COMP0169: Machine Learning for Visual Computing

Understanding CNNs

UCL

Lectures will be Recorded

Multiple Classes & Linear Regression

C classes: one-of-c coding (or one-hot encoding)

4 classes, i-th sample is in 3rd class:

$$\mathbf{y}^i = (0, 0, 1, 0)$$

Matrix notation:

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}^1 \\ \hline \vdots \\ \hline \mathbf{y}^N \end{bmatrix} = \left[\begin{array}{c|c|c} \mathbf{y}_1 & \dots & \mathbf{y}_C \end{array} \right] \quad \text{where} \quad \mathbf{y}_c = \begin{bmatrix} y_c^1 \\ \vdots \\ y_c^N \end{bmatrix}$$

$$\mathbf{W} = \left[\begin{array}{c|c|c} \mathbf{w}_1 & \dots & \mathbf{w}_C \end{array} \right]$$

Loss function:

$$L(\mathbf{W}) = \sum_{c=1}^{C} (\mathbf{y}_c - \mathbf{X}\mathbf{w}_c)^T (\mathbf{y}_c - \mathbf{X}\mathbf{w}_c)$$

Least squares fit (decouples per class):

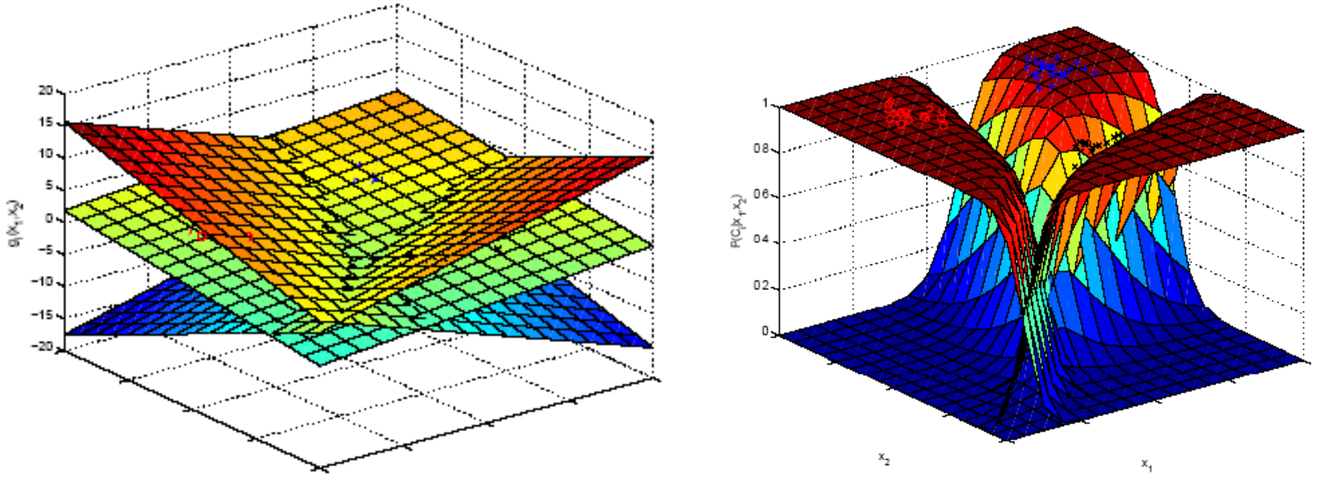
$$\mathbf{w}_c^* = (\mathbf{X}^T\mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}_c$$

Multiple Classes via SoftMax

Multiple classes & Logistic regression

Soft maximum (softmax) of competing classes:

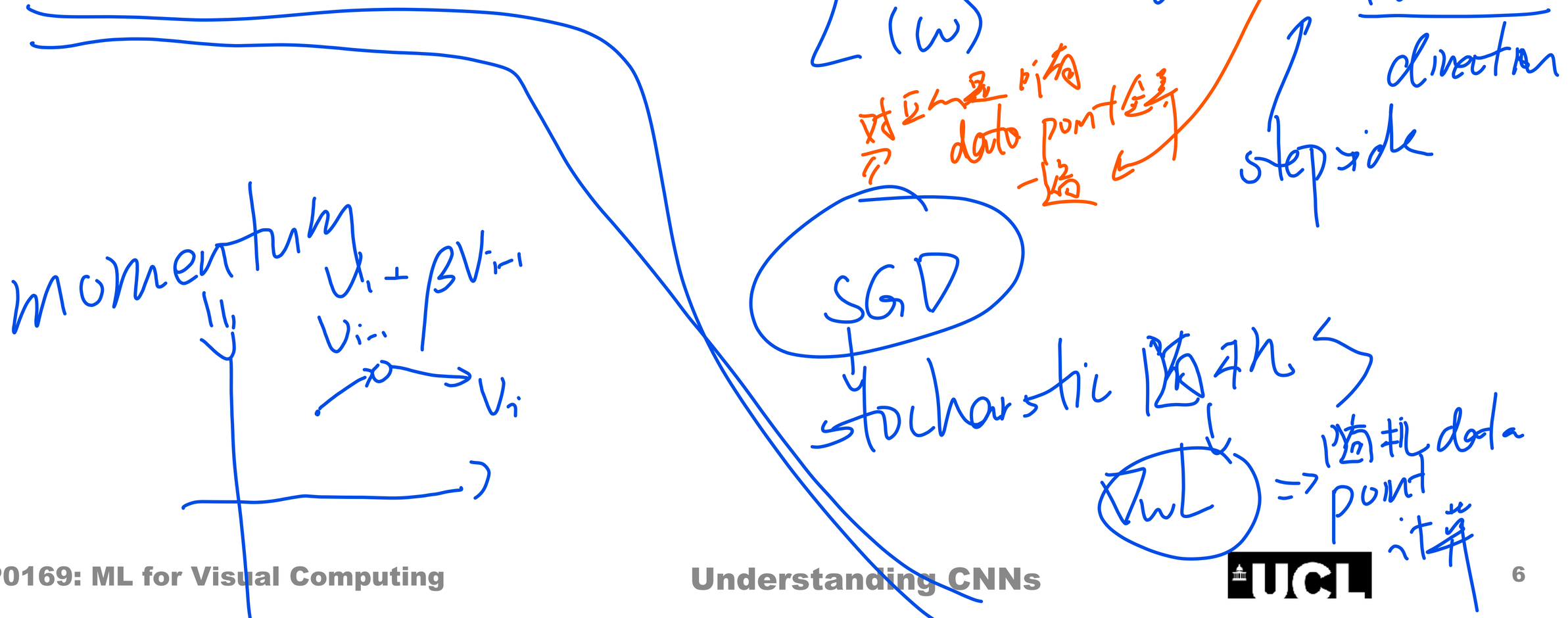
$$P(y = c|\mathbf{x}; \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x})} \doteq g_c(\mathbf{x}, \mathbf{W})$$



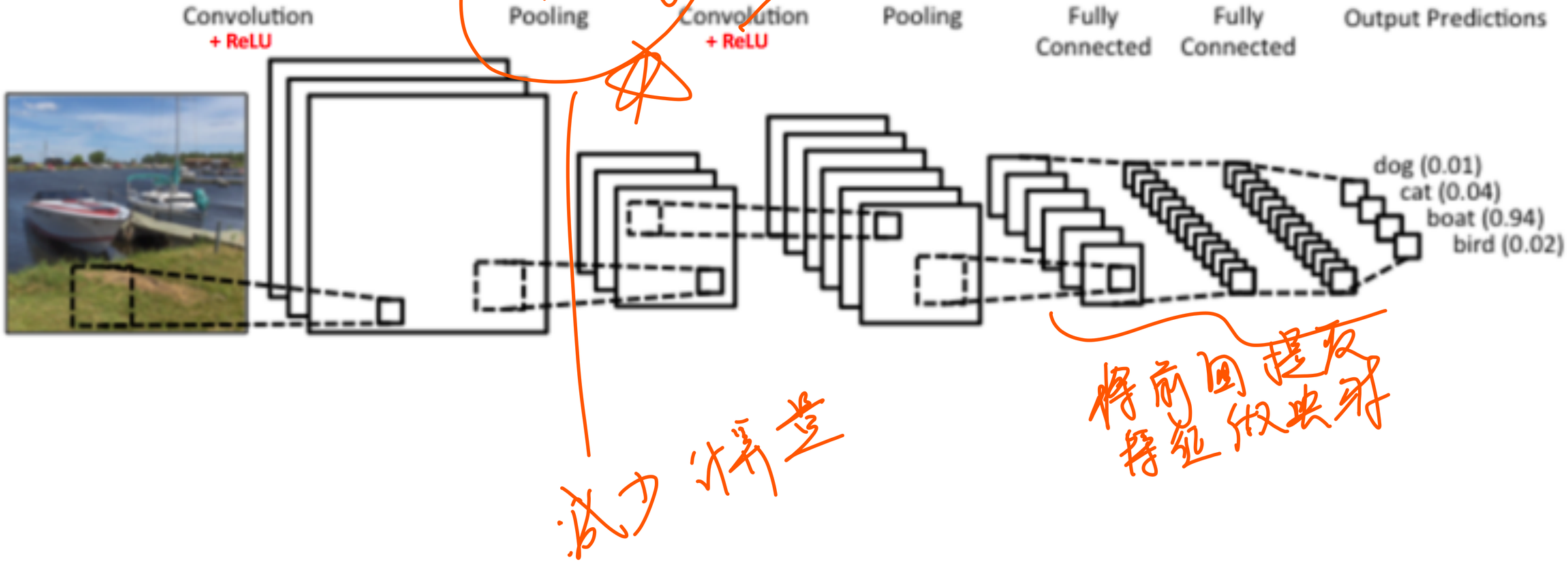
Discriminants (inputs) → Softmax (outputs)

Neural Networks (Recap)

Gradient Descent Revisited



CNN Recap


Convolution
+ ReLU
Pooling
Convolution
+ ReLU
Pooling
Fully
Connected
Fully
Connected
Output Predictions
dog (0.01)
cat (0.04)
boat (0.94)
bird (0.02)

Convolutions, Kernels, Activations, Features

Plan for Today

- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- Saliency versus Occlusion
- Guided backpropagation
- Gradient Ascent
- Feature Inversion
- DeepDreams

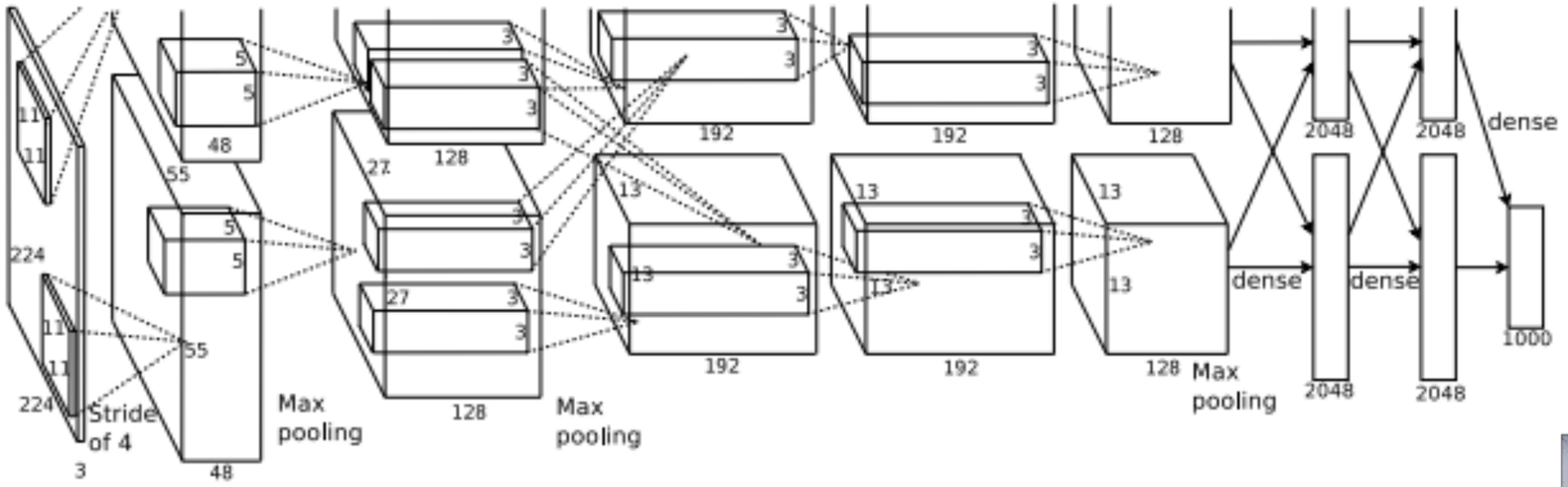
AlexNet

ImageNet Classification with Deep Convolutional Neural Networks

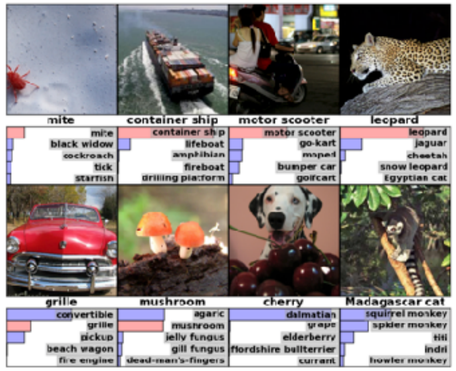
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

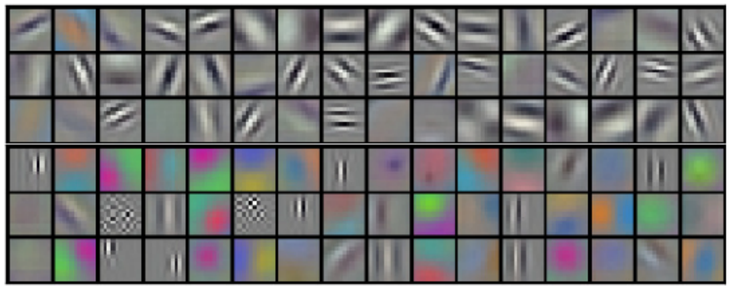
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



