Microsoft Research

Each year Microsoft Research hosts hundreds of influential speakers from around the world including leading scientists, renowned experts in technology, book authors, and leading academics, and makes videos of these lectures freely available.


Deep Learning for Computer Vision

NIPS 2013 Tutorial

Rob Fergus

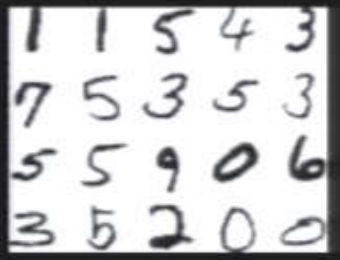
Dept. of Computer Science
New York University

Overview

- Primarily about object recognition, using supervised ConvNet models

- Focus on natural images
  - Rather than digits
  - Classification & Detection

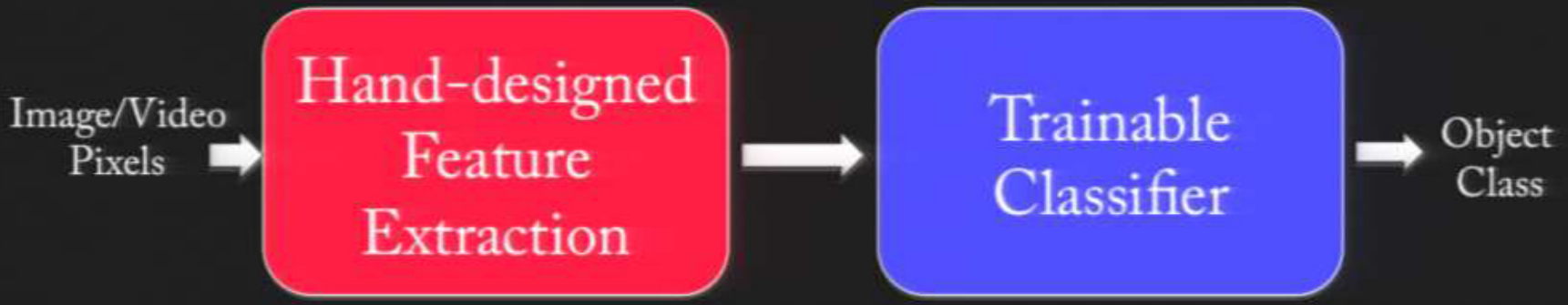
instead of



- Brief discussion of other vision problems

Motivation

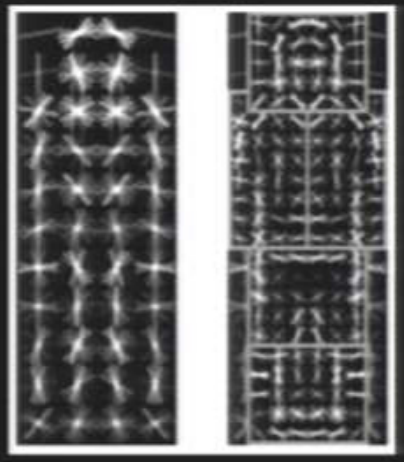
Existing Recognition Approach



- Features are not learned
- Trainable classifier is often generic (e.g. SVM)

Motivation

- Features are key to recent progress in recognition
- Multitude of hand-designed features currently in use
  - SIFT, HOG, LBP, MSER, Color-SIFT.....
- Where next? Better classifiers? Or keep building more features?



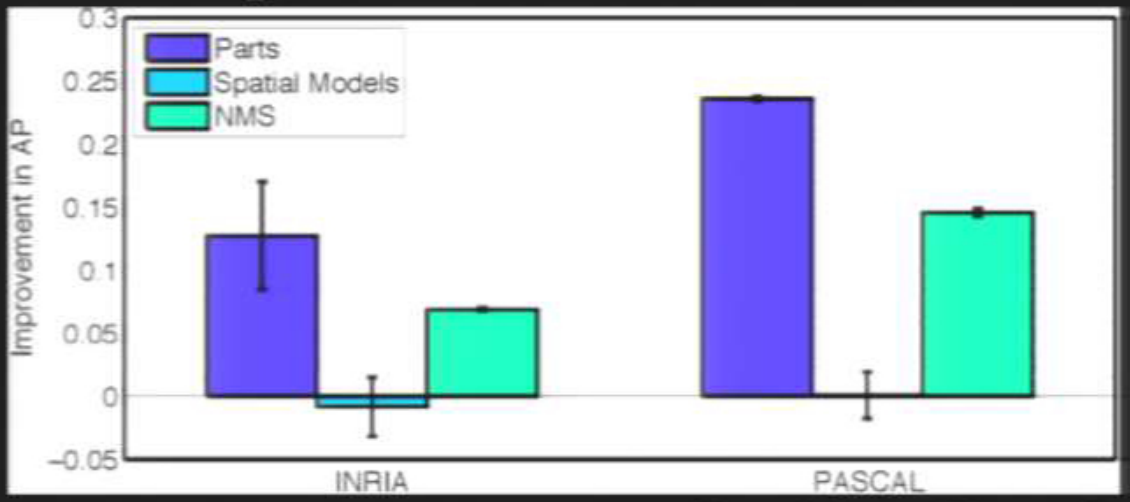
Felzenszwalb, Girshick, McAllester and Ramanan, PAMI 2007

Yan & Huang
(Winner of PASCAL 2010 classification competition)

What Limits Current Performance?

- Ablation studies on Deformable Parts Model
  - Felzenszwalb, Girshick, McAllester, Ramanan, PAMI'10

- Replace each part with humans (Amazon Turk):

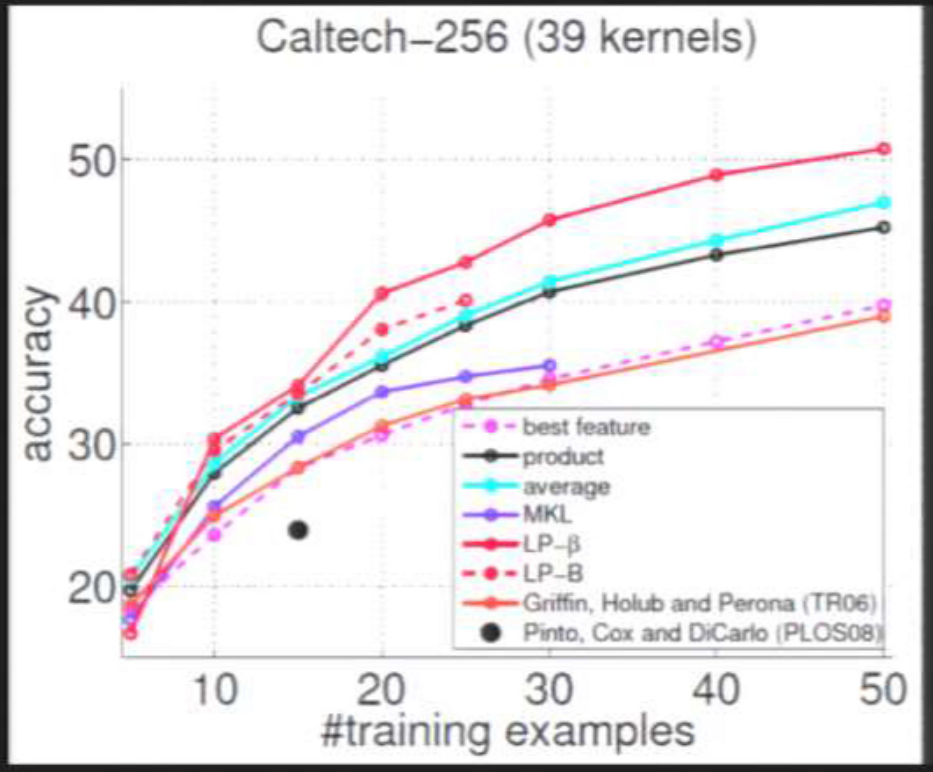


Parikh & Zitnick, CVPR'10

Hand-Crafted Features

- LP-β Multiple Kernel Learning
  - Gehler and Nowozin, On Feature Combination for Multiclass Object Classification, ICCV'09
- 39 different kernels
  - PHOG, SIFT, V1S+, Region Cov. Etc.
- MKL only gets few % gain over averaging features

