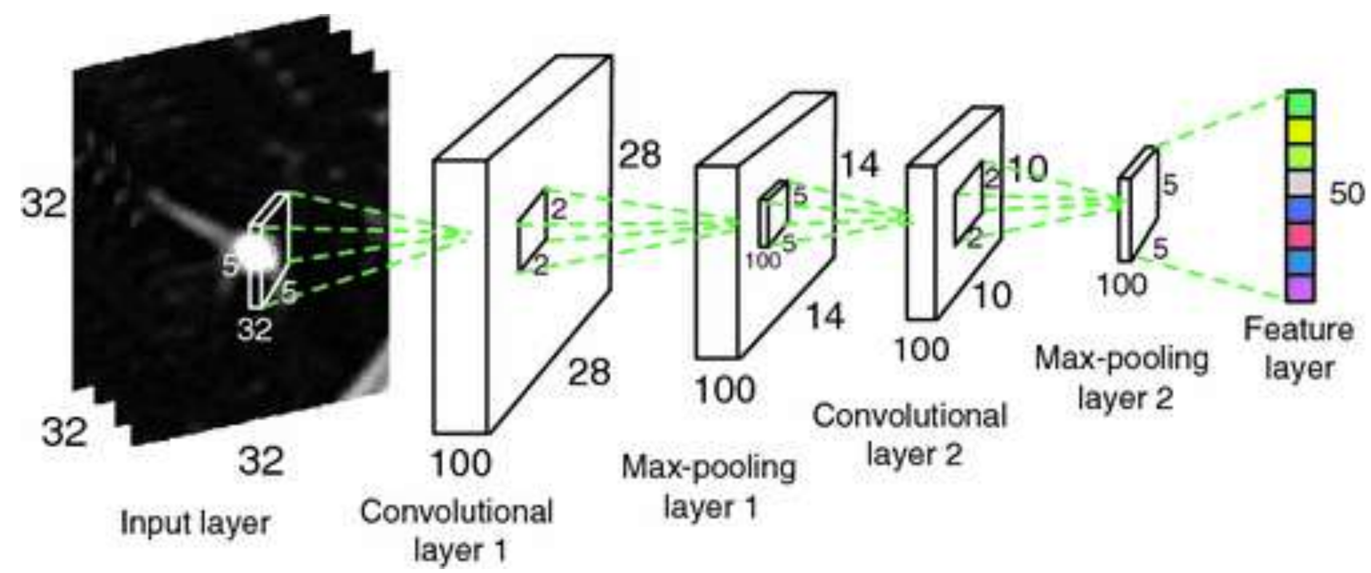
Run many images through the network, record values of chosen channel

Visualize image patches that correspond to maximal activations

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

Maximally Activating Patches





Maximally Activating Patches

1. Take 1 neuron in any activation map in any layer

2. Feed images to the CNN and record the activation value

3. Trace back to identify which patch in which image maximally activates this neuron

Maximally Activating Patches

1. Take 1 neuron in any activation map in any layer

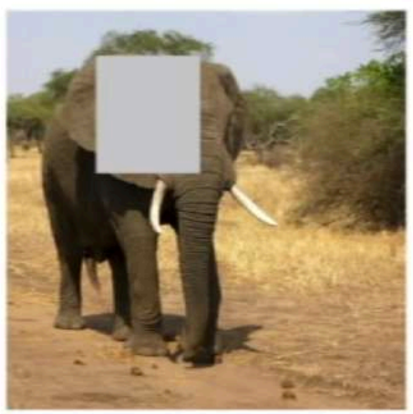
2. Feed images to the CNN and record the activation value