Class scores for 1000 Classes

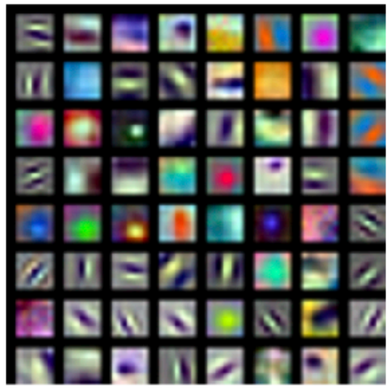


First Layer Filters



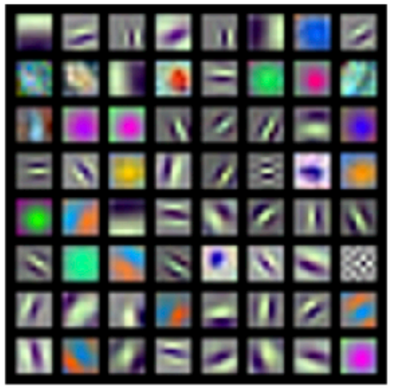
edge

AlexNet
96 x 11x11x3

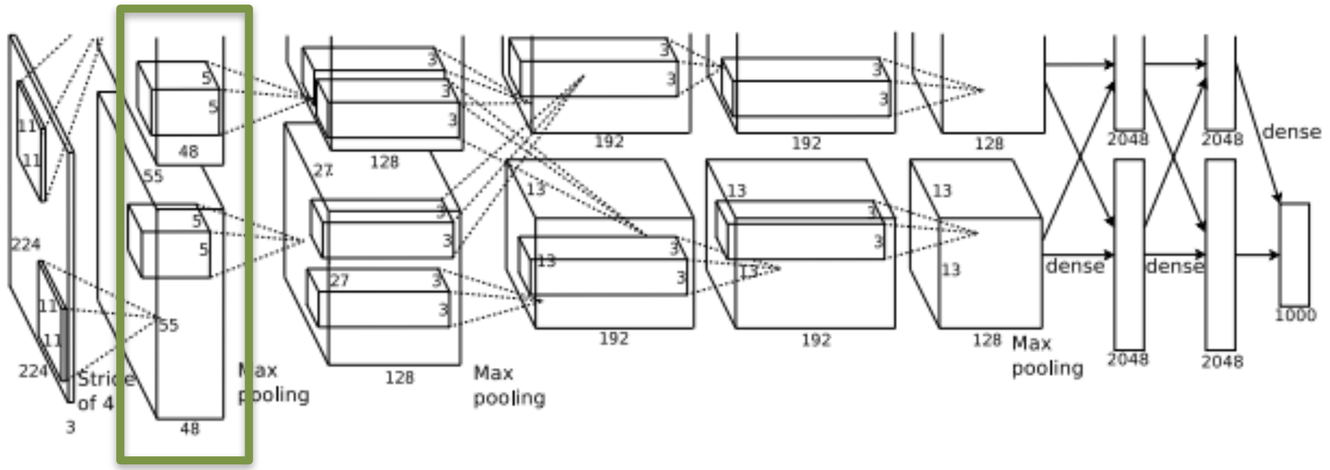


ResNet-18
64 x 7x7x3

ResNet-101
64 x 7x7x3



DenseNet-121
64 x 7x7x3



Higher Level Filters

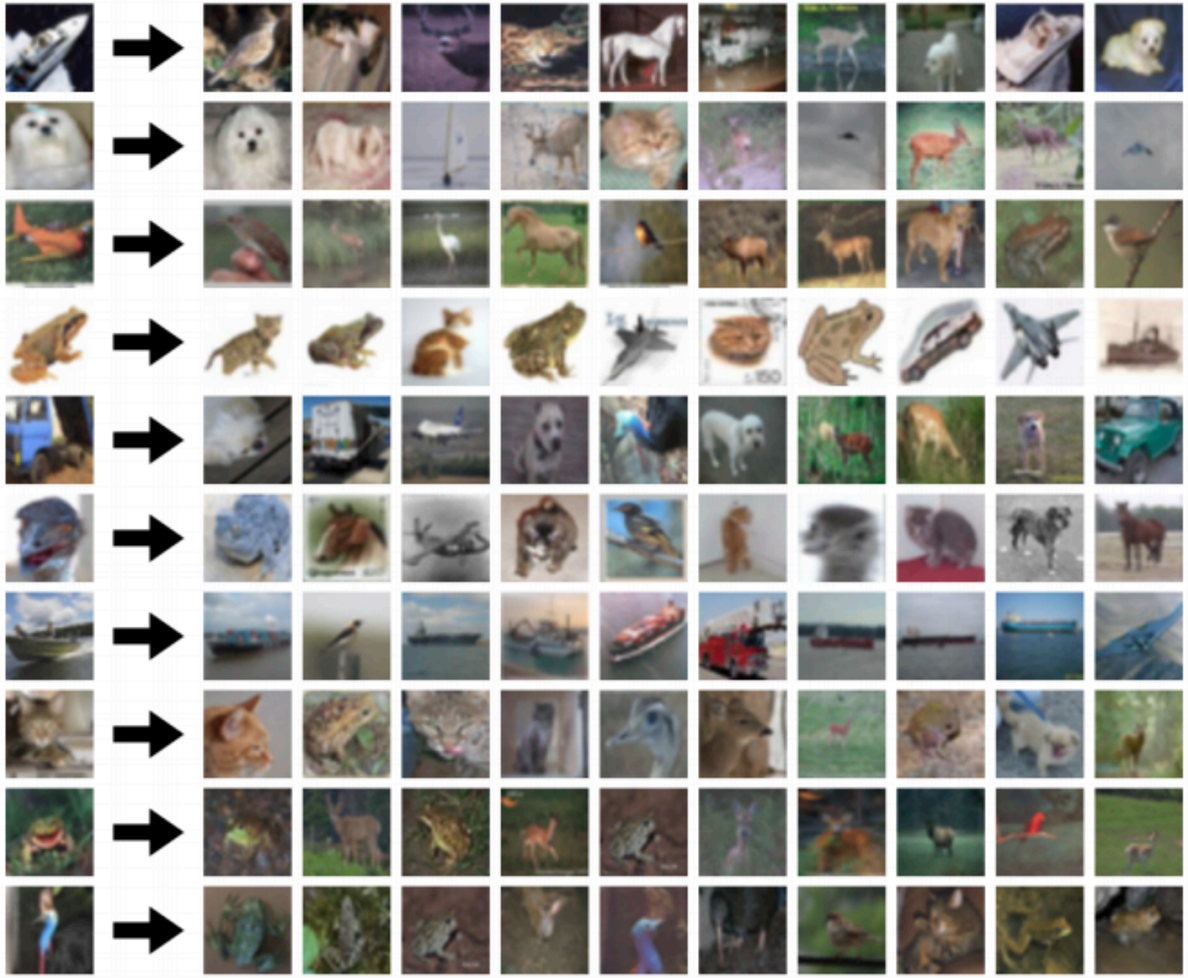
First layer weights: 16 x 3 x 7 x 7

Second layer weights:
20 x 16 x 7 x 7

Third layer weights:
20 x 20 x 7 x 7

Source: ConvNetJS CIFAR-10 example
https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

NN in Image Space

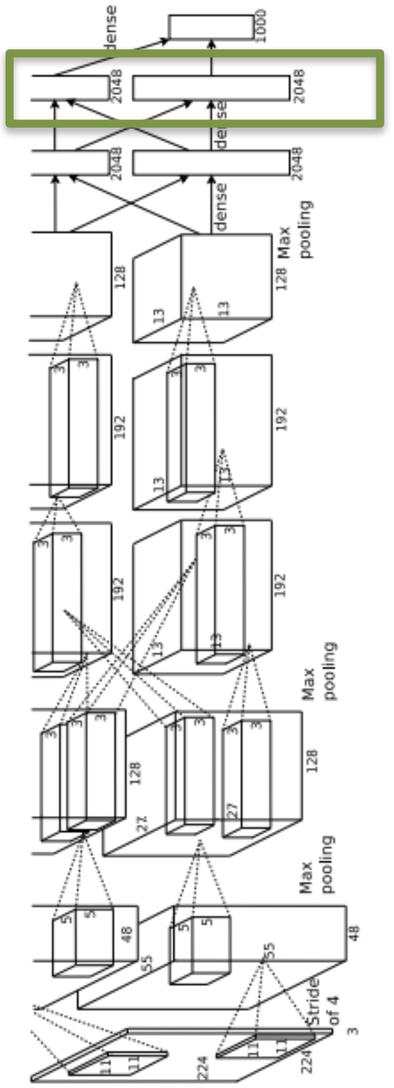


Last Layer

Feature size 2048+2048=4096.

Pretrain network. Run ‘feature extractor’ on many images.
FC7 features.