→ Features are doing the work

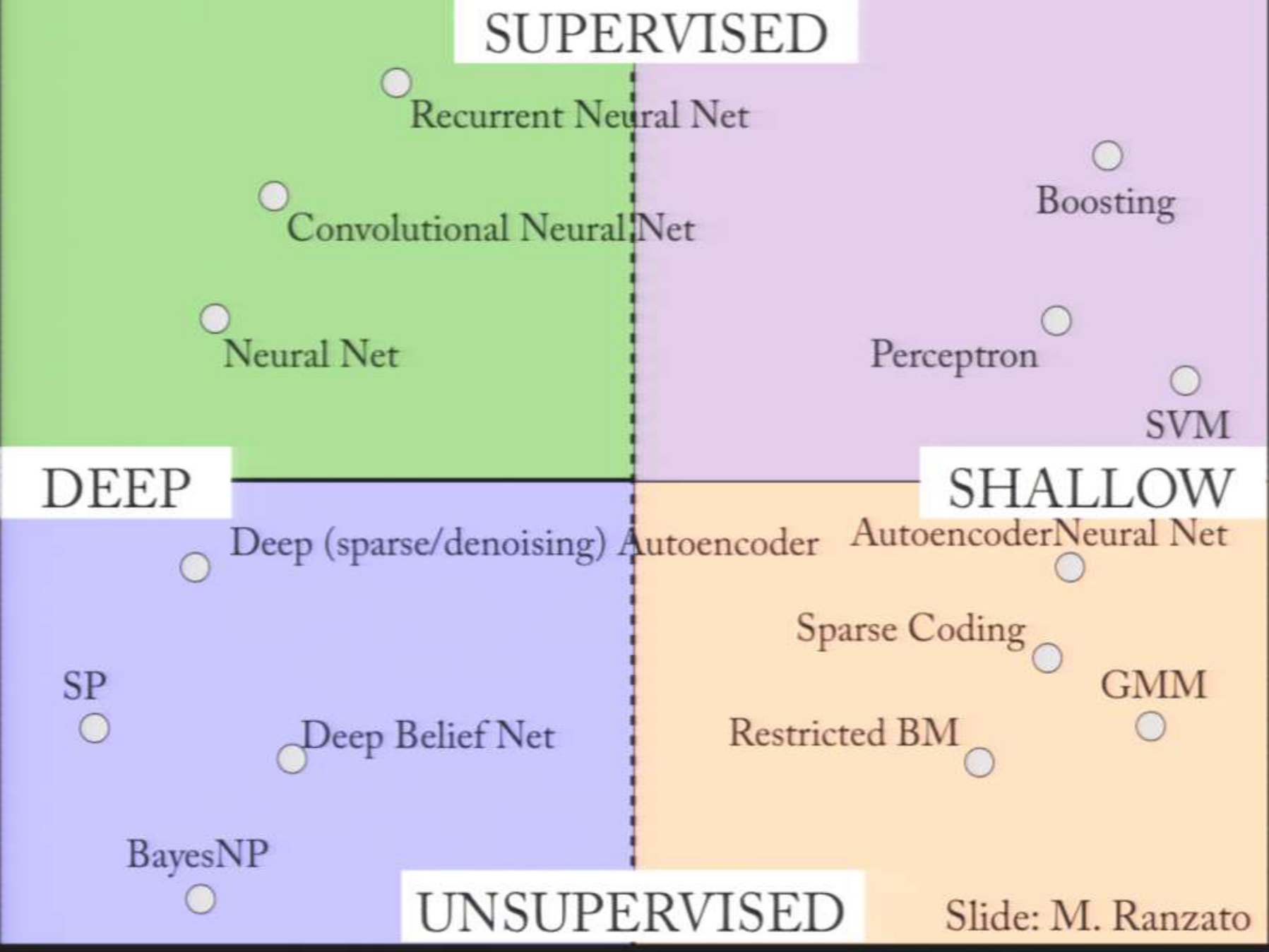


What about Learning the Features?

- Perhaps get better performance?
- Deep models: hierarchy of feature extractors
- All the way from pixels → classifier
- One layer extracts features from output of previous layer

- Train all layers jointly

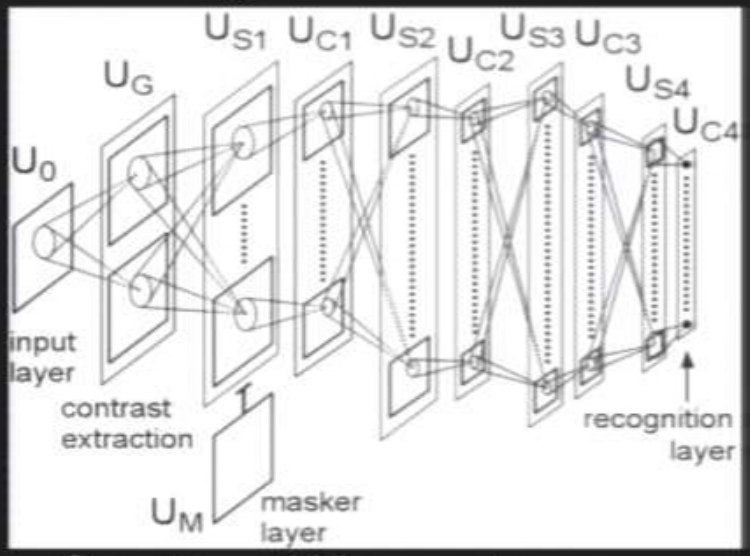
Deep Learning


SUPERVISED
Recurrent Neural Net
Convolutional Neural Net
Neural Net
Boosting
Perceptron
SVM
DEEP
SHALLOW
Deep (sparse/denoising) Autoencoder
AutoencoderNeural Net
Sparse Coding
GMM
SP
Deep Belief Net
Restricted BM
BayesNP
UNSUPERVISED
Slide: M. Ranzato

Multistage Hubel-Wiesel Architecture

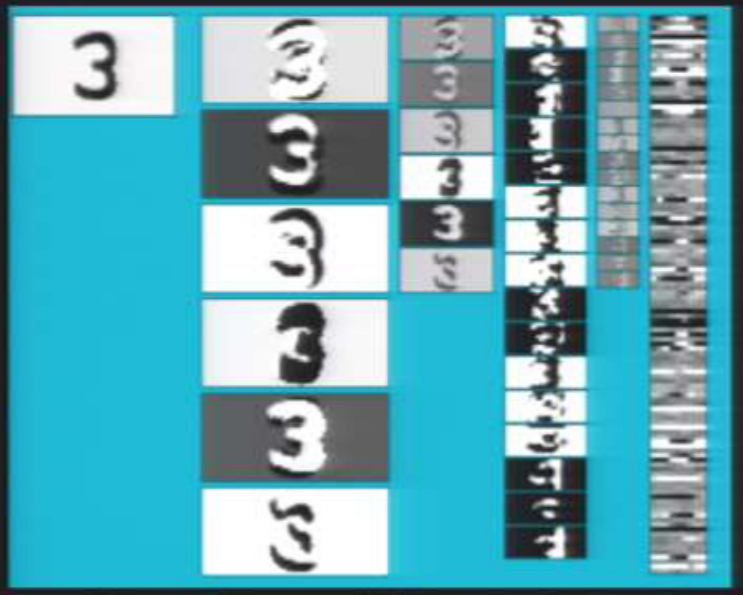
Slide: Y.LeCun

- [Hubel & Wiesel 1962]
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a retinotopic neighborhood.



Cognitron / Neocognitron
[Fukushima 1971-1982]

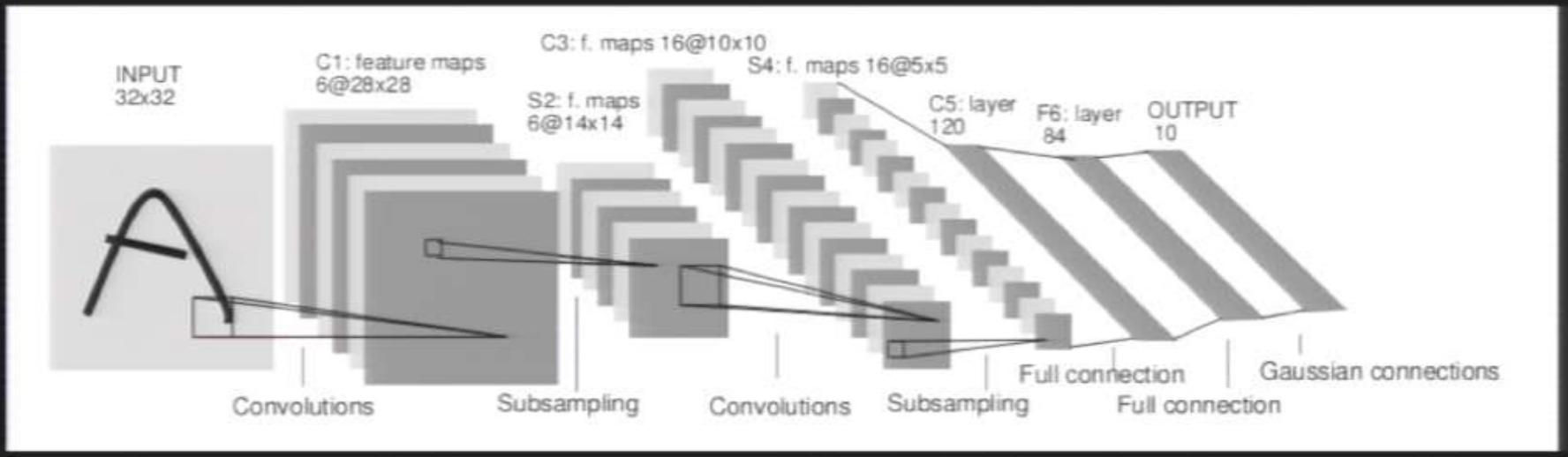
- Also HMAX [Poggio 2002-2006]