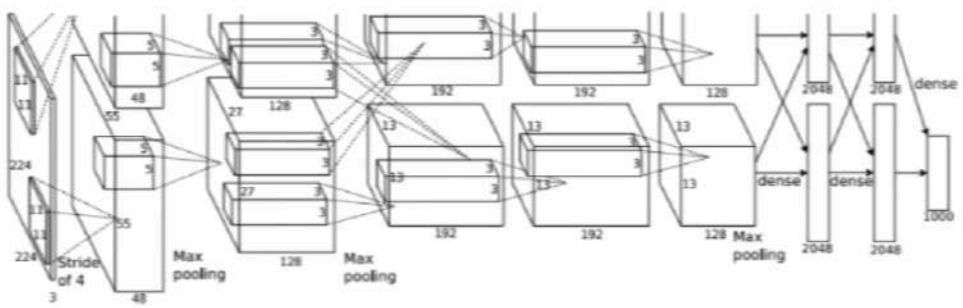
3. Trace back to identify which patch in which image maximally activates this neuron

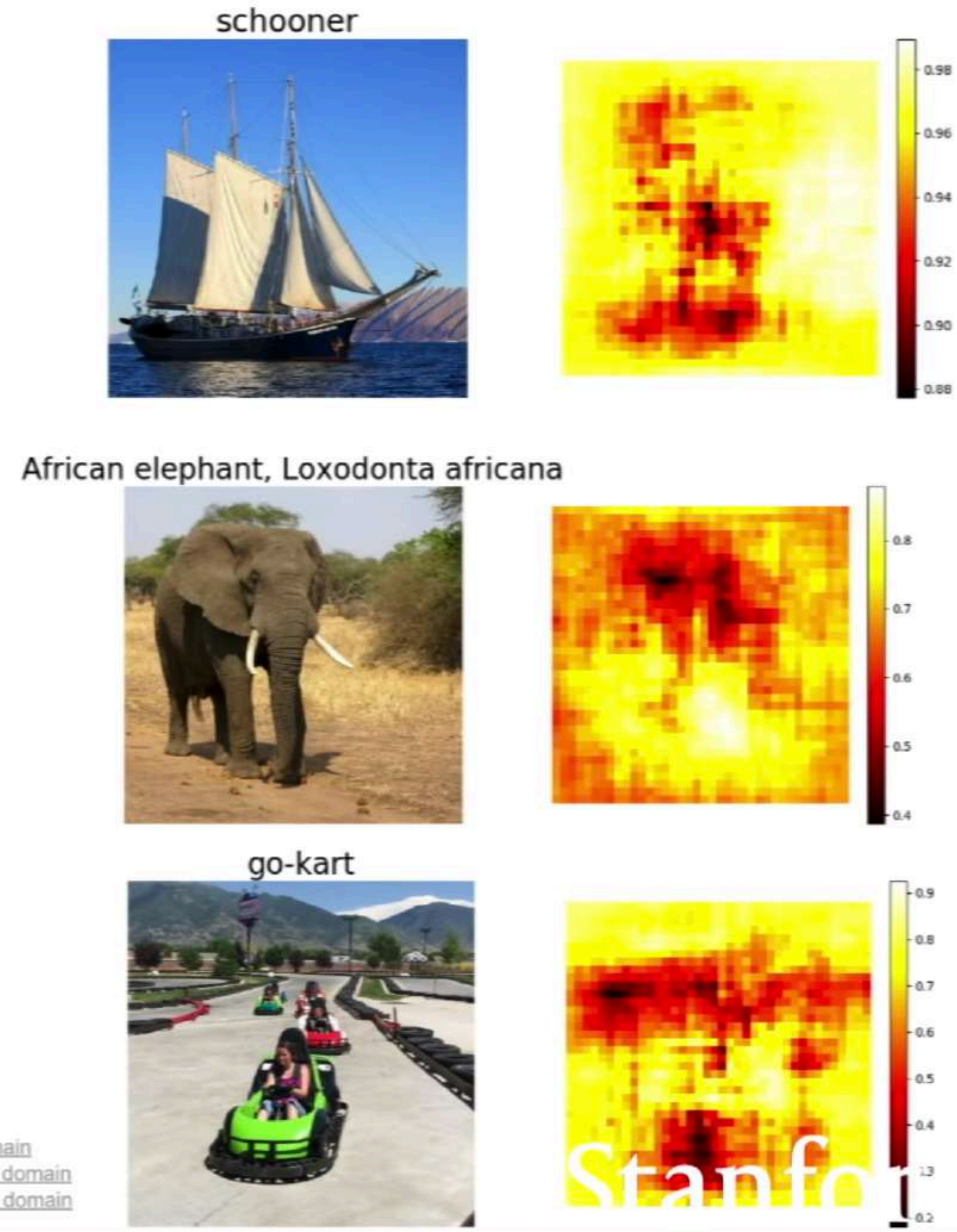
- **Limitations:**
  - **Still guesswork due to complex image content**