$$\mathbb{R}^{224\times224\times3} \rightarrow \mathbb{R}^{4096}$$



Nearest Neighbors in Feature Space

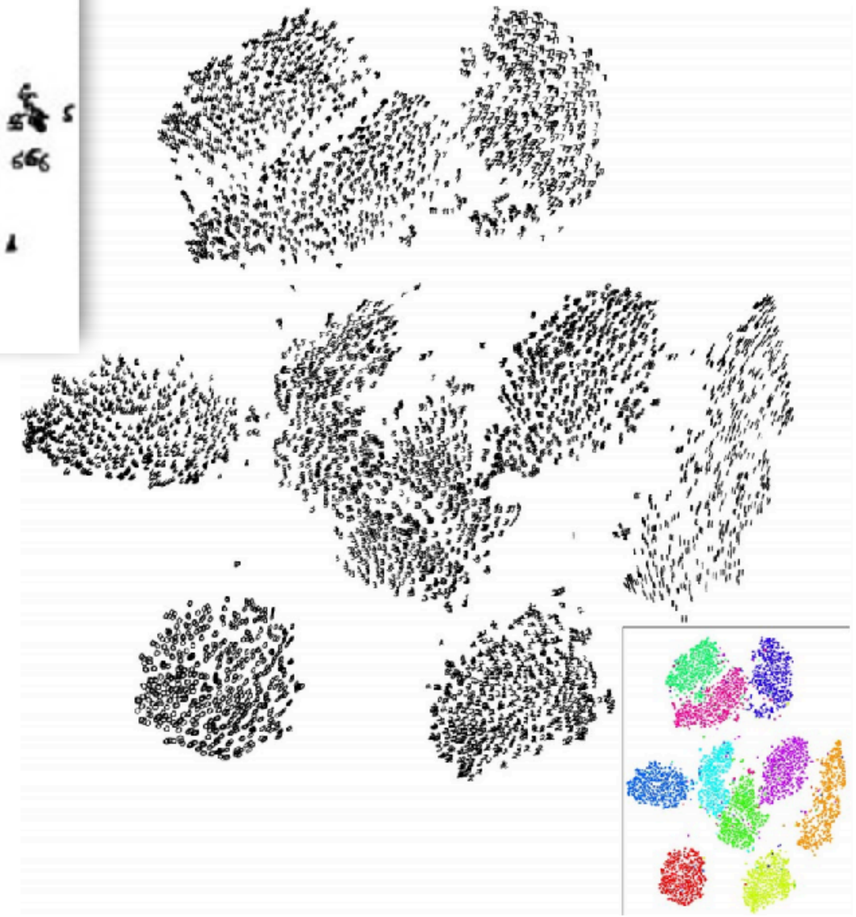
Nearest neighbors in pixel space

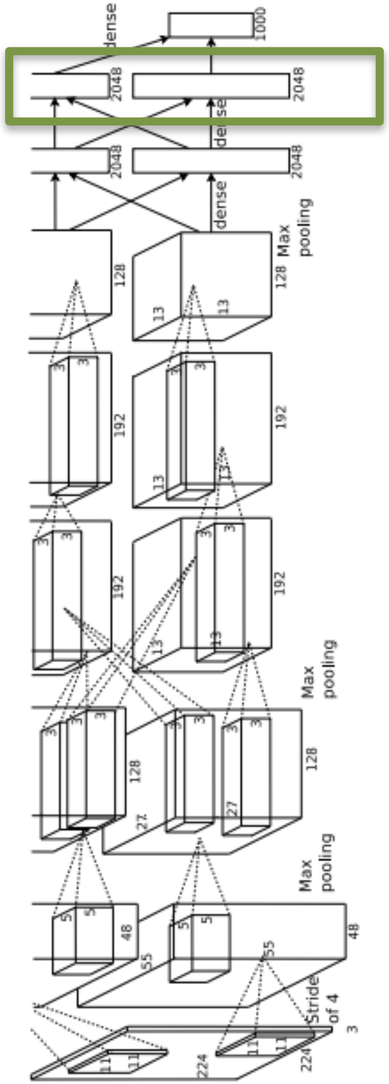
Nearest neighbors in FC7 space

Dimensionality Reduction

$$\mathbb{R}^{4096} \rightarrow \mathbb{R}^{2}$$

1. PCA
2. t-SNE





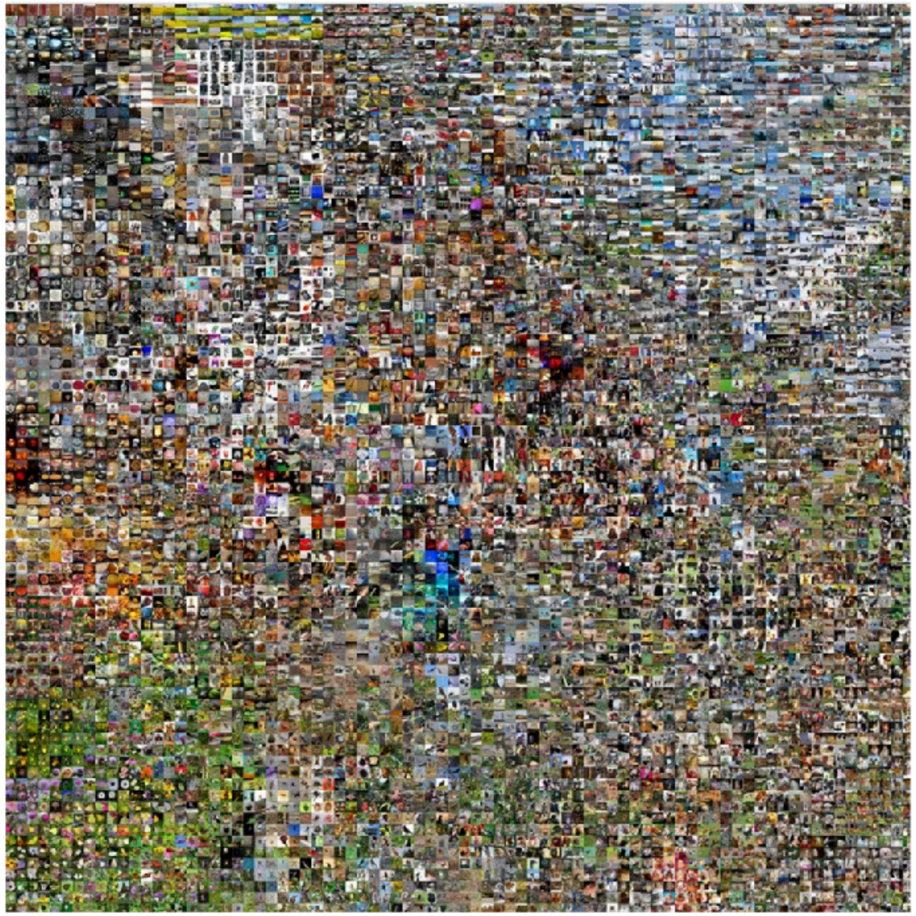
Visualizing Data using t-SNE

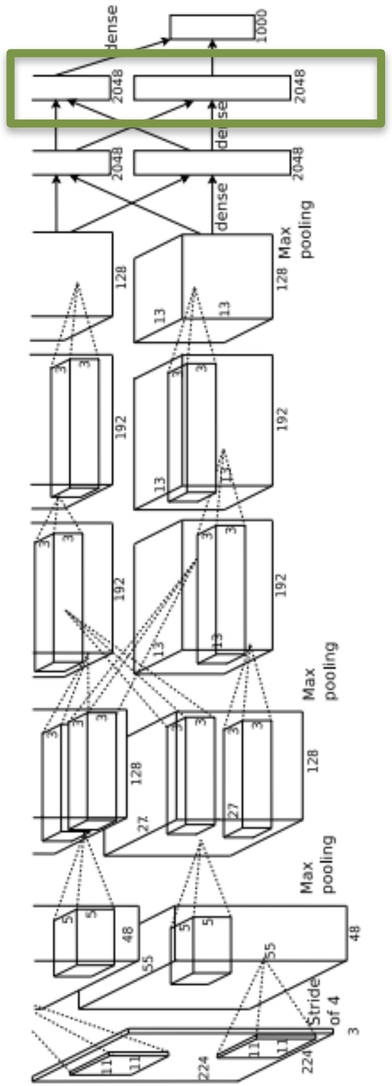
Laurens van der Maaten LVDMAATEN@GMAIL.COM
TiCC
Tilburg University
P.O. Box 90153, 5000 LE Tilburg, The Netherlands

Geoffrey Hinton HINTON@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
6 King's College Road, M5S 3G4 Toronto, ON, Canada

Editor: Yoshua Bengio

Dimensionality Reduction





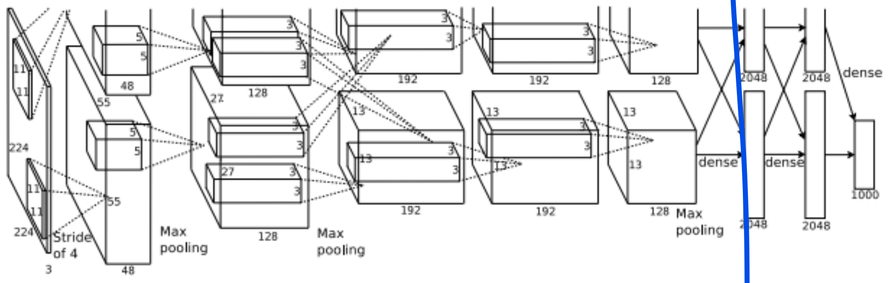
See high-resolution versions at http://cs.stanford.edu/people/karpathy/cnnembed/

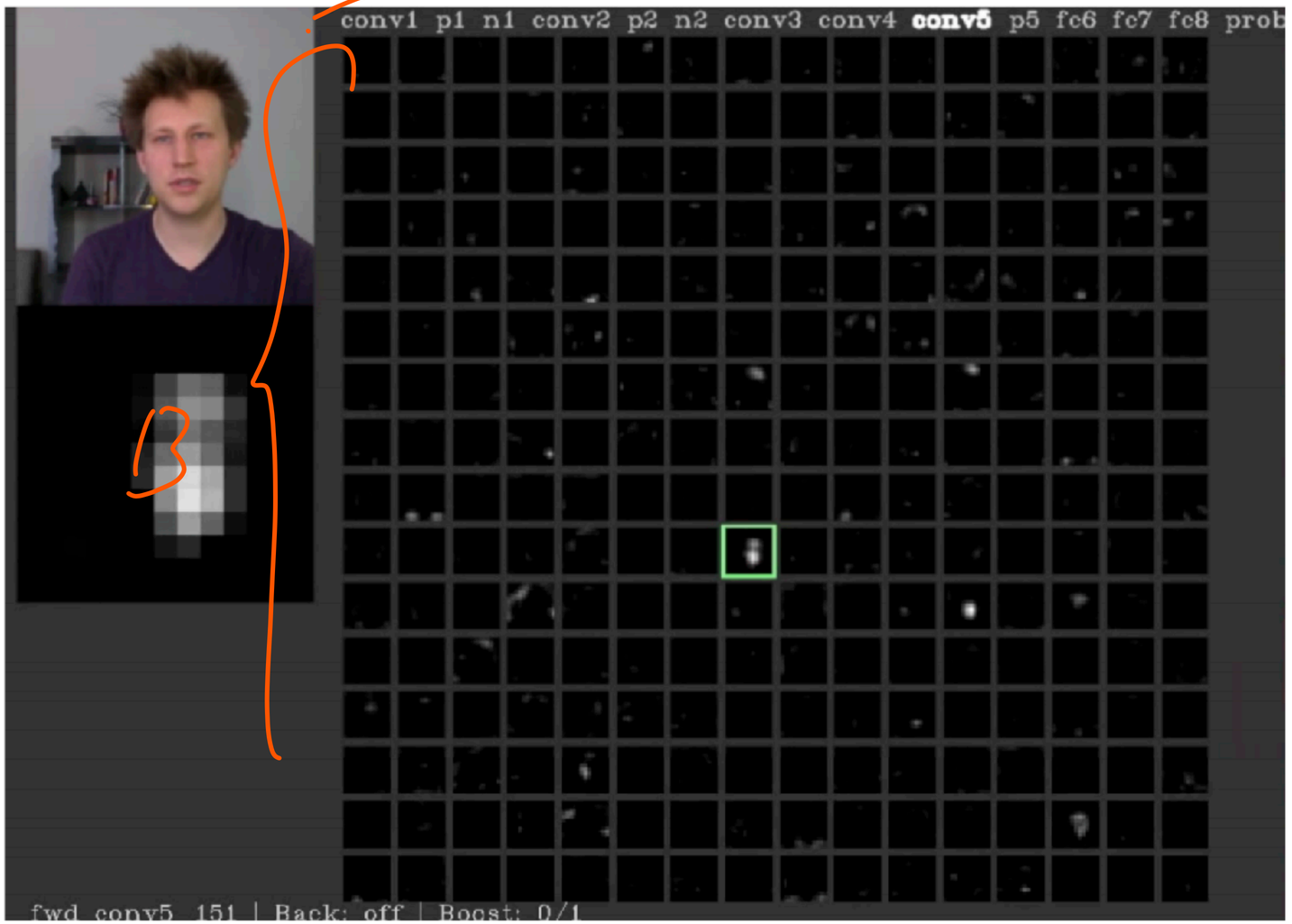
Plan for Today

- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- **Visualize activations at different layers**
- **Saliency versus Occlusion**
- **Guided backpropagation**
- **Gradient Ascent**
- **Feature Inversion**
- **DeepDreams**

Visualizing Activations: Which Filters 'Fire'?

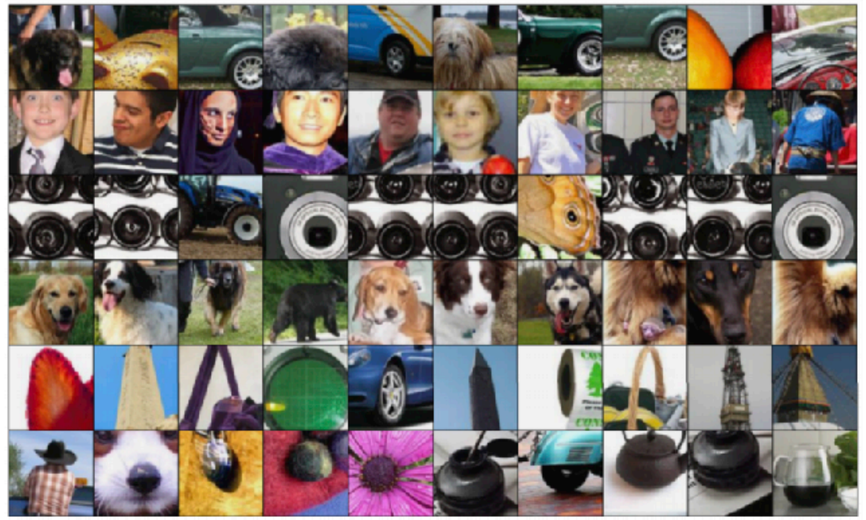
13x13 filters, 128 channels

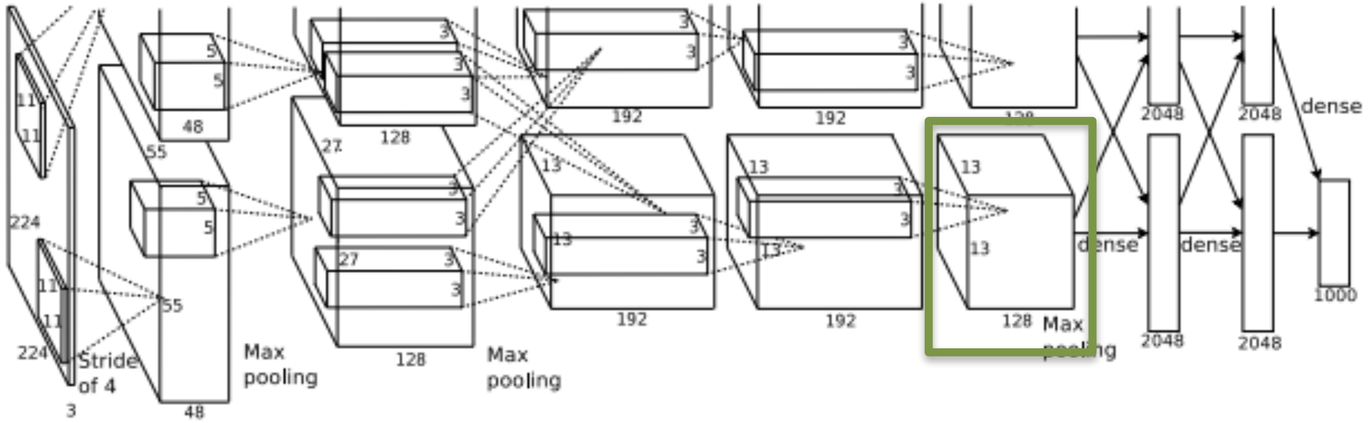




Maximally Active Filters

- Run many images through the network, record values of chosen channel.
- Visualize image patches that correspond to maximal activations.





STRIVING FOR SIMPLICITY:
THE ALL CONVOLUTIONAL NET

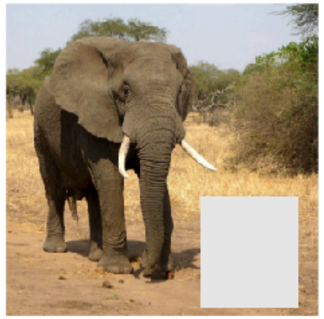
Jost Tobias Springenberg*, Alexey Dosovitskiy*, Thomas Brox, Martin Riedmiller
Department of Computer Science
University of Freiburg
Freiburg, 79110, Germany
{springj, dosovits, brox, riedmiller}@cs.uni-freiburg.de

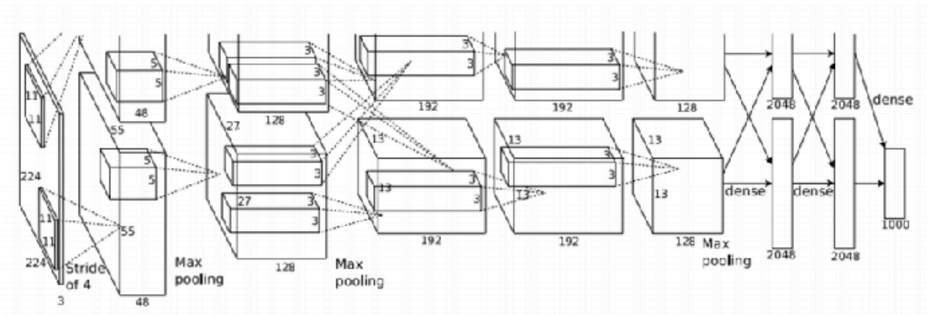
Plan for Today

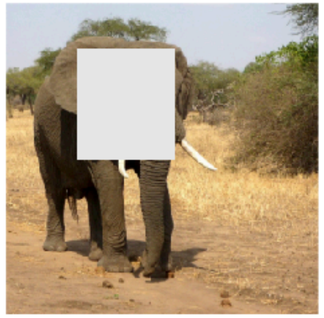
- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- **Saliency versus Occlusion**
- **Guided backpropagation**
- **Gradient Ascent**
- **Feature Inversion**
- **DeepDreams**

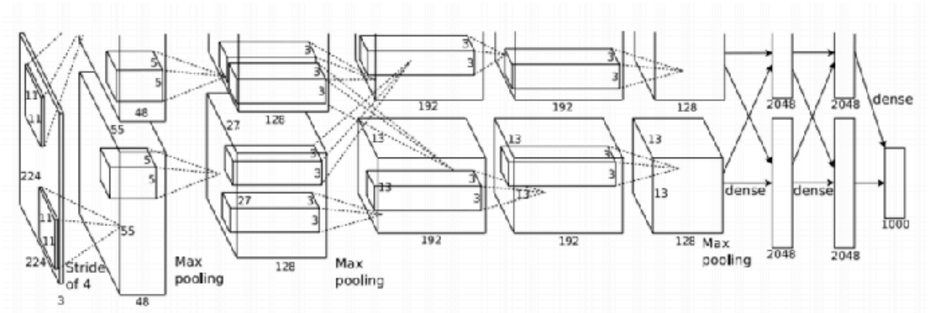
Saliency vs Occlusion: Which Pixels Matter?