Convolutional Networks
[LeCun 1988-present]

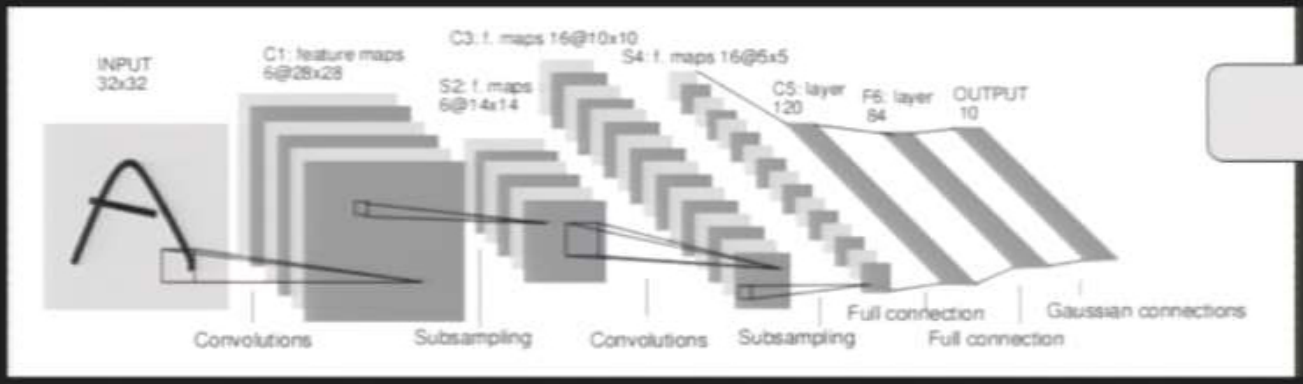
Convolutional Neural Networks

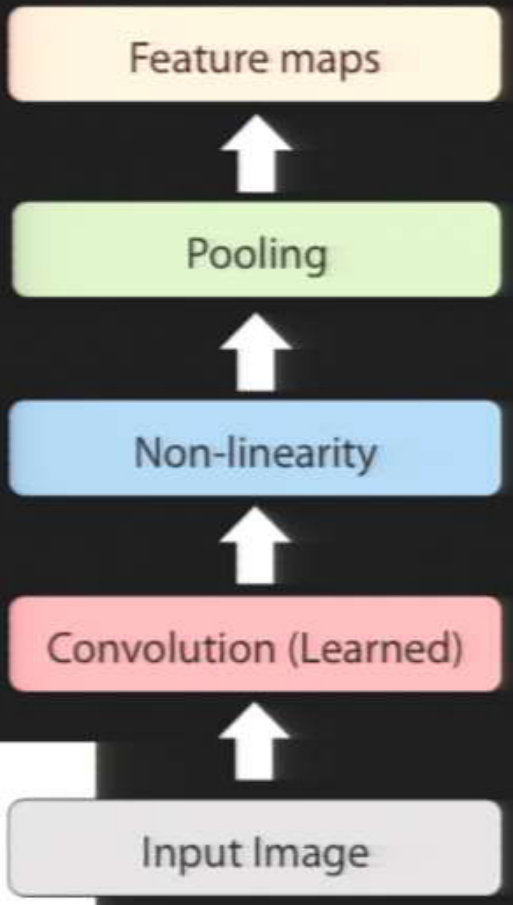
- LeCun et al. 1989
- Neural network with specialized connectivity structure



Recap of Convnets

- Feed-forward:
  - Convolve input
  - Non-linearity (rectified linear)
  - Pooling (local max)
- Supervised
- Train convolutional filters by back-propagating classification error





[LeCun et al. 1989]

Convnet Successes

- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

Convnet Successes

- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

- Simpler recognition benchmarks
  - CIFAR-10 (9.3% error [Wan et al. 2013])
  - Traffic sign recognition
    - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

Convnet Successes

- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

- Simpler recognition benchmarks
  - CIFAR-10 (9.3% error [Wan et al. 2013])
  - Traffic sign recognition
    - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]

- But (until recently) less good at more complex datasets
  - E.g. Caltech-101/256 (few training examples)

Application to ImageNet

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

[Deng et al. CVPR 2009]

Application to ImageNet

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk

[Deng et al. CVPR 2009]

ImageNet Classification with Deep Convolutional Neural Networks [NIPS 2012]

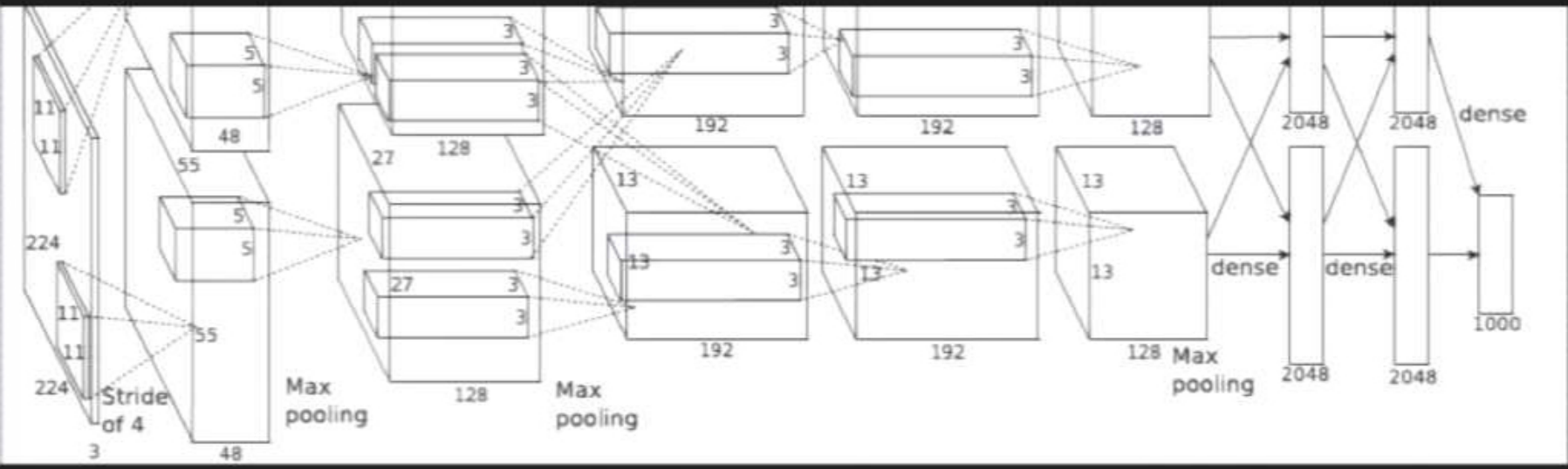
Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

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

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

Krizhevsky et al. [NIPS2012]

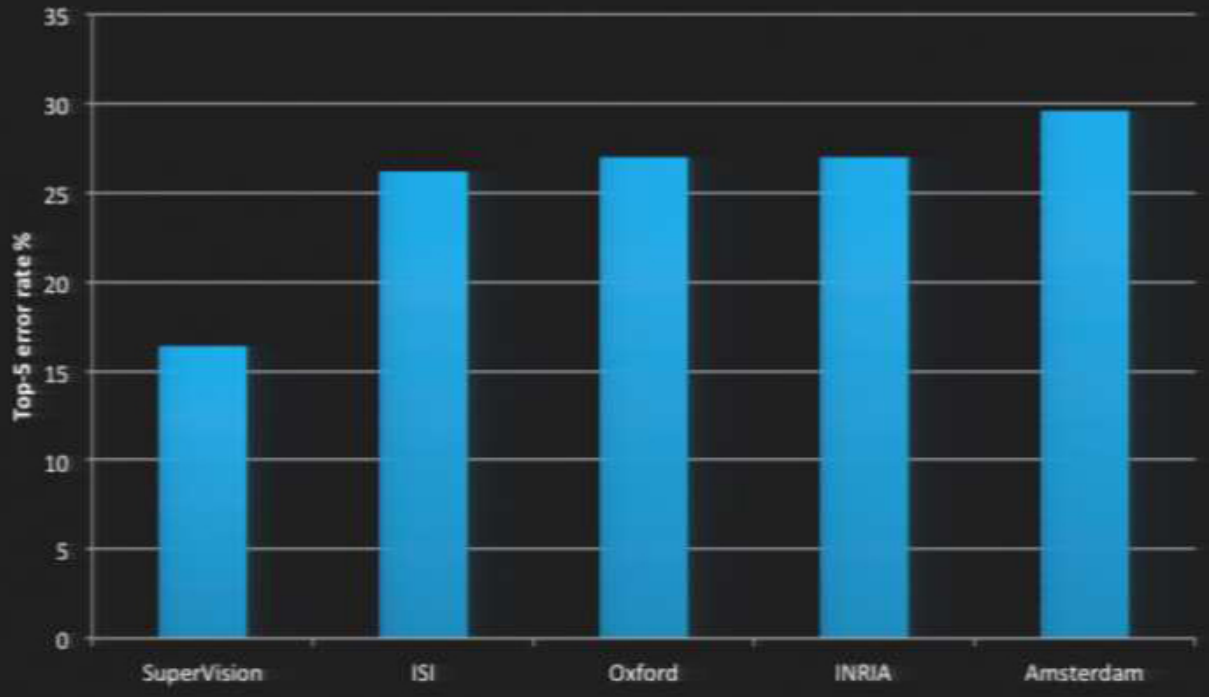
- Same model as LeCun'98 but:
  - Bigger model (8 layers)
  - More data (10^6 vs 10^3 images)
  - GPU implementation (50x speedup over CPU)
  - Better regularization (DropOut)