Occlusion Experiments

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

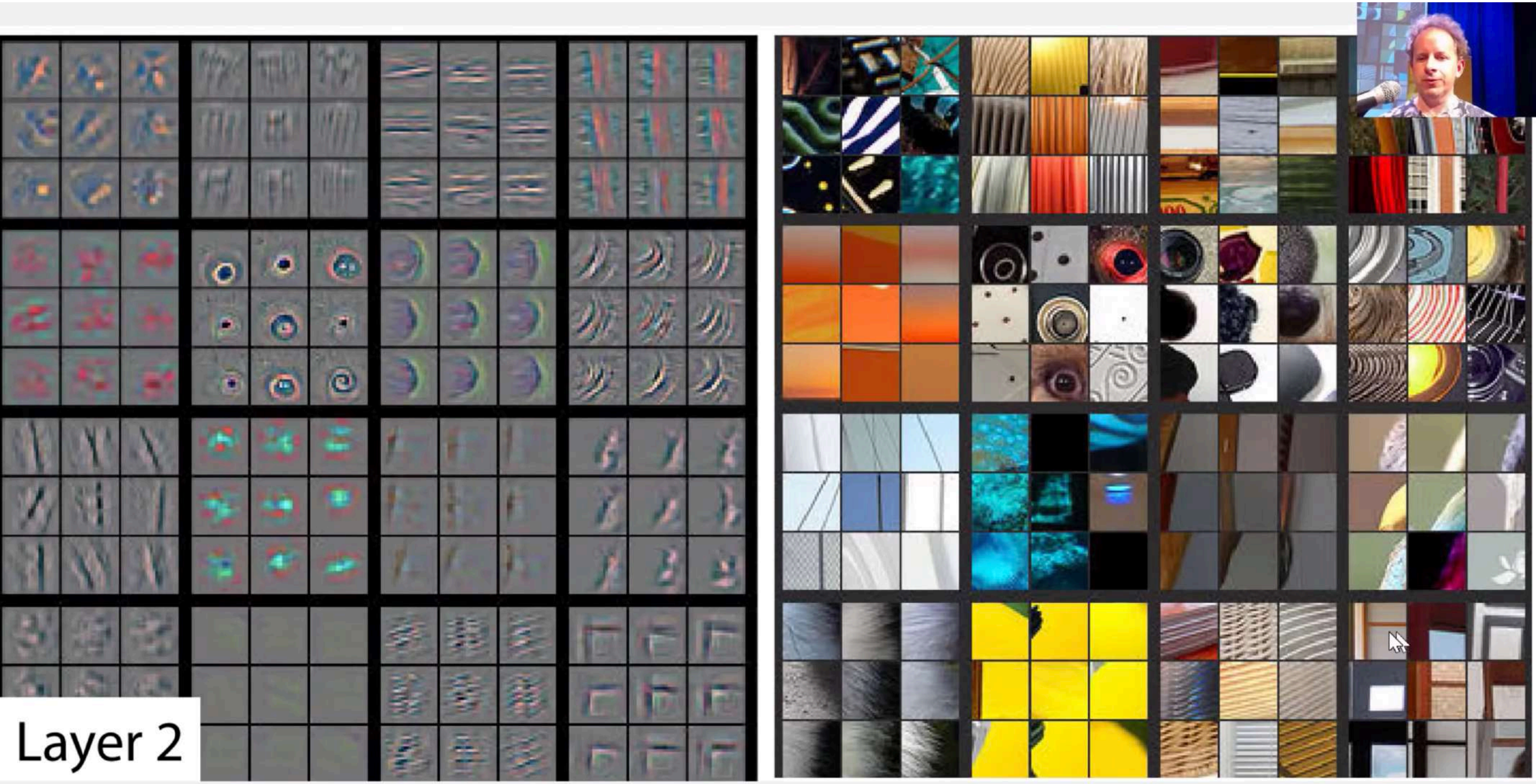