- Mask out pixels before sending to CN
- Check sensitivity of class probability









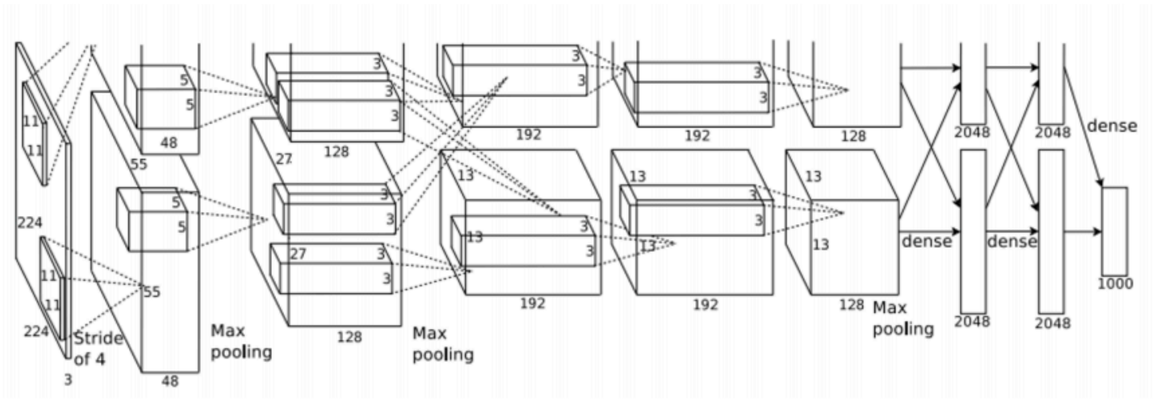
Visualizing and Understanding Convolutional Networks

Matthew D. Zeiler
Dept. of Computer Science, Courant Institute, New York University

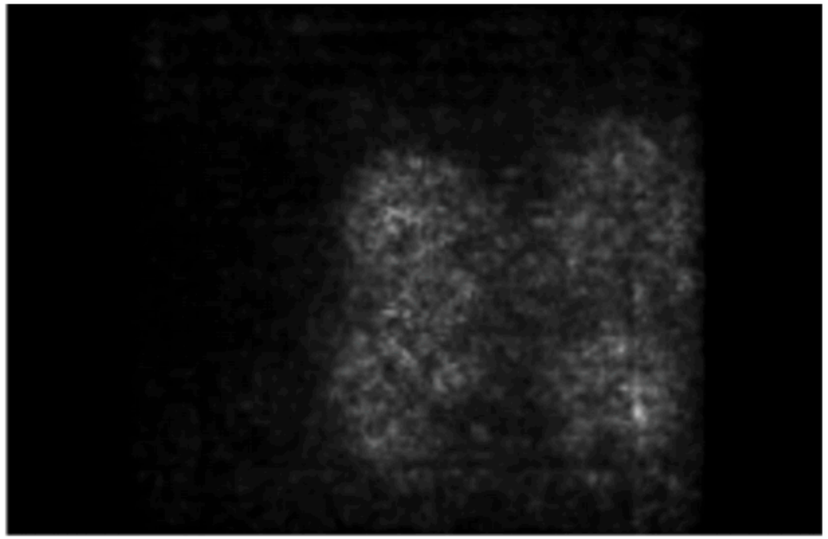
Rob Fergus
Dept. of Computer Science, Courant Institute, New York University

Saliency via Backprop

Forward pass

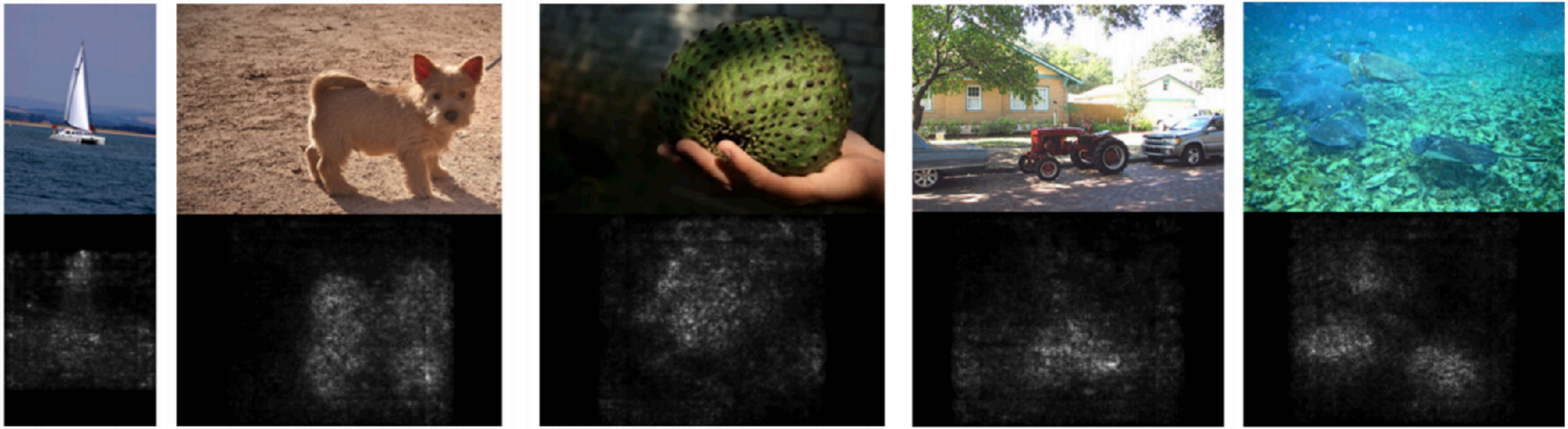


Backward pass



- **Compute gradients wrt image pixels**
- **Take absolute values**
- **Max over RGB channels**

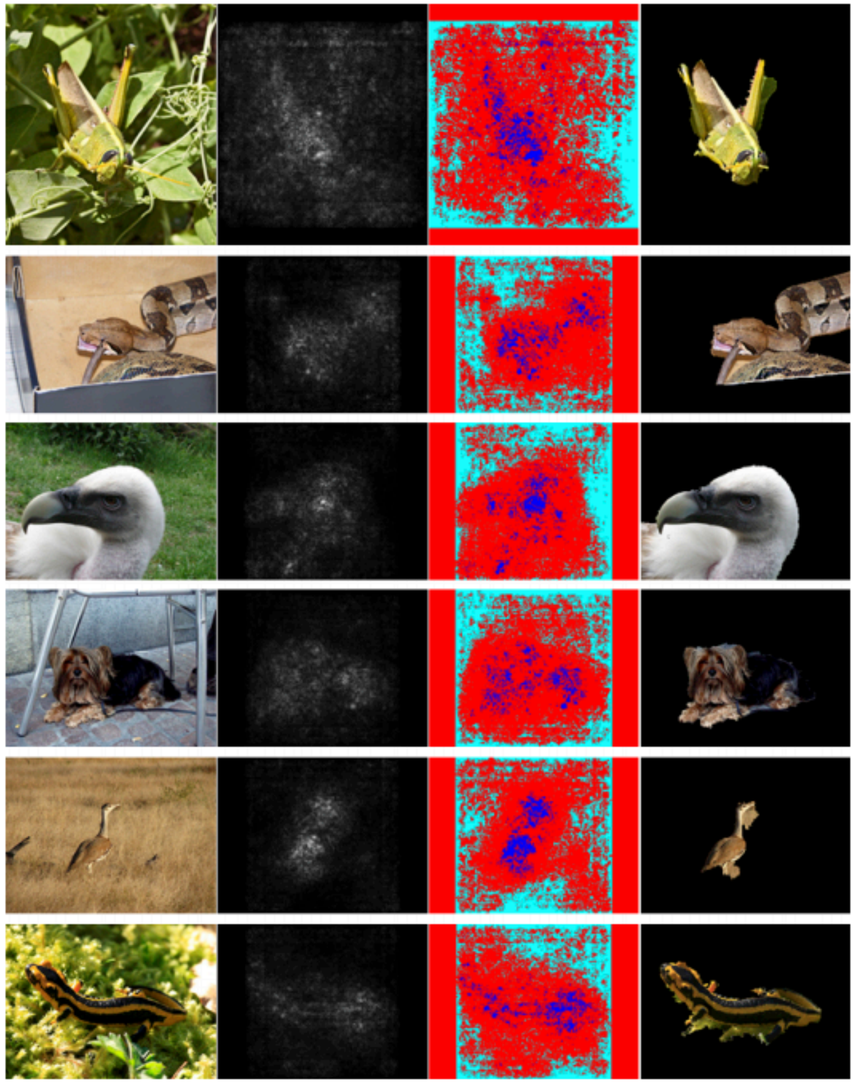
Saliency via Backprop



Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

Karen Simonyan Andrea Vedaldi Andrew Zisserman
Visual Geometry Group, University of Oxford
{karen,vedaldi,az}@robots.ox.ac.uk

Segmentation w/o Supervision



GrabCut on saliency maps

Plan for Today

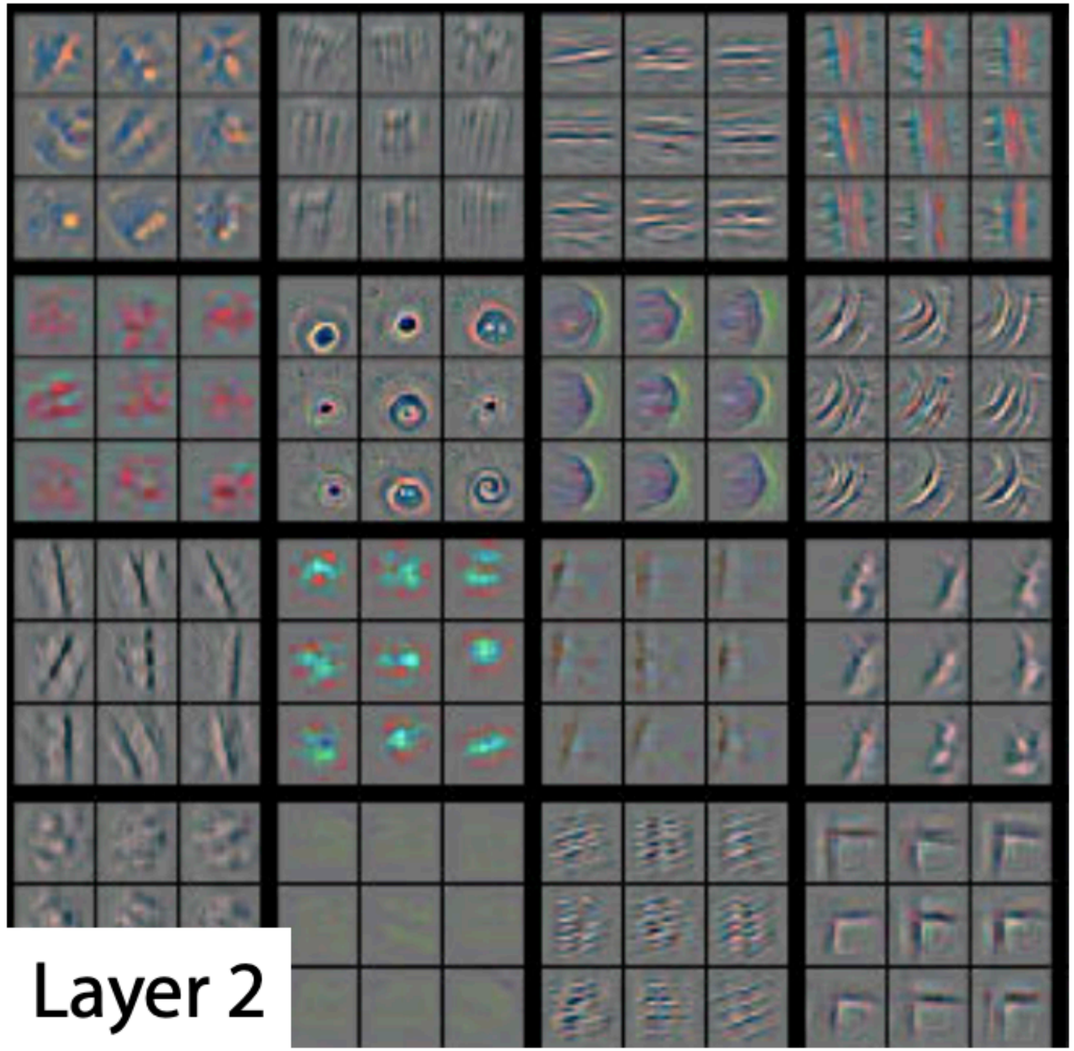
- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- Saliency versus Occlusion
- **Guided backpropagation**
- **Gradient Ascent**
- **Feature Inversion**
- **DeepDreams**