- 7 hidden layers, 650,000 neurons, 60,000,000 parameters
- Trained on 2 GPUs for a week

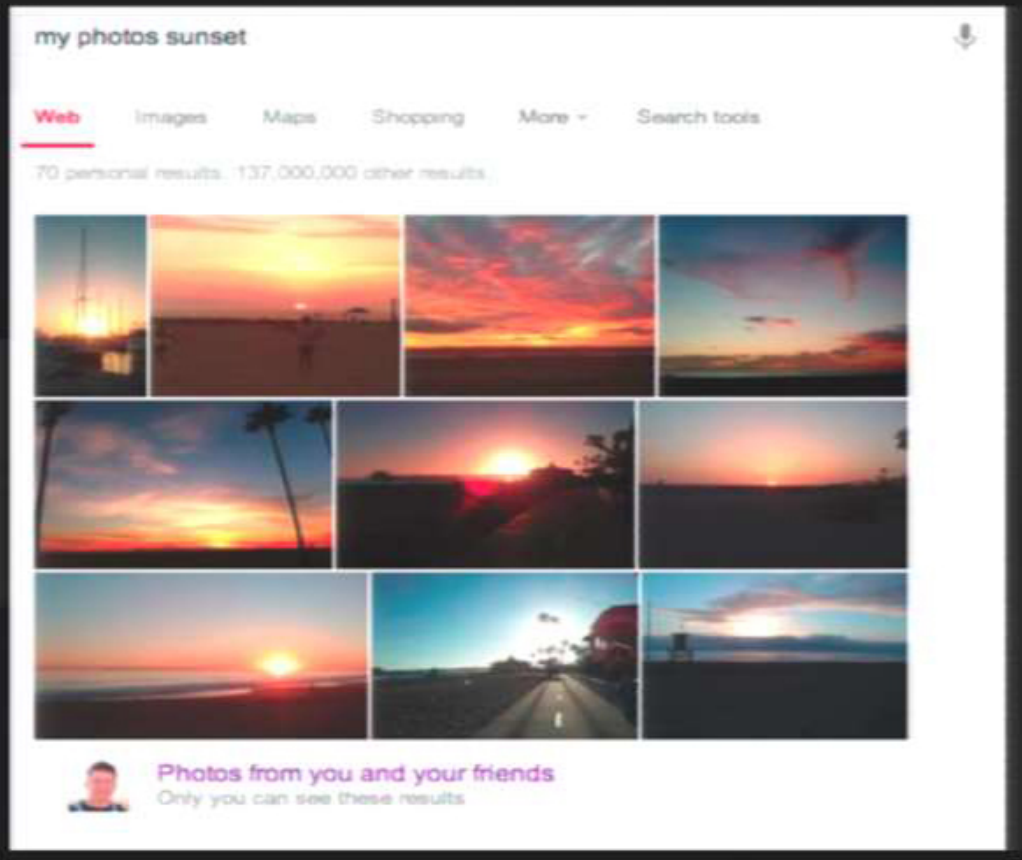
ImageNet Classification 2012

- Krizhevsky et al. -- 16.4% error (top-5)
- Next best (non-convnet) – 26.2% error



Commercial Deployment

- Google & Baidu, Spring 2013 for personal image search



Large Convnets for Image Classification

Large Convnets for Image Classification

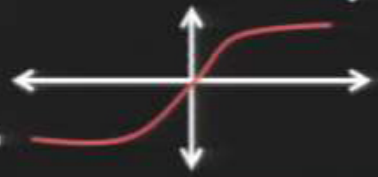
- Operations in each layer
- Architecture
- Training
- Results

Components of Each Layer

Pixels / Features

Filter with Dictionary (convolutional or tiled)

+ Non-linearity



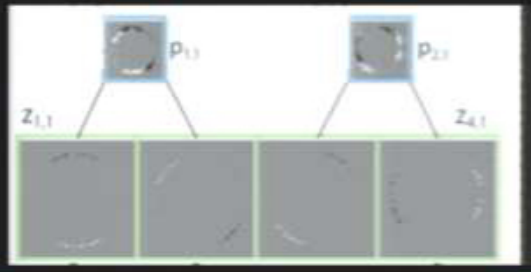
Components of Each Layer

Pixels / Features

Filter with Dictionary (convolutional or tiled)

+ Non-linearity

Spatial/Feature (Sum or Max)



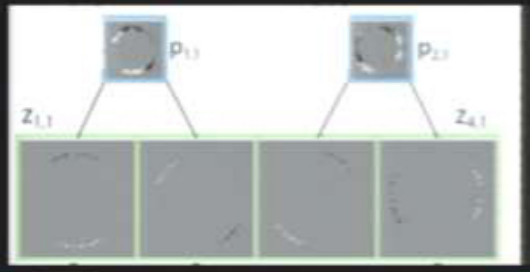
Components of Each Layer

Pixels / Features

Filter with Dictionary (convolutional or tiled)

+ Non-linearity

Spatial/Feature (Sum or Max)



[Optional]

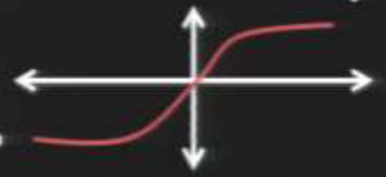
Normalization between feature responses

Components of Each Layer

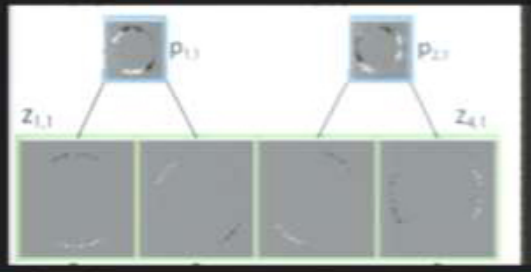
Pixels / Features

Filter with Dictionary (convolutional or tiled)

+ Non-linearity



Spatial/Feature (Sum or Max)



[Optional]

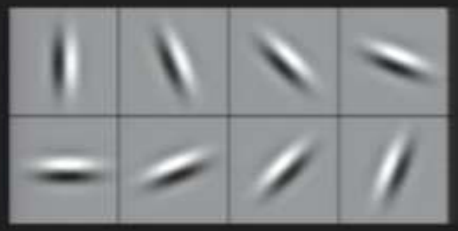
Normalization between feature responses

Output Features

Compare: SIFT Descriptor

Image Pixels

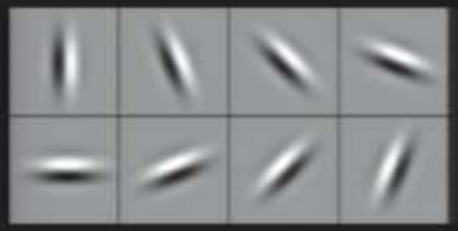
Apply Gabor filters



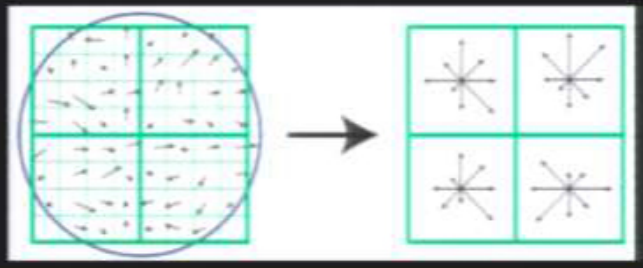
Compare: SIFT Descriptor

Image Pixels

Apply Gabor filters



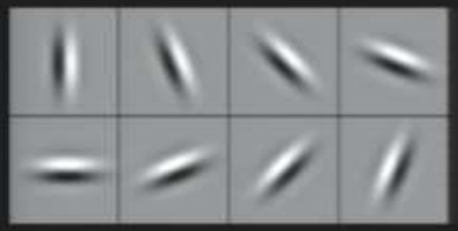
Spatial pool (Sum)