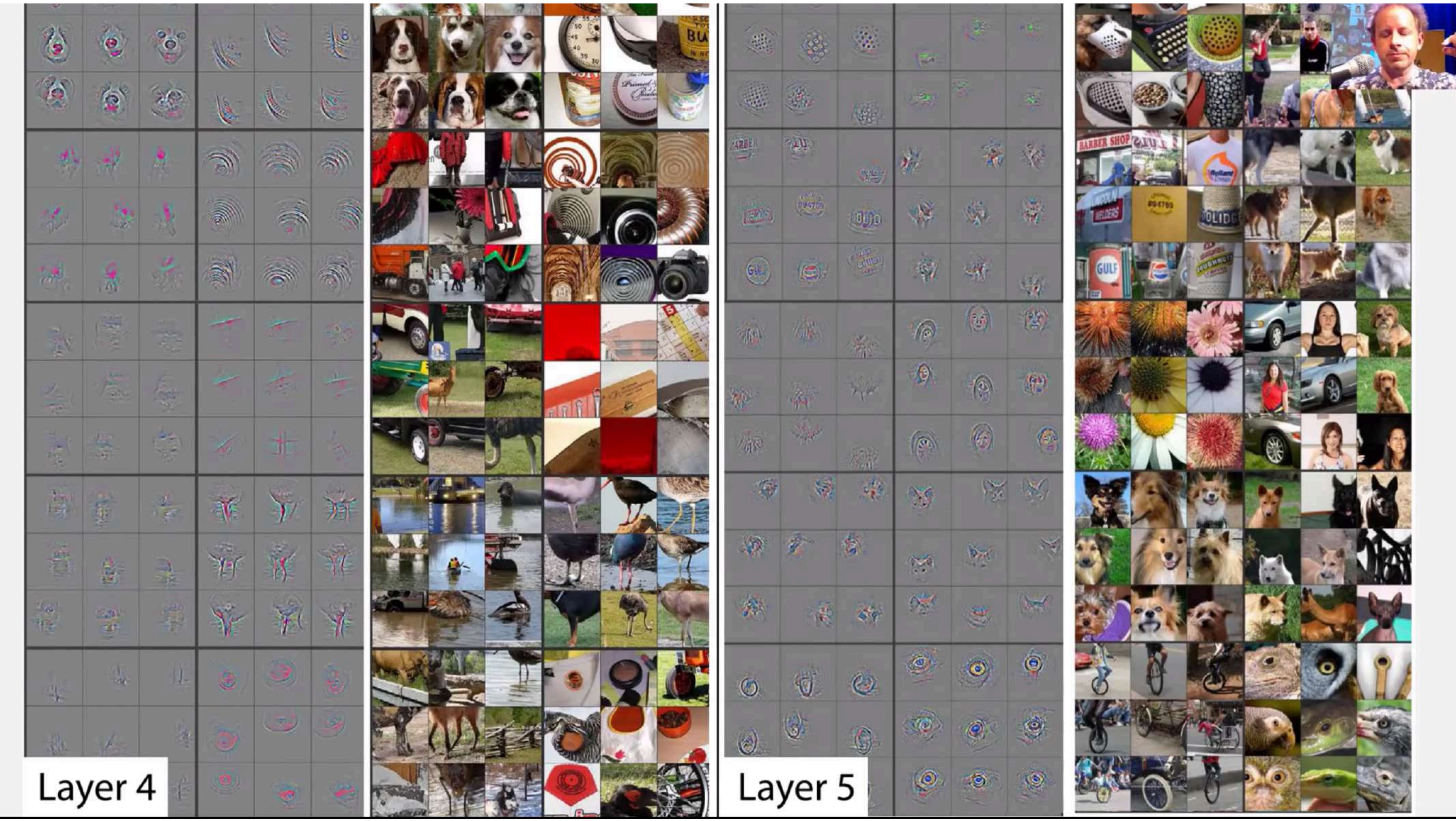


Zeiler and Fergus, "Visualizing and Understanding Convolutional Networks", ECCV 2014