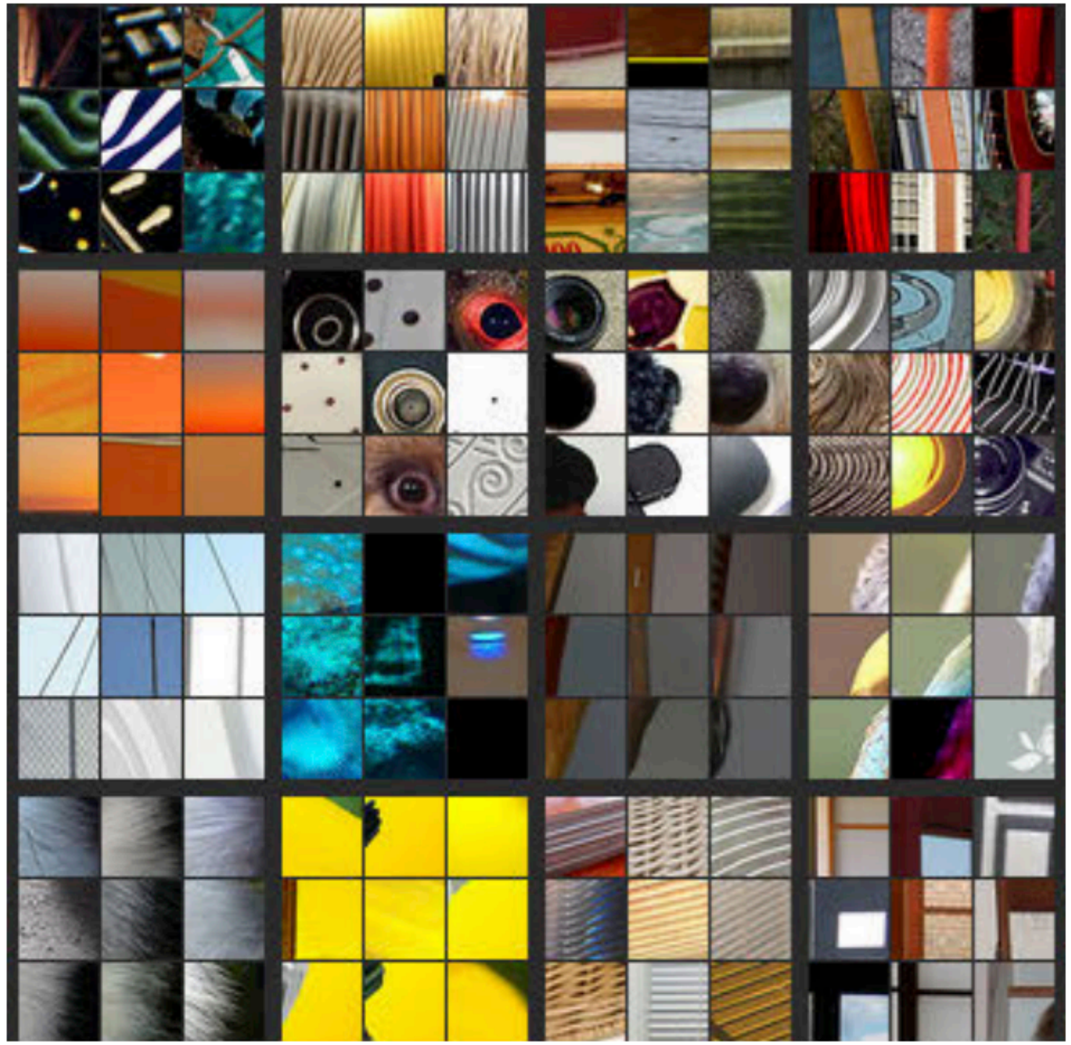
Guided Backprop

- **Pick single intermediate features e.g., one channel of 128x13x13 covn5 feature map**

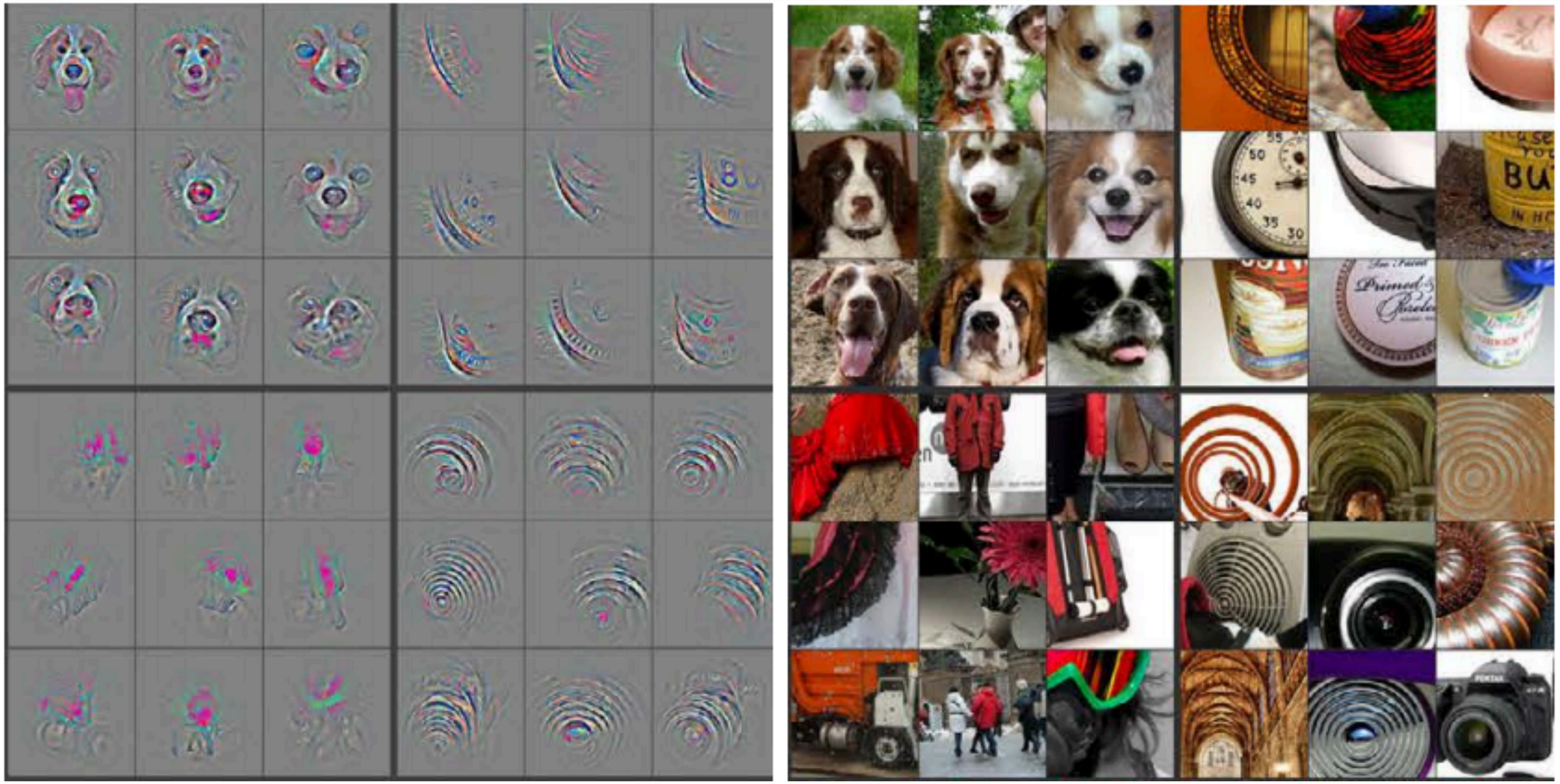
- **Compute gradients of neuron values wrt image pixels**

Guided Backprop (Layer 2)

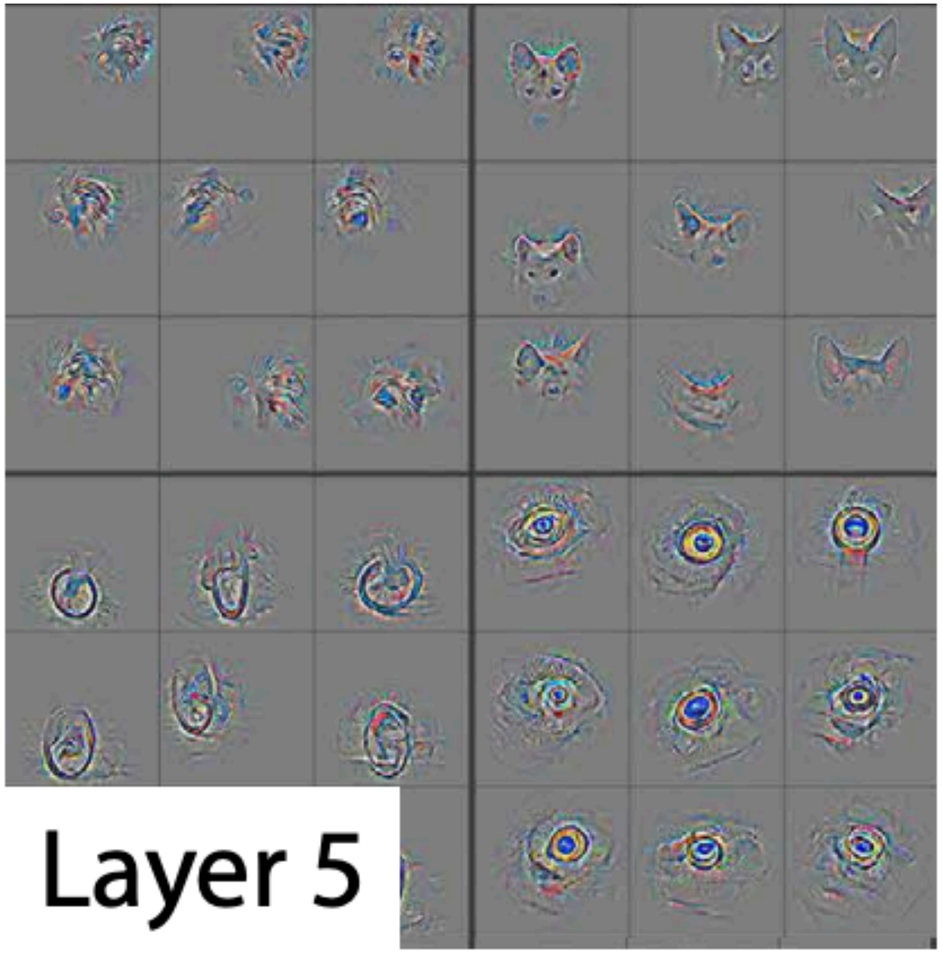




Guided Backprop (Layer 4)



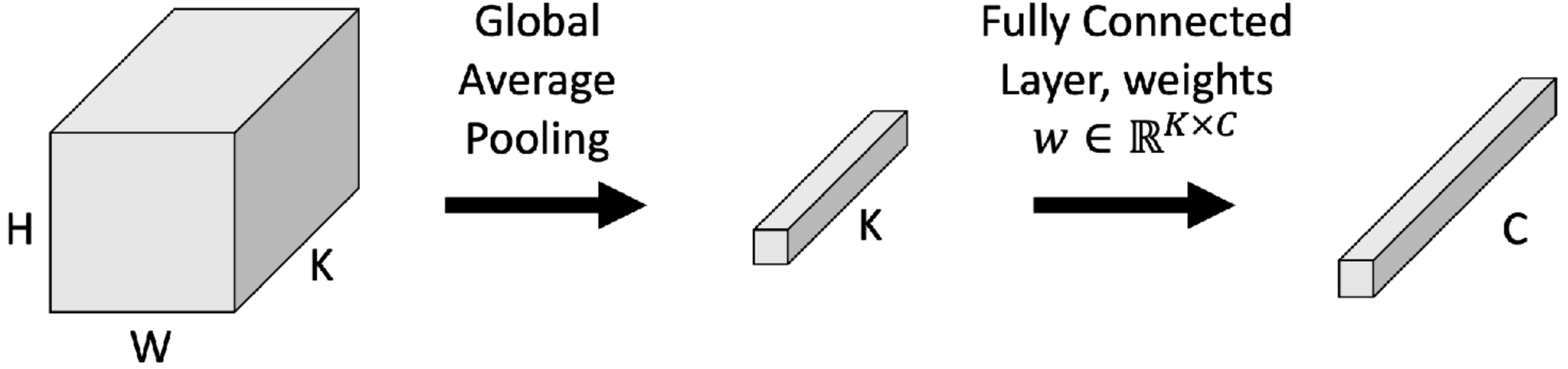
Guided Backprop (Layer 5)



Class Activation Map (CAM)

Learning Deep Features for Discriminative Localization

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba
Computer Science and Artificial Intelligence Laboratory, MIT
{bzhou, khosla, agata, oliva, torralba}@csail.mit.edu