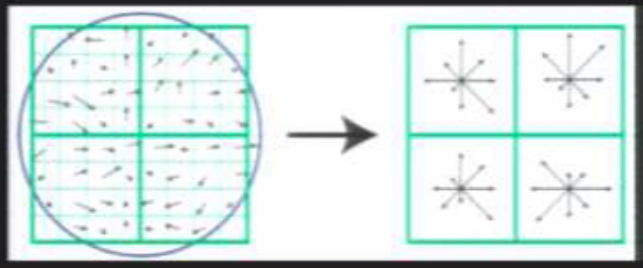
Compare: SIFT Descriptor

Image Pixels

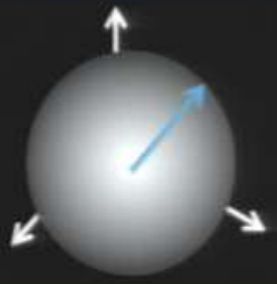
Apply Gabor filters



Spatial pool (Sum)



Normalize to unit length



Feature Vector

Compare: Spatial Pyramid Matching

SIFT Features

Filter with Visual Words

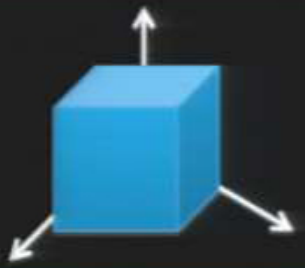
Lazebnik, Schmid, Ponce [CVPR 2006]

Compare: Spatial Pyramid Matching

SIFT Features

Filter with Visual Words

Max



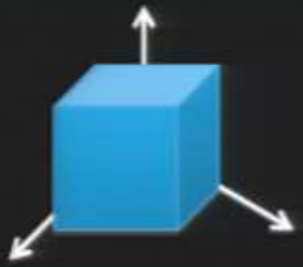
Lazebnik, Schmid, Ponce [CVPR 2006]

Compare: Spatial Pyramid Matching

SIFT Features

Filter with Visual Words

Max



Multi-scale spatial pool (Sum)

Classifier

Lazebnik, Schmid, Ponce [CVPR 2006]

Filtering

- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)

Input

Filtering

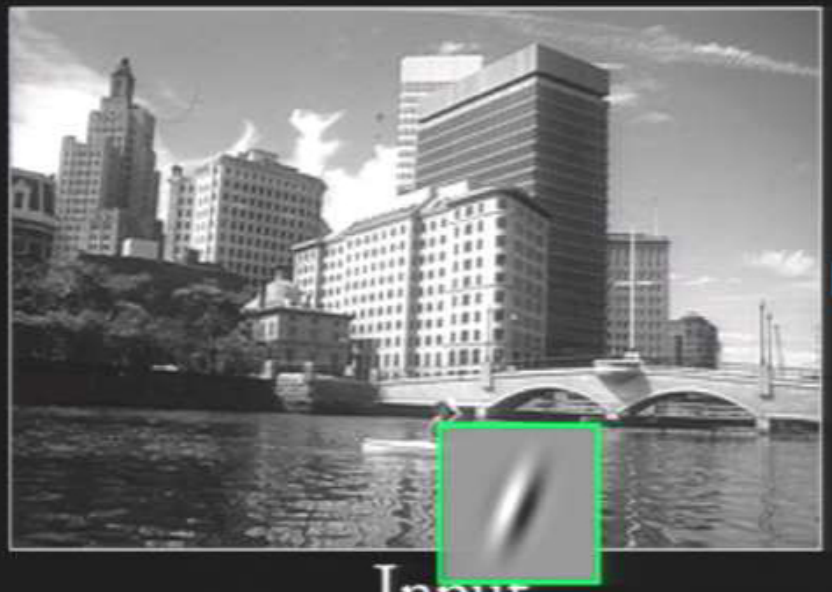
- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)

Input

Feature Map

Filtering

- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)



Input

Feature Map

Filtering

- Convolutional
  - Dependencies are local
  - Translation equivariance
  - Tied filter weights (few params)
  - Stride 1,2,… (faster, less mem.)

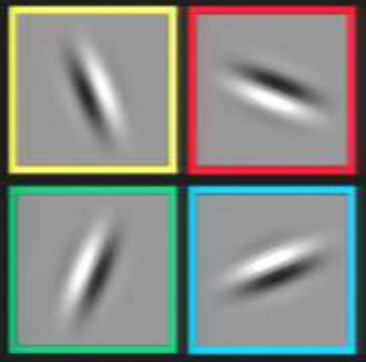
Input

Feature Map

Filtering

- Tiled
  - Filters repeat every n
  - More filters than convolution for given # features

Input

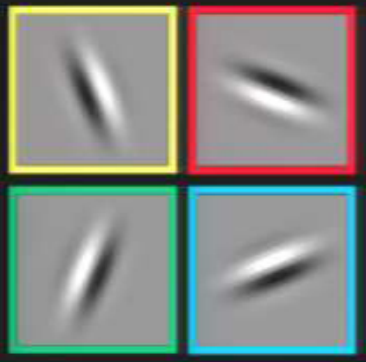


Filters

Filtering

- Tiled
  - Filters repeat every n
  - More filters than convolution for given # features

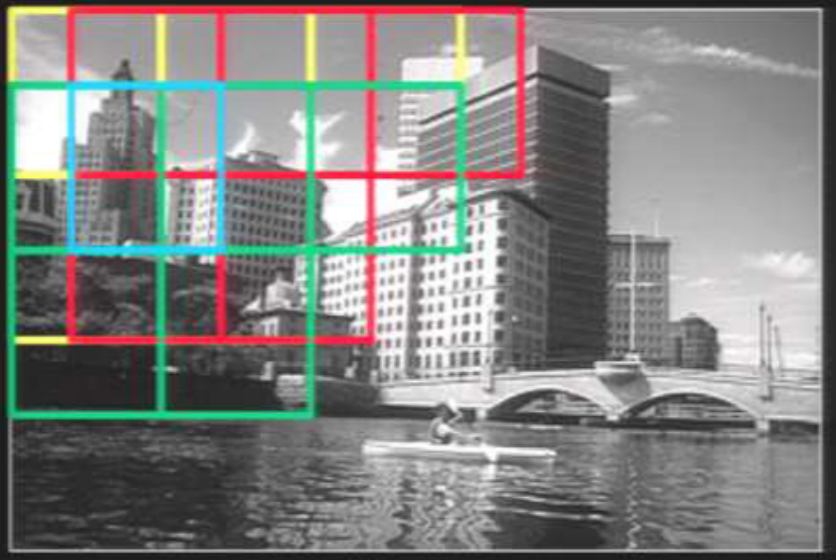
Input



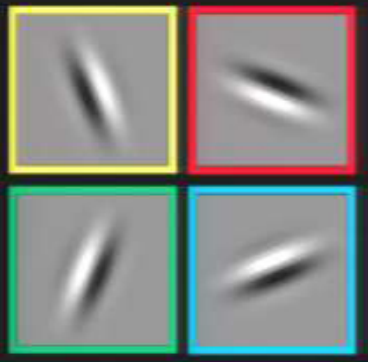
Filters

Filtering

- Tiled
  - Filters repeat every n
  - More filters than convolution for given # features