Intermediate layers



Intermediate layers



Guided Backpropagation

Visualizing CNN features: Gradient Ascent

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

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

$$I^* = \arg\max_I \boxed{f(I)} + \boxed{R(I)}$$

Neuron value

Natural image regularizer

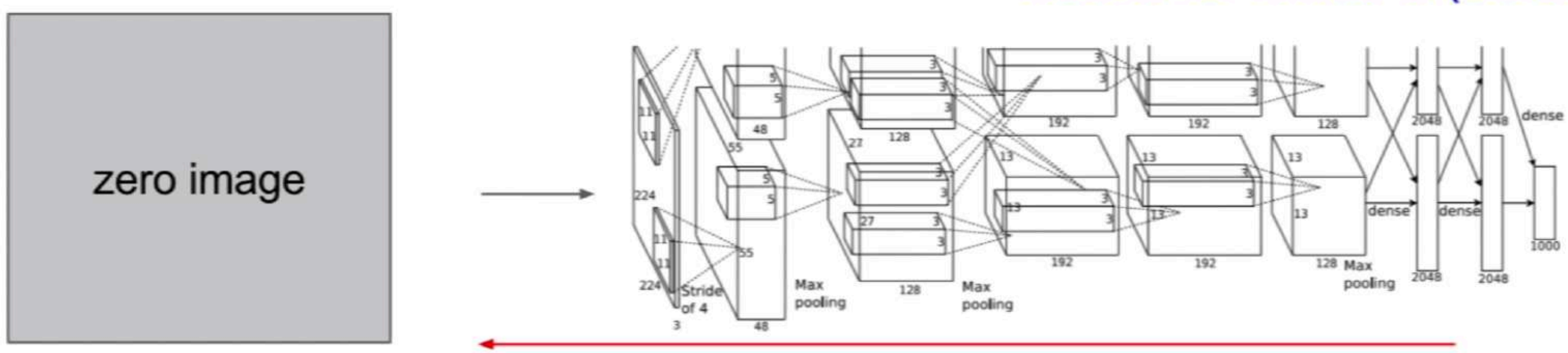
Guided Backpropagation

Visualizing CNN features: Gradient Ascent

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

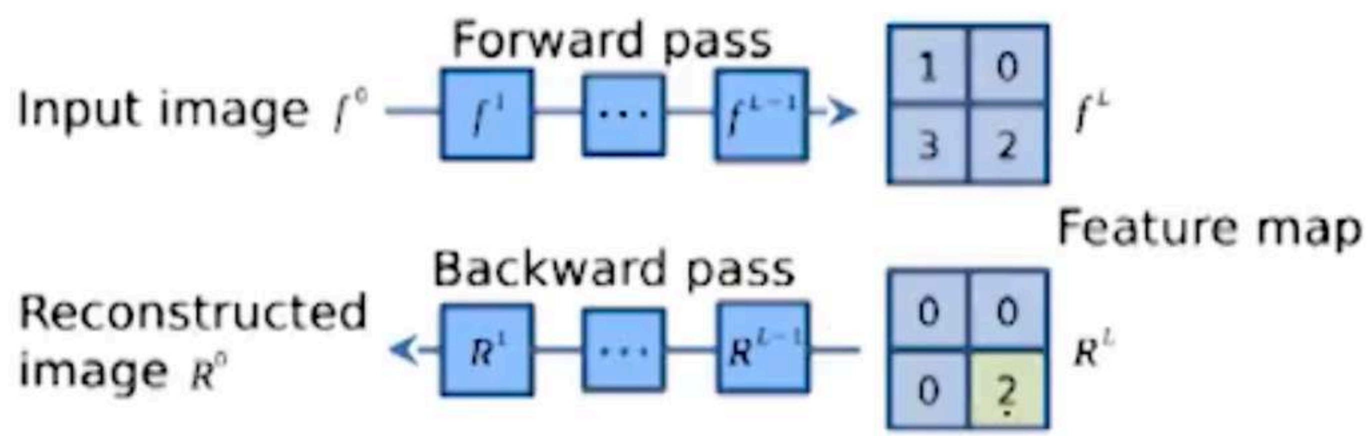
score for class c (before Softmax)