Last layer CNN features:
$f \in \mathbb{R}^{H \times W \times K}$

Pooled features:
$F \in \mathbb{R}^{K}$

Class Scores:
$S \in \mathbb{R}^{C}$

$$F_k = \frac{1}{HW} \sum_{h,w} f_{h,w,k}$$

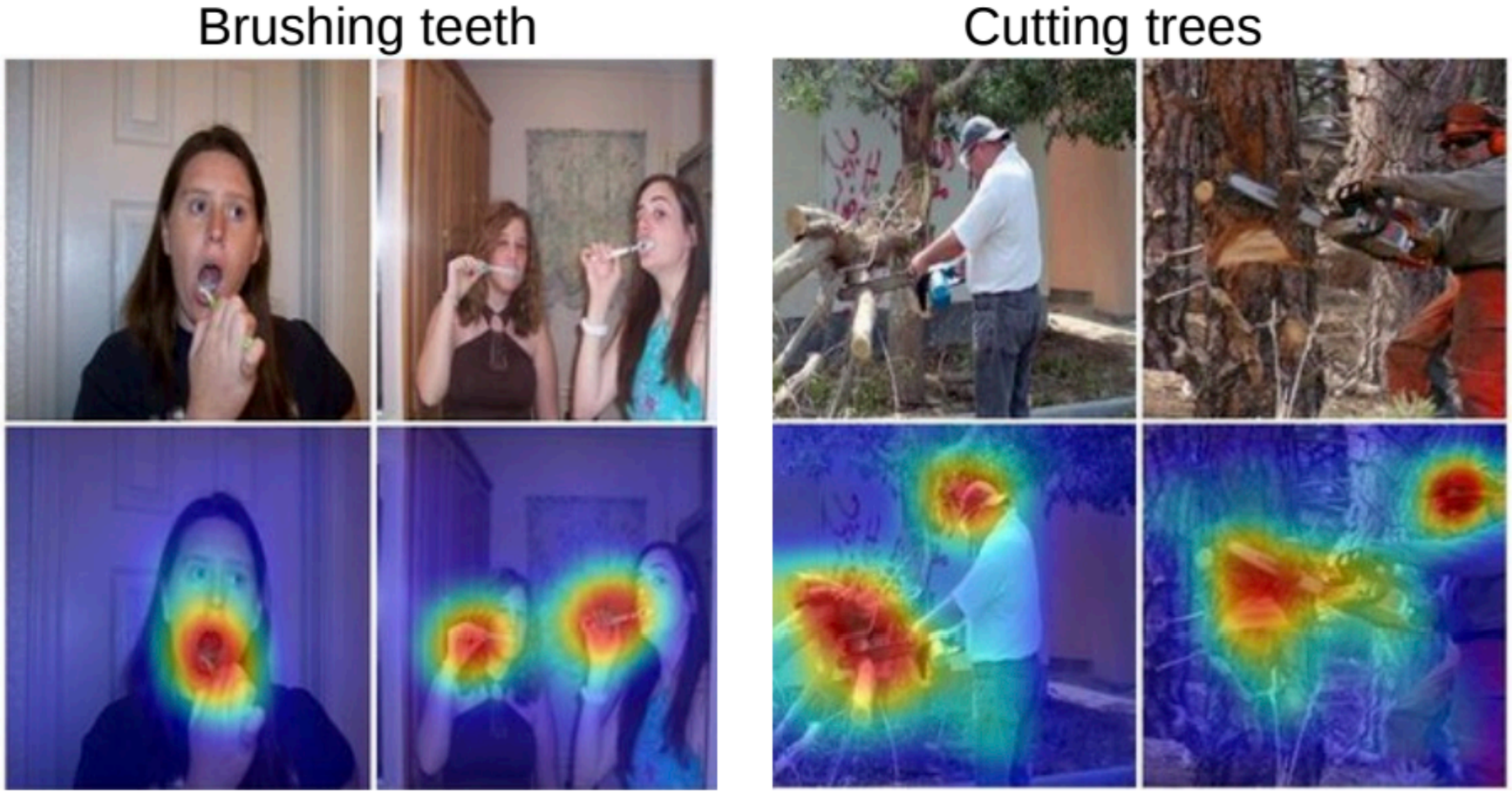
$$S_c = \sum_k w_{k,c} F_k$$

$$= \frac{1}{HW} \sum_{h,w} \sum_k w_{k,c} f_{h,w,k}$$

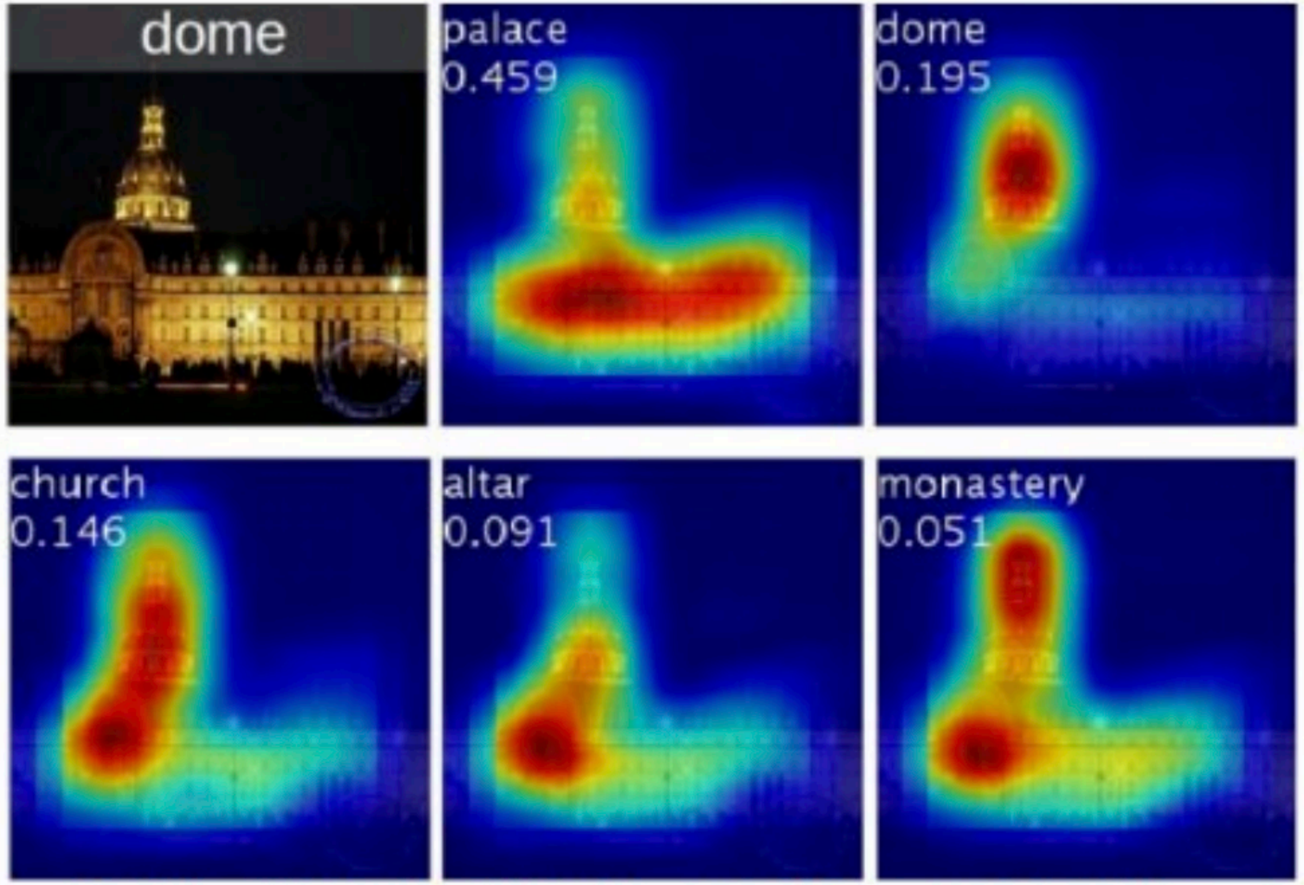
$$M_{c,h,w} = \sum_k w_{k,c} f_{h,w,k}$$

Class Activation Map

Class Activation Maps



Class Activation wrt Multiple Classes



Applied wrt only the last layer

Gradient-Weighted CAM (Grad-CAM)

1. **Pick a layer with activations** $A \in \mathbb{R}^{H \times W \times K}$

2. **Compute gradient of class score** S_c **with respect to** A**,** $\frac{\partial S_c}{\partial A} \in \mathbb{R}^{H \times W \times K}$

3. **Average pool the gradients to get weights** $\alpha \in \mathbb{R}^{K}$**,** $\alpha_k = \frac{1}{HW} \sum_{h,w} \frac{\partial S_c}{\partial A_{h,w,k}}$

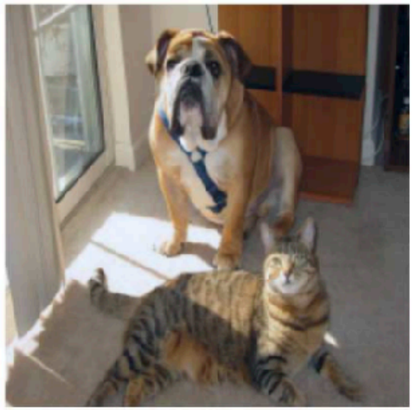
4. **Compute weighted activation maps** $M^c_{h,w} = ReLU\left(\sum_k \alpha_k A_{h,w,k} \right)$

Grad-CAM

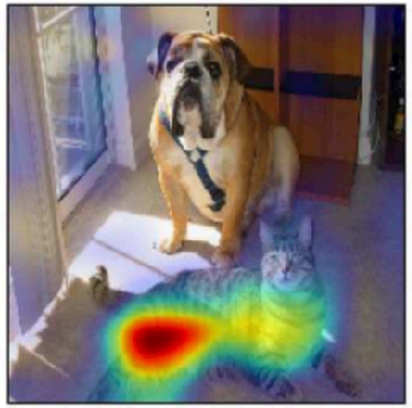
(but if noise will get worse)

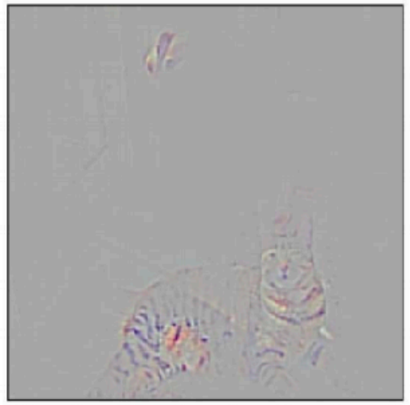
Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

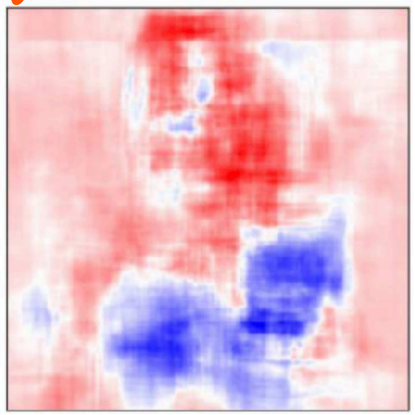
Ramprasaath R. Selvaraju · Michael Cogswell · Abhishek Das · Ramakrishna Vedantam · Devi Parikh · Dhruv Batra

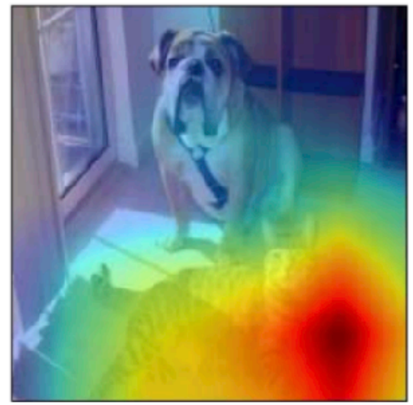

(a) Original Image


(b) Guided Backprop 'Cat'


(c) Grad-CAM 'Cat'


(d) Guided Grad-CAM 'Cat'


(e) Occlusion map for 'Cat'

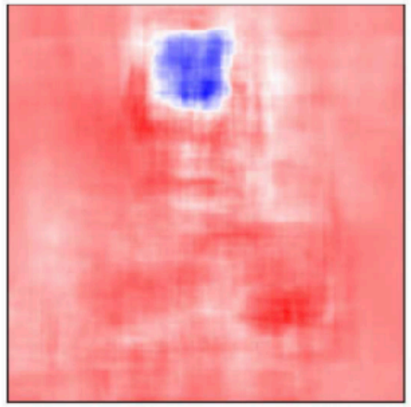

(f) ResNet Grad-CAM 'Cat'


(g) Original Image


(h) Guided Backprop 'Dog'

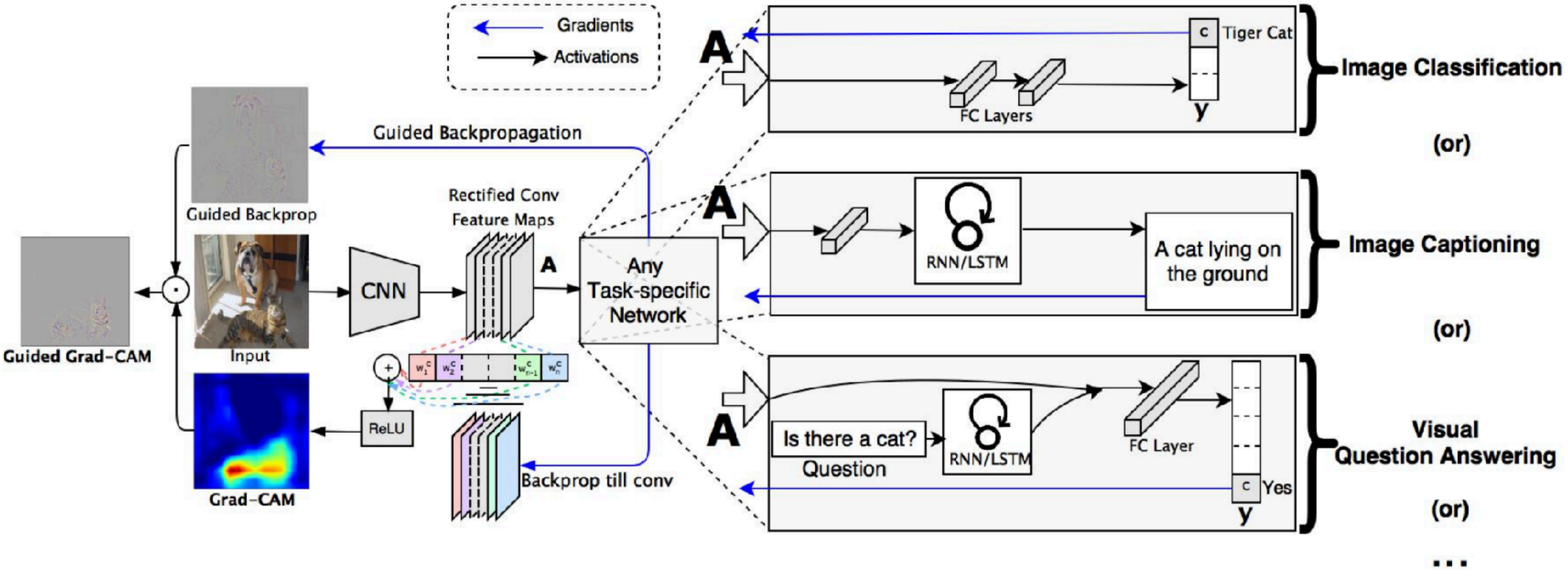

(i) Grad-CAM 'Dog'


(j) Guided Grad-CAM 'Dog'

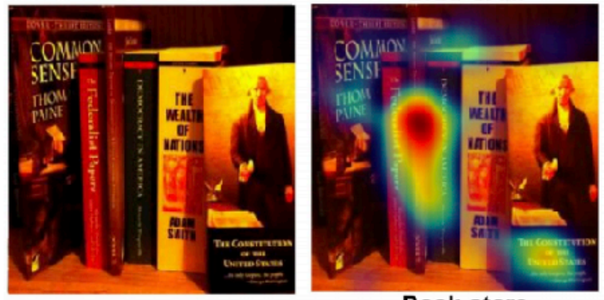

(k) Occlusion map for 'Dog'


(l) ResNet Grad-CAM 'Dog'

Guided-CAM


Gradients
Activations
Guided Backpropagation
Guided Backprop
Guided Grad-CAM
Input
CNN
Rectified Conv Feature Maps
A
Any Task-specific Network
ReLU
Grad-CAM
Backprop till conv
A
C
Tiger Cat
FC Layers
y
Image Classification
(or)
A
RNN/LSTM
A cat lying on the ground
Image Captioning
(or)
A
Is there a cat?
Question
RNN/LSTM
FC Layer
C
Yes
y
Visual Question Answering
(or)
...