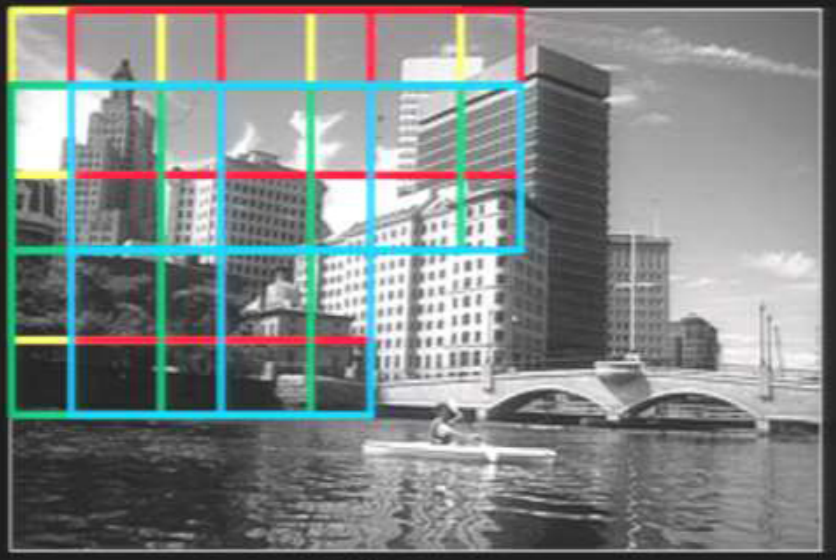
Input



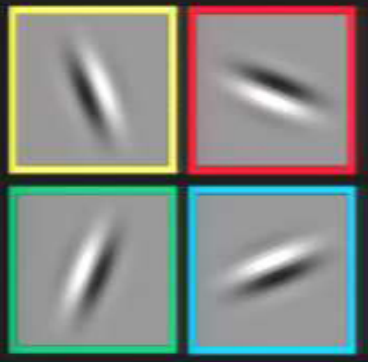
Filters

Filtering

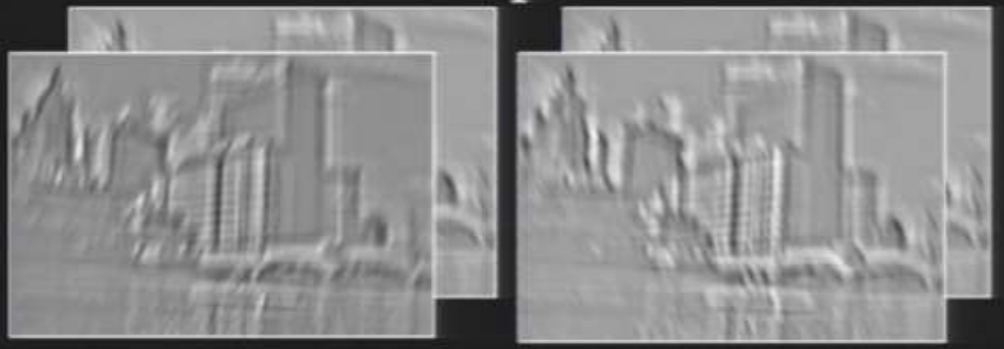
- Tiled
  - Filters repeat every n
  - More filters than convolution for given # features



Input



Filters

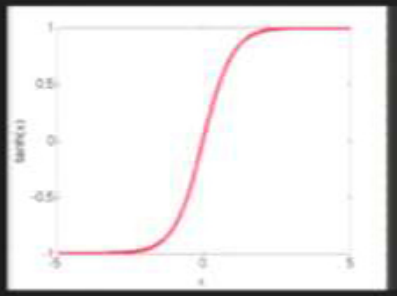


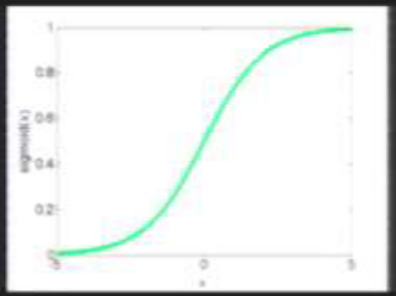
Feature maps

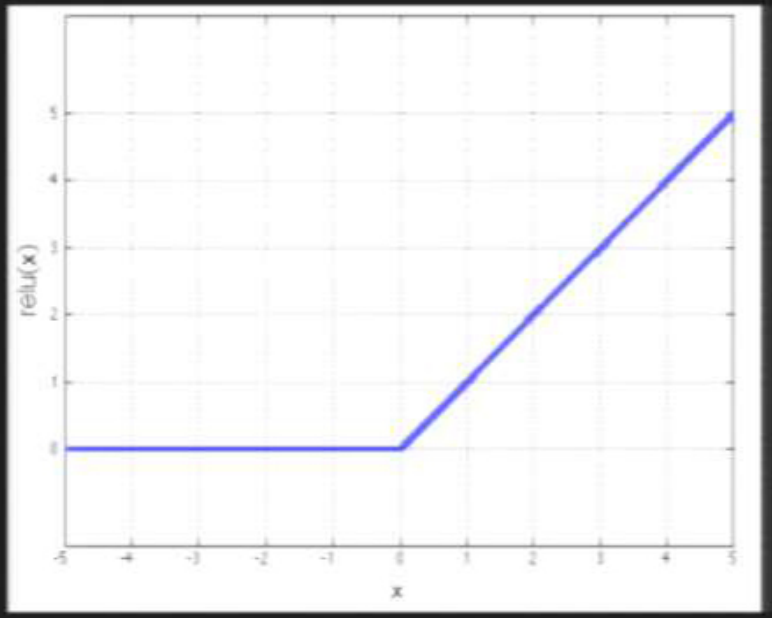
Non-Linearity

- Non-linearity
  - Per-feature independent
  - Tanh
  - Sigmoid: 1/(1+exp(-x))
  - Rectified linear
    - Simplifies backprop
    - Makes learning faster
    - Avoids saturation issues

  → Preferred option

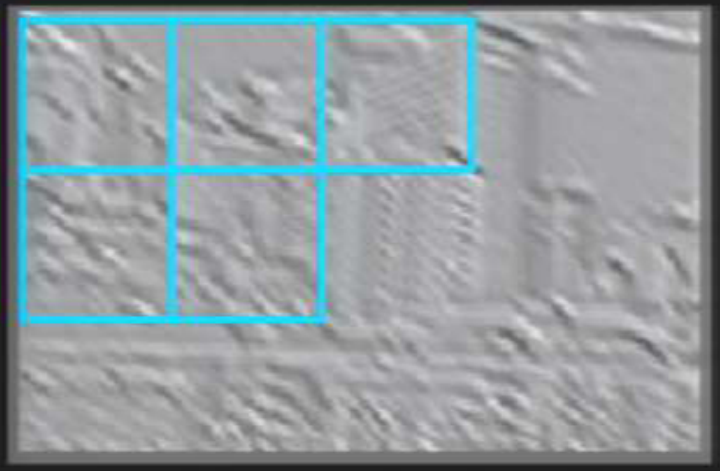






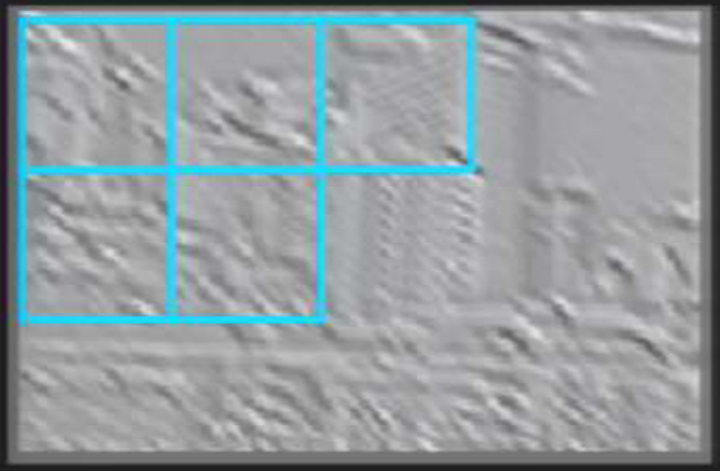
Pooling

- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis



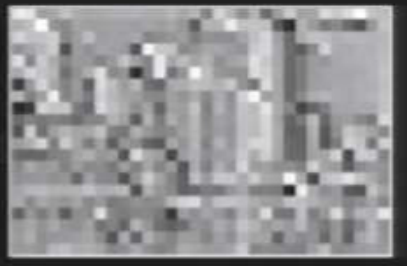
Pooling

- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis



Max

Sum

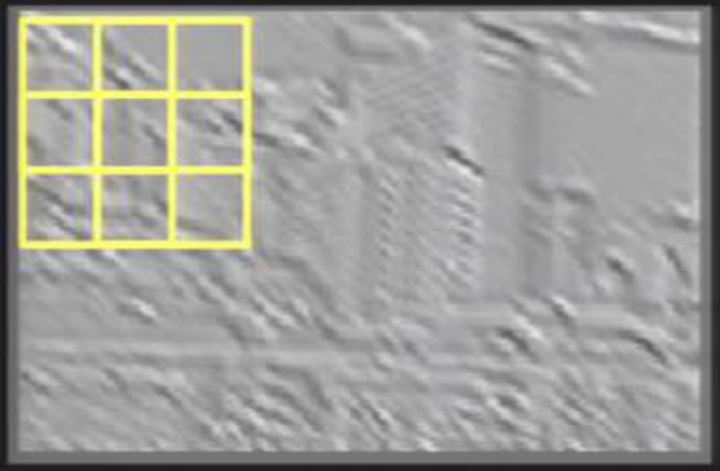


Pooling

- Pooling across feature groups
  - Additional form of inter-feature competition
  - MaxOut Networks [Goodfellow et al. ICML 2013]

Pooling

- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis



Max

Sum

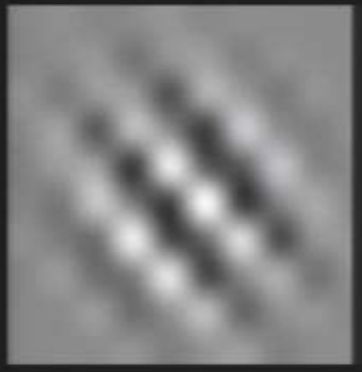
Pooling

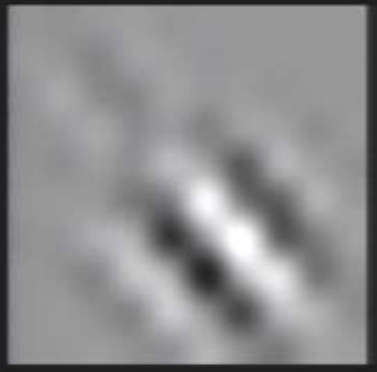
- Pooling across feature groups
  - Additional form of inter-feature competition
  - MaxOut Networks [Goodfellow et al. ICML 2013]