1. Initialize image to zeros



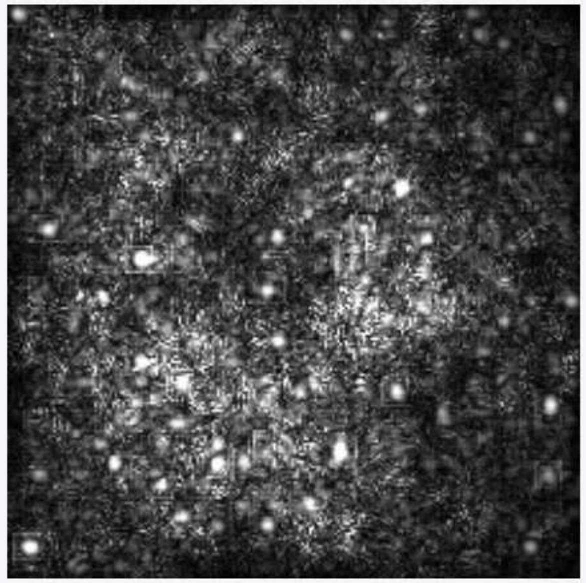
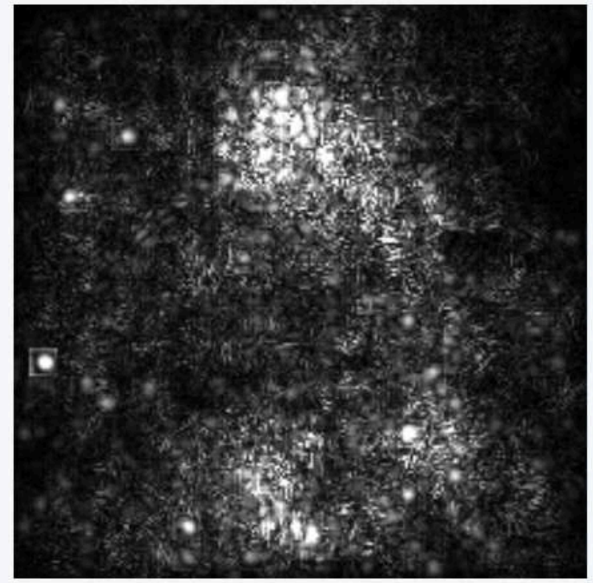
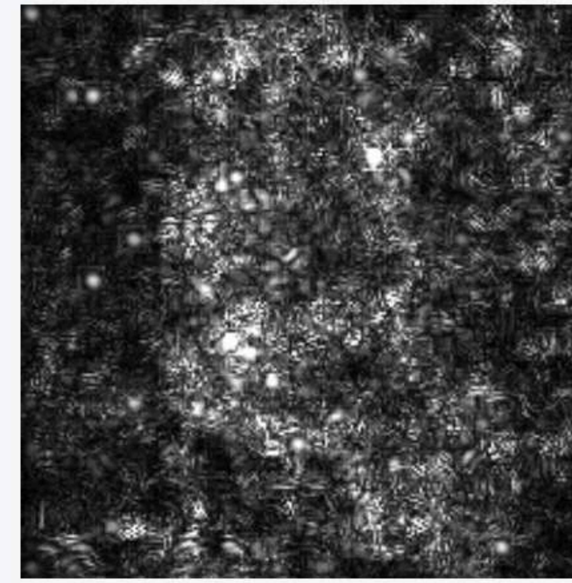
Repeat:
2. Forward image to compute current scores
3. Backprop to get gradient of neuron value with respect to image pixels
4. Make a small update to the image

Guided Backpropagation



1. Same model architecture with fixed weights and bias
2. Use weights as features => generate image
3. Loss function: maximise the activation value

- Tricks:
  - Regularization term
  - Relu to set negative activations & gradients to zero - avoid canceling out, ends up a grey image

Vanilla Backpropagation Saliency			
Guided Backpropagation Saliency (GB)