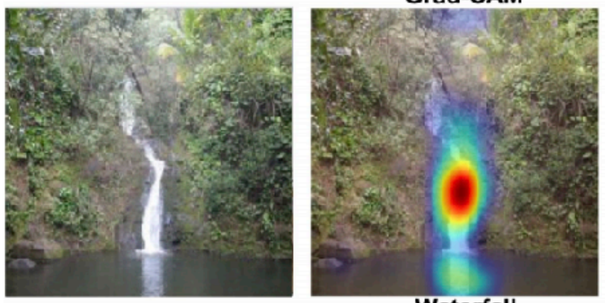
Grad-CAM on Image Captioning



(a)

Important concepts for `**Book-store**`

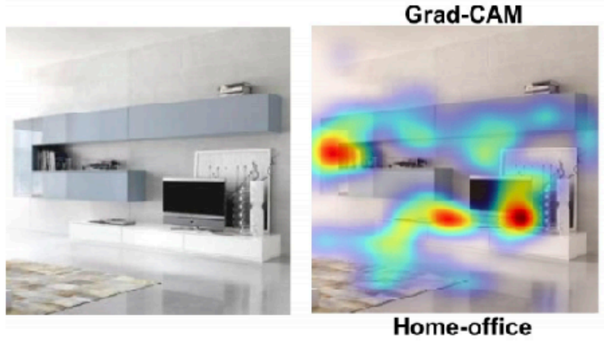
Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
78	Book	237	Sky
318	Book	357	road
502	Striped	148	Water
311	Shelf	404	Car
156	Swirly	71	Flower



(b)

Important concepts for `**Waterfall**`

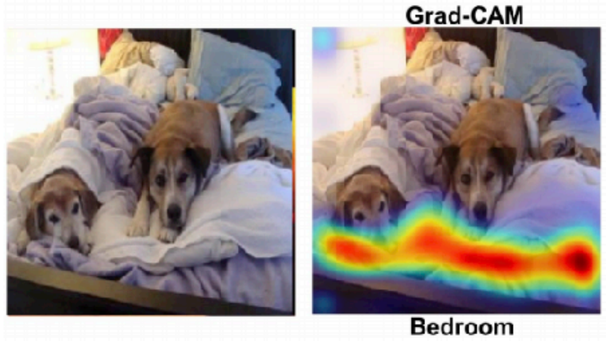
Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
117	Waterfall	115	Corridor
106	Closet	166	Road
148	Water	494	Bus
143	Water	106	Laundromat
216	Stratified	412	Grid



(c)

Important concepts for `**Home-office**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
78	Book	186	Chequered
312	Desk	237	Sky
75	Office	494	Swimming pool
492	Stove	334	Sidewalk
305	Screen	498	Crosswalk



(d)

Important concepts for `**Bedroom**`

Positive		Negative	
Neuron ID	Concept	Neuron ID	Concept
317	Bed	187	Spiralled
290	Bed	294	Pantry
226	Painting	26	Toiled
175	Cushion	9	Shoe shop
117	Waterfall	182	Amusement park

Plan for Today

- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- Saliency versus Occlusion
- Guided backpropagation
- **Gradient Ascent**
- **Feature Inversion**
- **DeepDreams**

Visualizing CNN Features: Gradient Ascent

- **Guided Backprop:**
  Find parts of image that maximally responds to a neuron

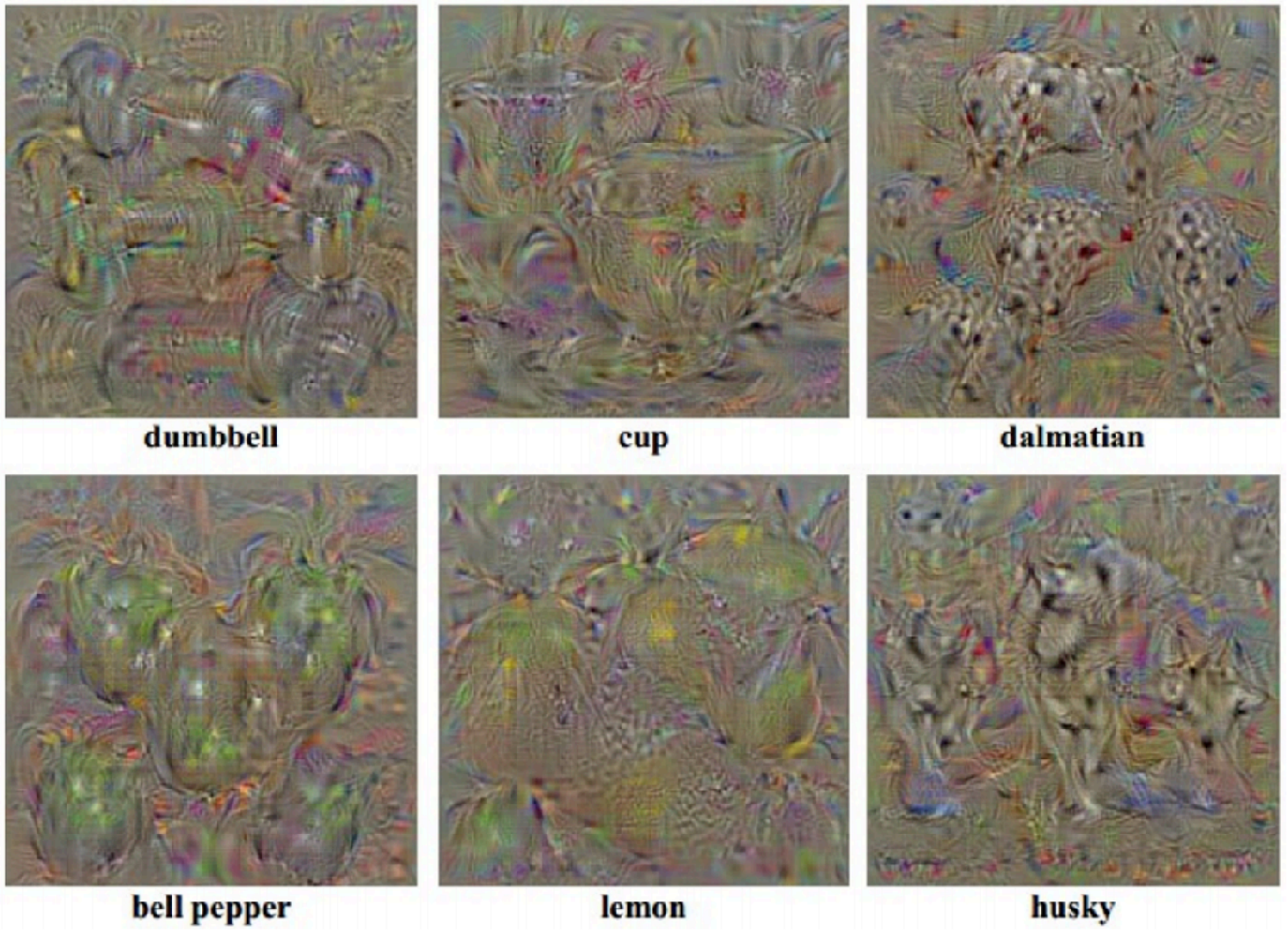
- **Gradient Ascent:**
  Find a synthetic image that maximally activates a neuron $$I^\star = \arg\max_I f(I) + \lambda R(I)$$

Gradient Ascent $I^\star = \arg\max_I \{S_c(I) - \lambda \|I\|^2\}$

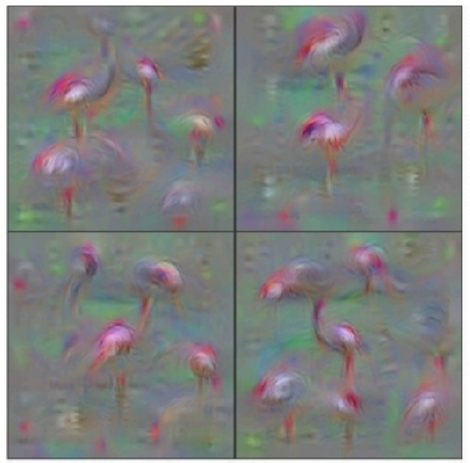
- **Initialize image** $I = 0$
- **Forward image to compute scores** $S_c(I)$
- **Backprop to get gradients of neuron values wrt image pixels**
- **Update image as** $I \leftarrow I + \frac{\partial S_c(I)}{\partial I}$

Gradient Ascent

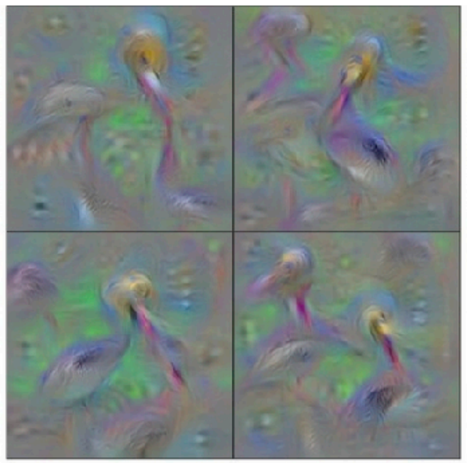
Other regularizers:
1. Gaussian blur
2. Clip pixels with values ~ 0
3. Clip pixels with gradients ~ 0