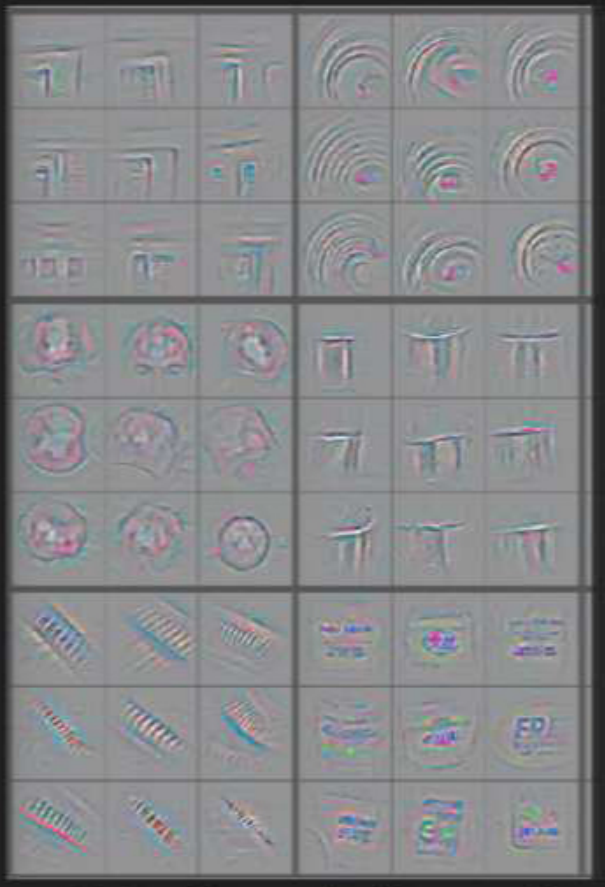
Role of Pooling

- Spatial pooling
  - Invariance to small transformations
  - Larger receptive fields (see more of input)

Visualization technique from [Le et al. NIPS'10]:







Zeiler, Fergus [arXiv 2013]

Videos from: http://ai.stanford.edu/~quocle/TCNNweb

Normalization

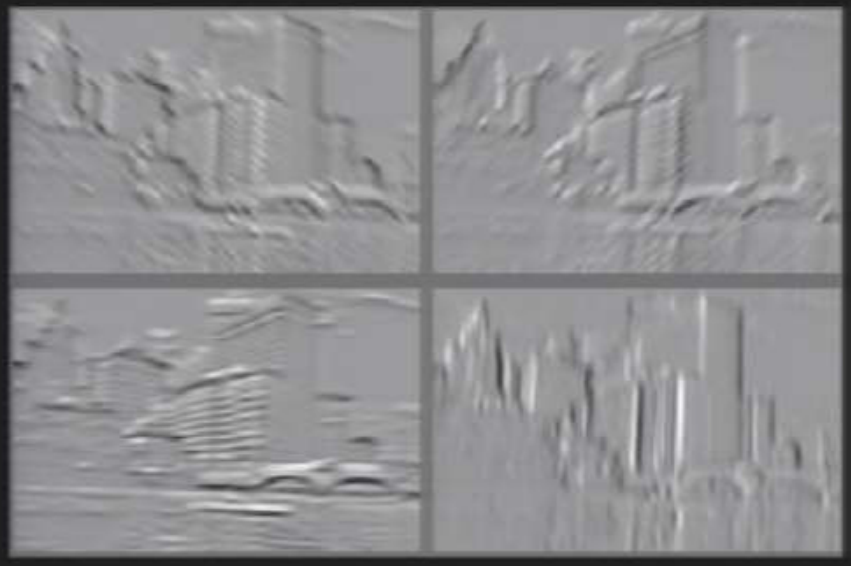
- Contrast normalization
  - See Divisive Normalization in Neuroscience

Input

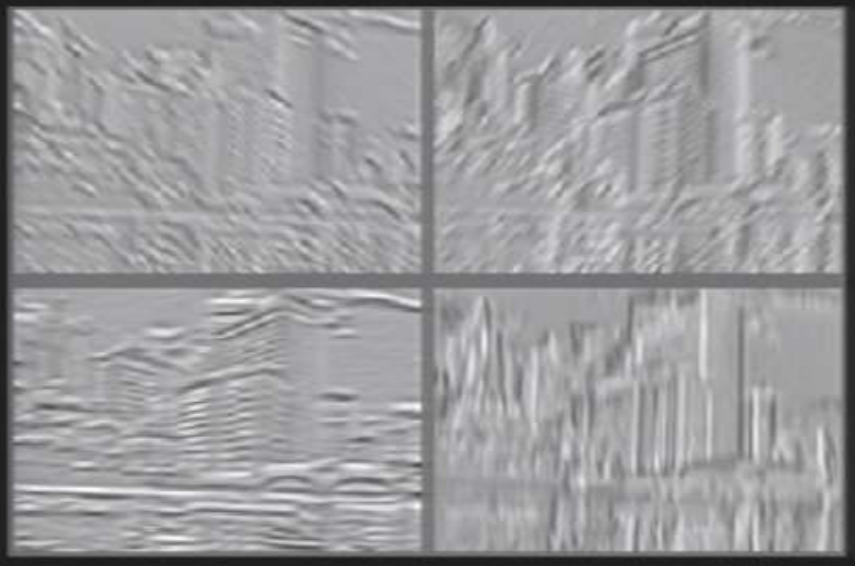
Filters

Normalization

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



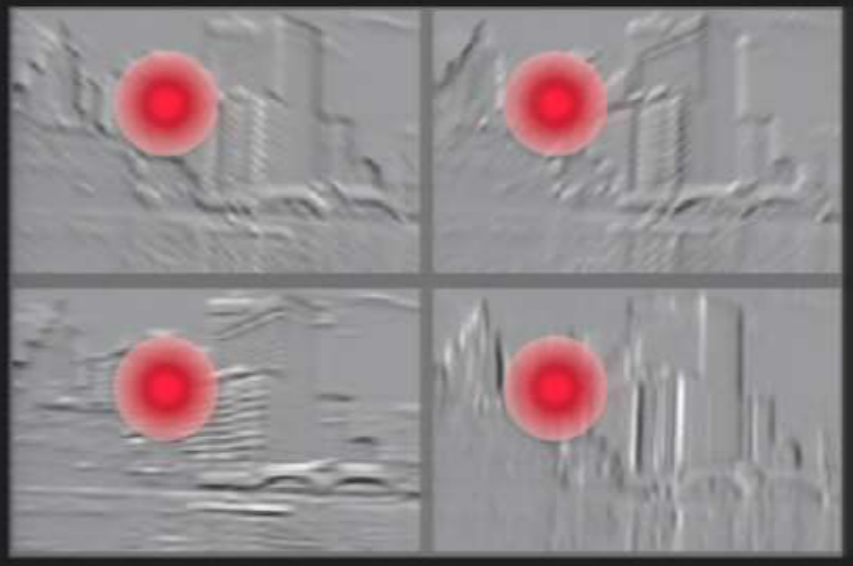
Feature Maps



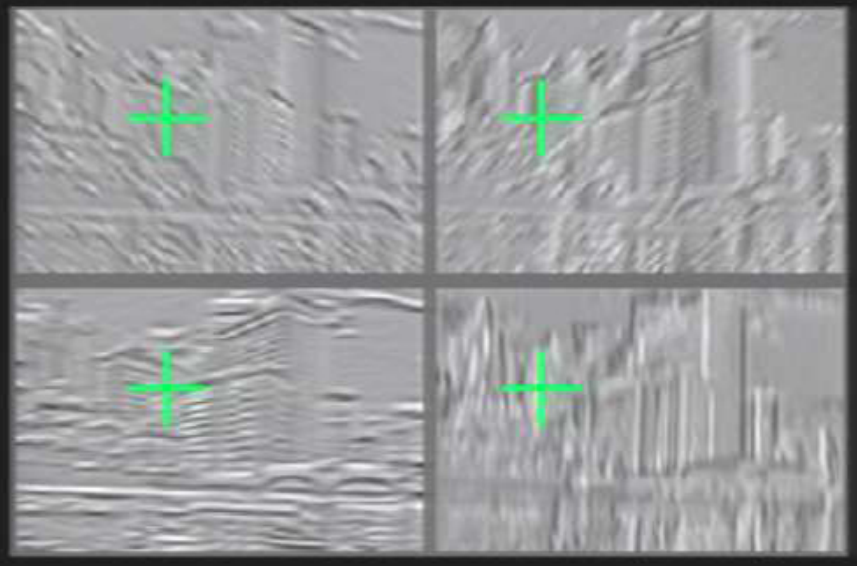
Feature Maps
After Contrast Normalization