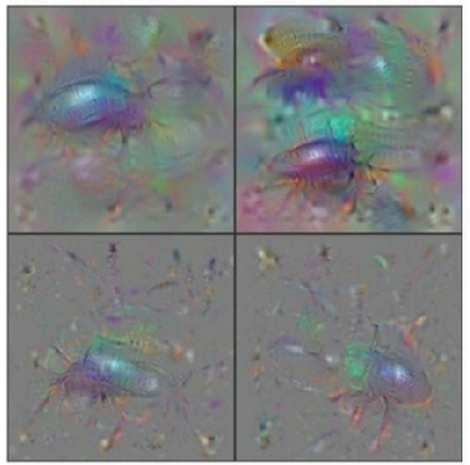
GA Results: More



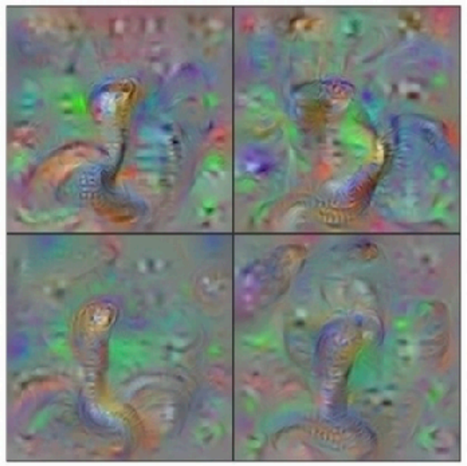
Flamingo



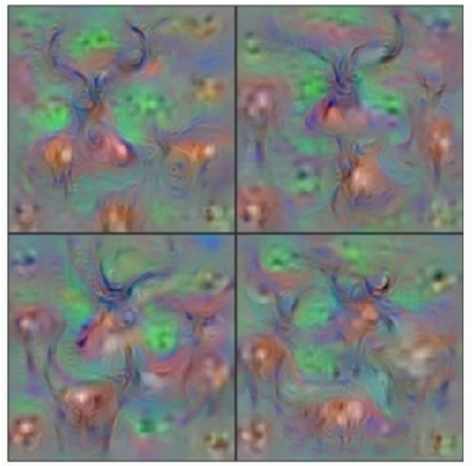
Pelican



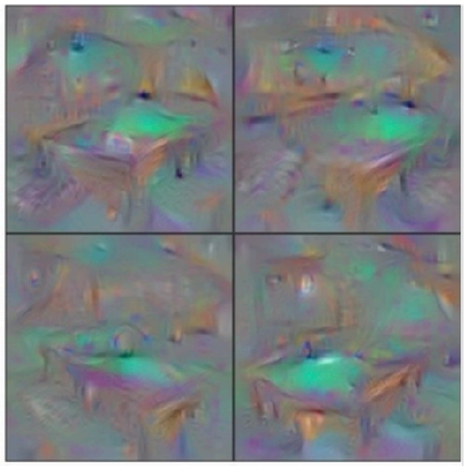
Ground Beetle



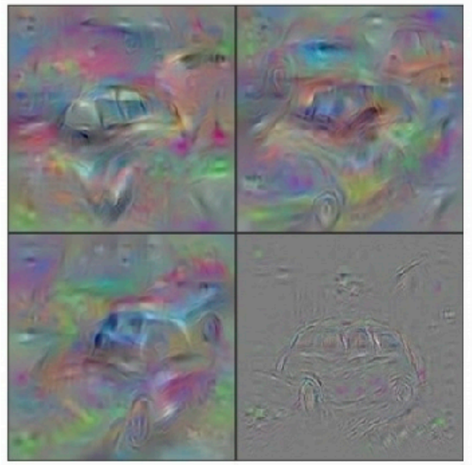
Indian Cobra



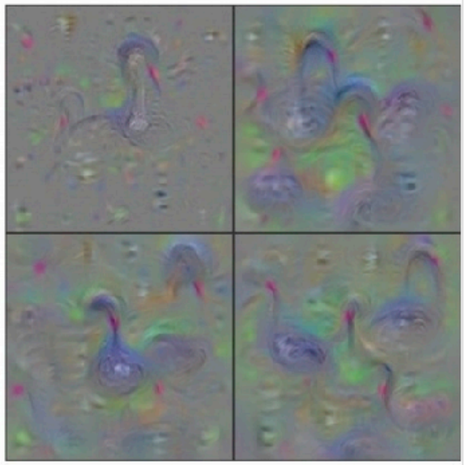
Hartebeest



Billiard Table

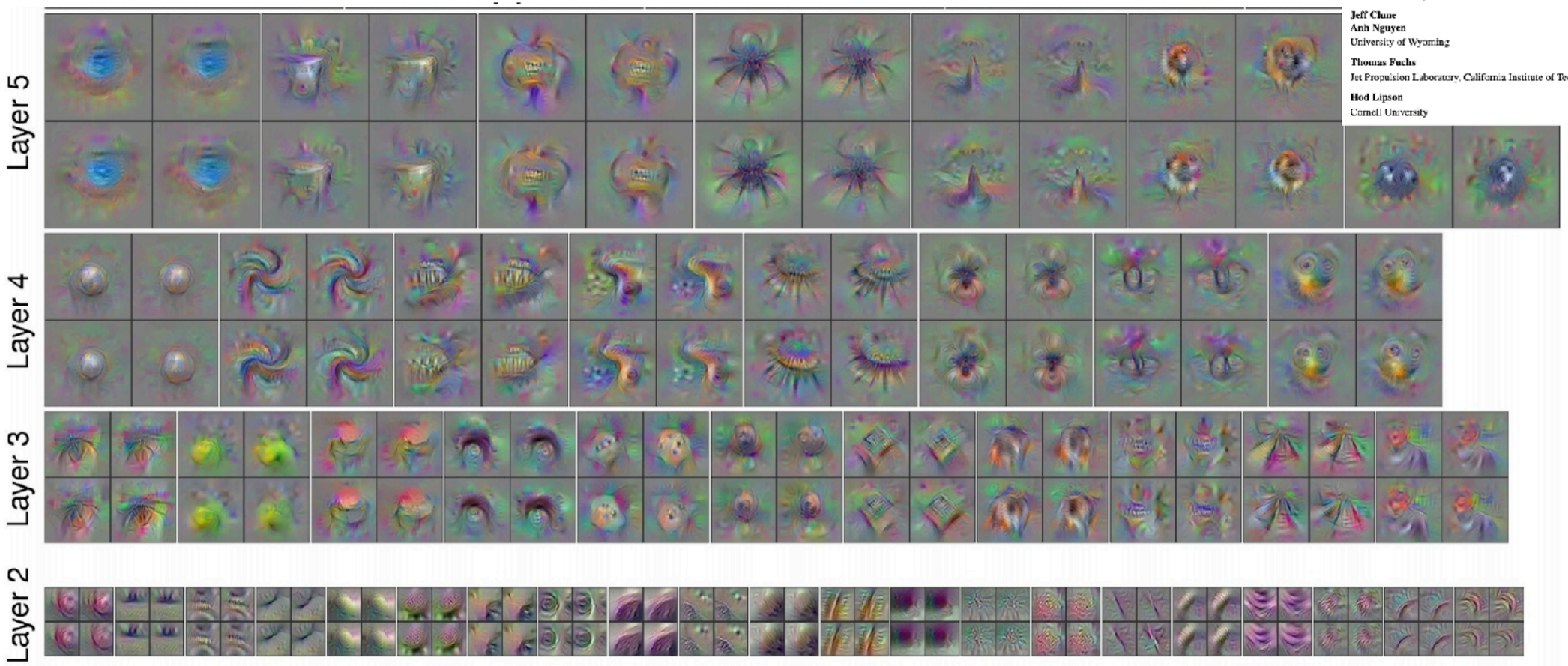


Station Wagon



Black Swan

Visualize Intermediate Features



Understanding Neural Networks Through Deep Visualization

Jason Yosinski YOSINSKI@CS.CORNELL.EDU
Cornell University

Jeff Clune JEFFCLUNE@UWYO.EDU
Anh Nguyen ANGUYEN8@UWYO.EDU
University of Wyoming

Thomas Fuchs FUCHS@CALTECH.EDU
Jet Propulsion Laboratory, California Institute of Technology

Hod Lipson HOD.LIPSON@CORNELL.EDU
Cornell University

Stronger Regularizers for GAs

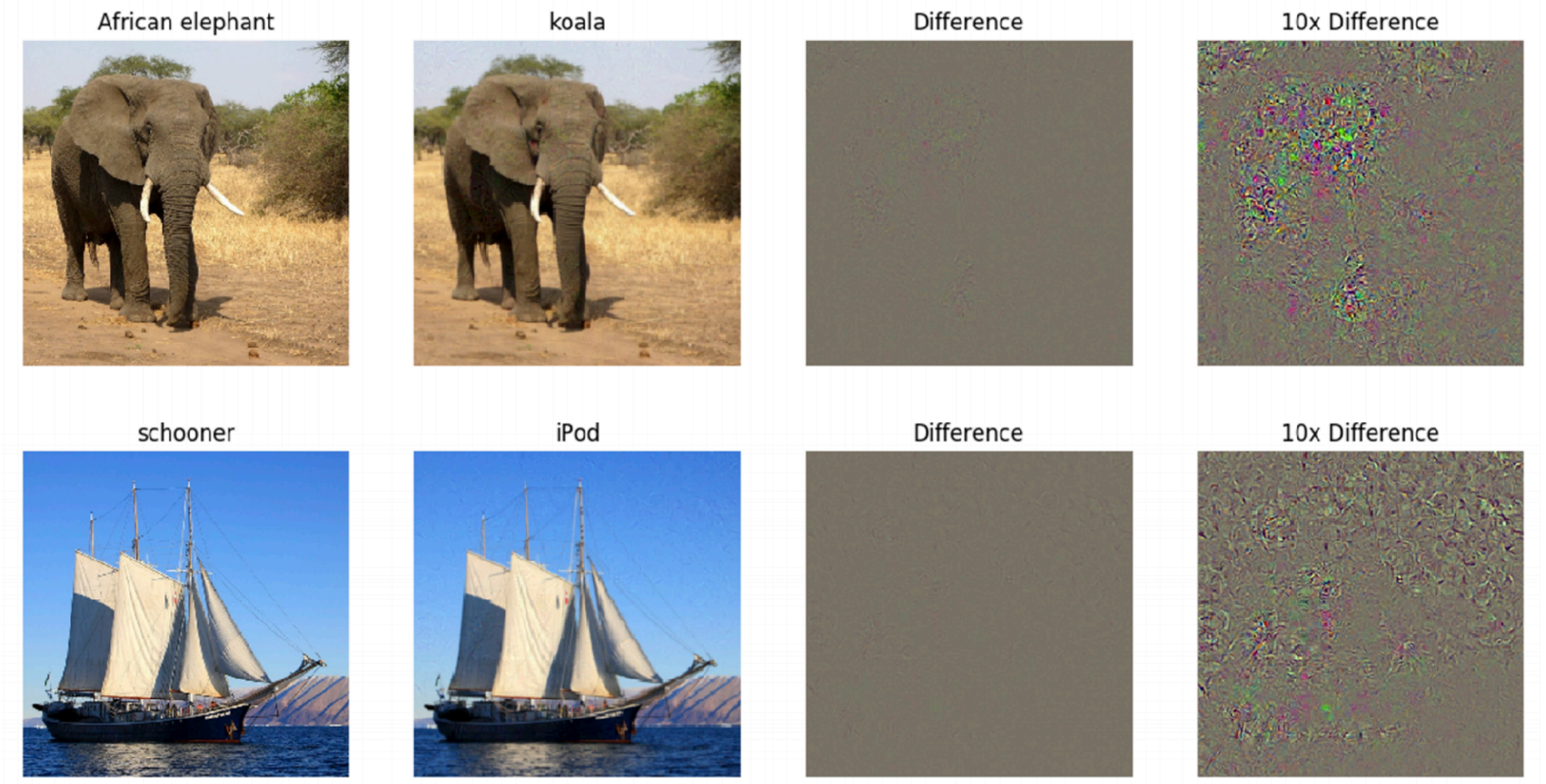
Reconstructions of multiple feature types (facets) recognized by the same “grocery store” neuron

Corresponding example training set images recognized by the same neuron as in the "grocery store" class

More results: Stronger Regularizers

Relation to Adversarial Training

Adversarial Training: Examples



Plan for Today

- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- Saliency versus Occlusion
- Guided backpropagation
- Gradient Ascent
- **Feature Inversion**
- **DeepDreams**

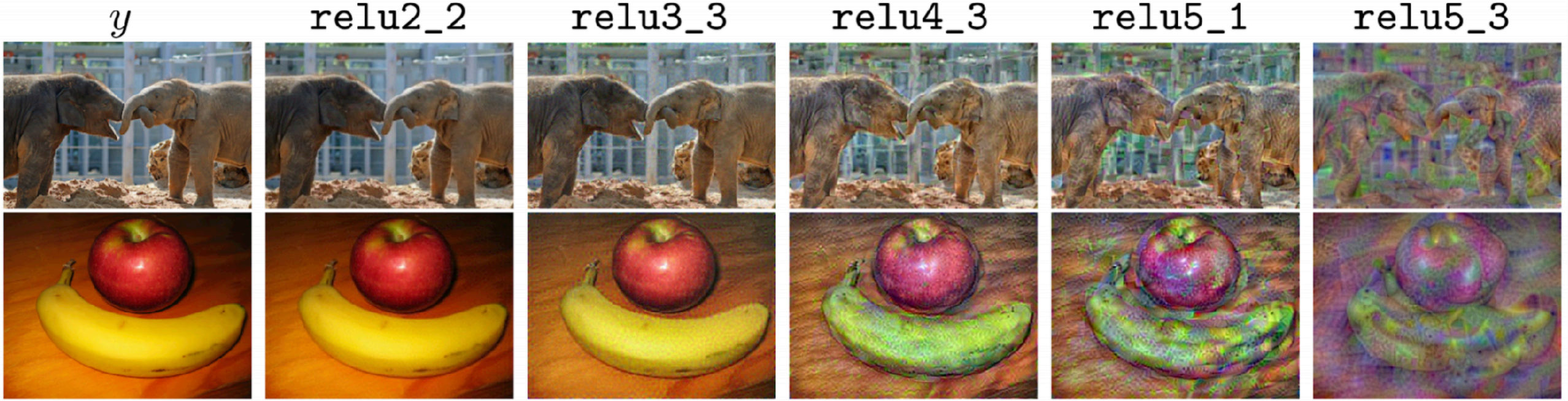
Feature Inversion

$$x^\star = \arg\min_I \{ l(\Phi(I), \Phi_0) + \lambda \mathcal{R}(I) \}$$

$$l(\Phi(I), \Phi_0) = \|\Phi(I) - \Phi_0\|^2$$

$$\mathcal{R}_\beta(I) = \sum_{i,j} \left(I(i, j+1) - I(i,j) \right)^2 + \left(I(i+1, j) - I(i,j) \right)^2 \Big)^{\frac{\beta}{2}}$$

Feature Inversion

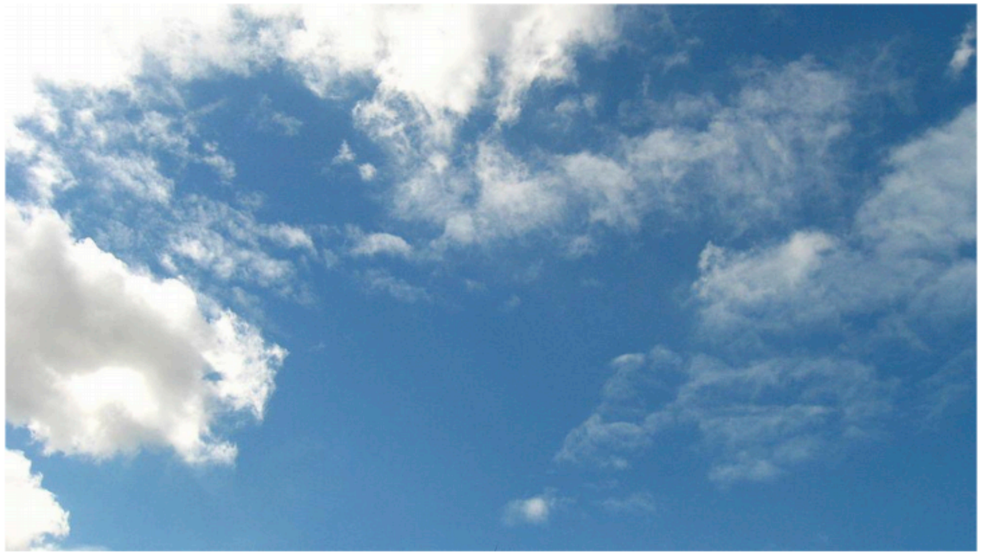


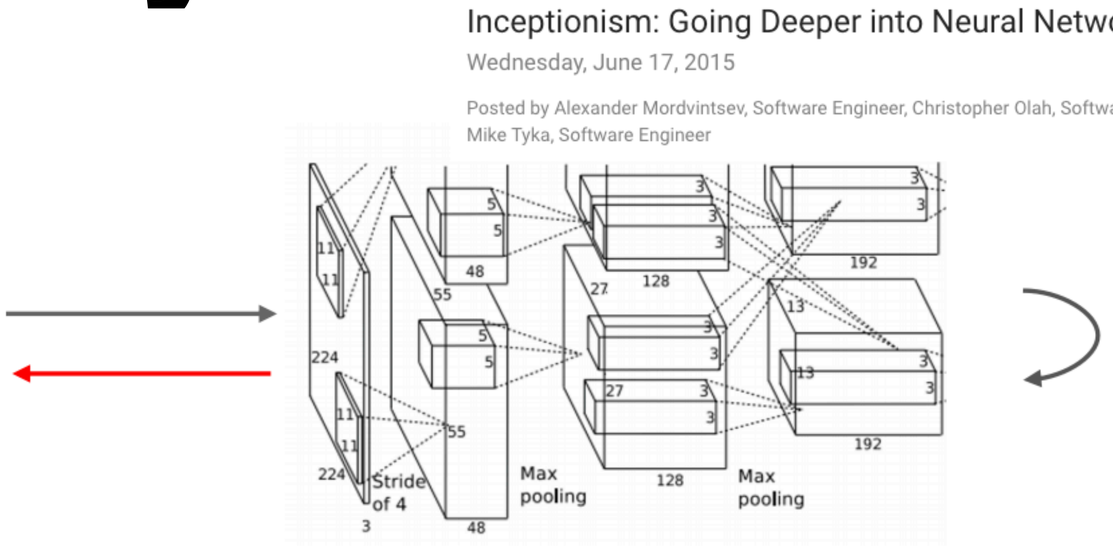
Plan for Today

- Visualize filter (kernels)
- Use CNNs as feature maps
- What are the different layers learning?
- Visualize activations at different layers
- Saliency versus Occlusion
- Guided backpropagation
- Gradient Ascent
- Feature Inversion
- **DeepDreams**

Deep Dream: Amplify Features

- C
- S
- C
- U





$$I^\star = \arg\max_I \sum_i f_i(I)^2$$

DeepDream

DeepDream