Normalization

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



Feature Maps



Feature Maps
After Contrast Normalization

Normalization

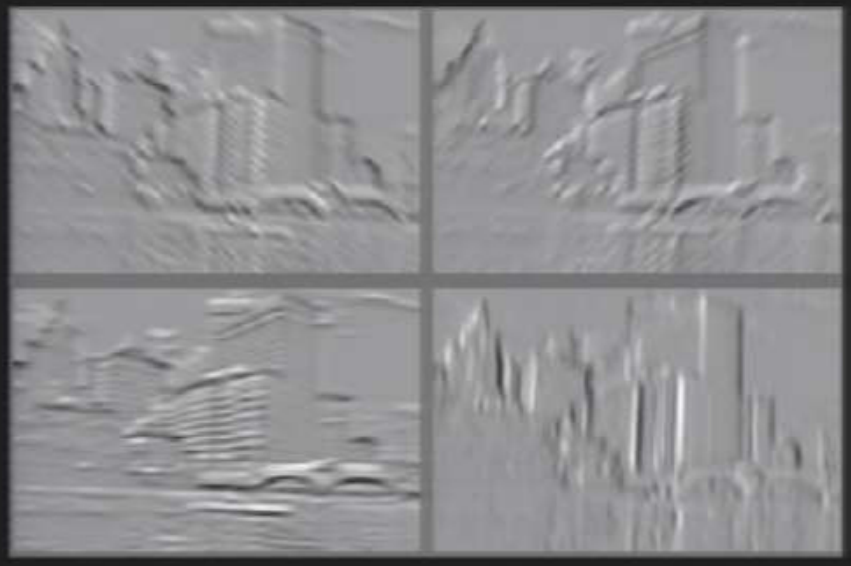
- Contrast normalization
  - See Divisive Normalization in Neuroscience

Input

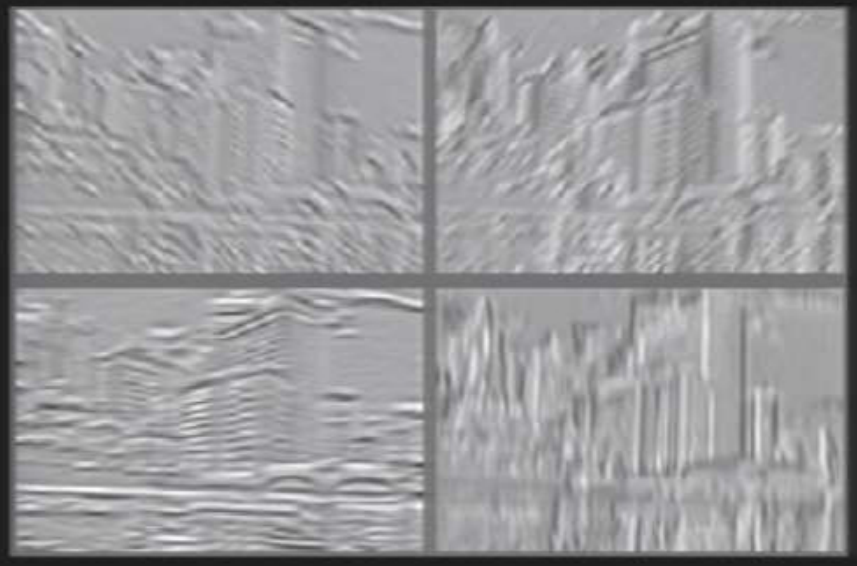
Filters

Normalization

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



Feature Maps



Feature Maps
After Contrast Normalization

Normalization

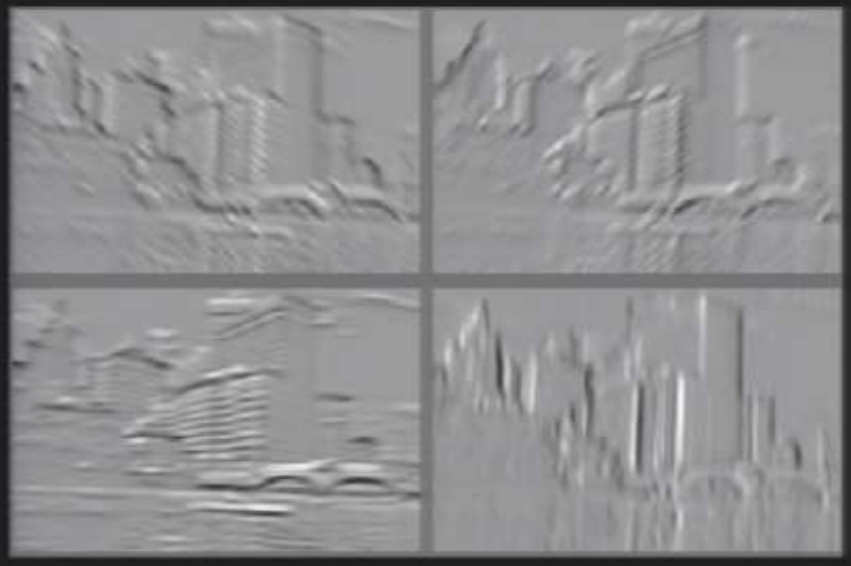
- Contrast normalization
  - See Divisive Normalization in Neuroscience

Input

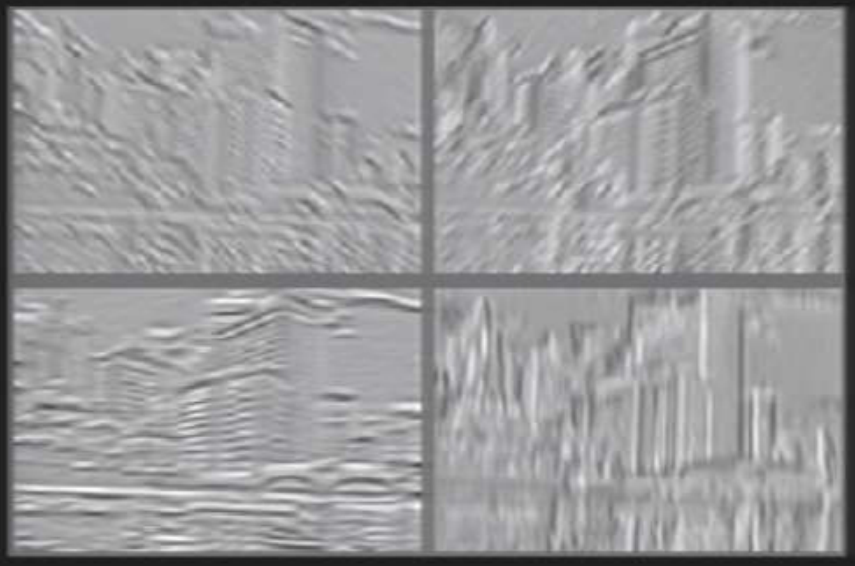
Filters

Normalization

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



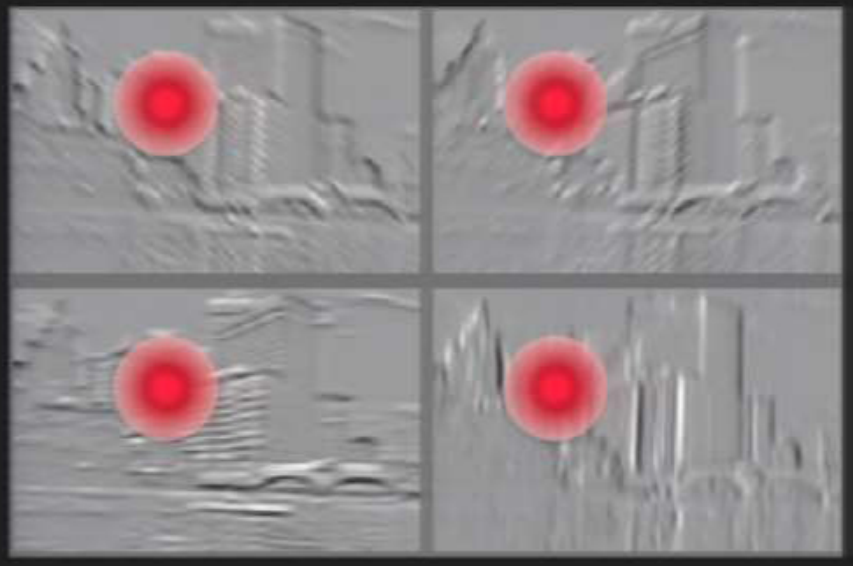
Feature Maps



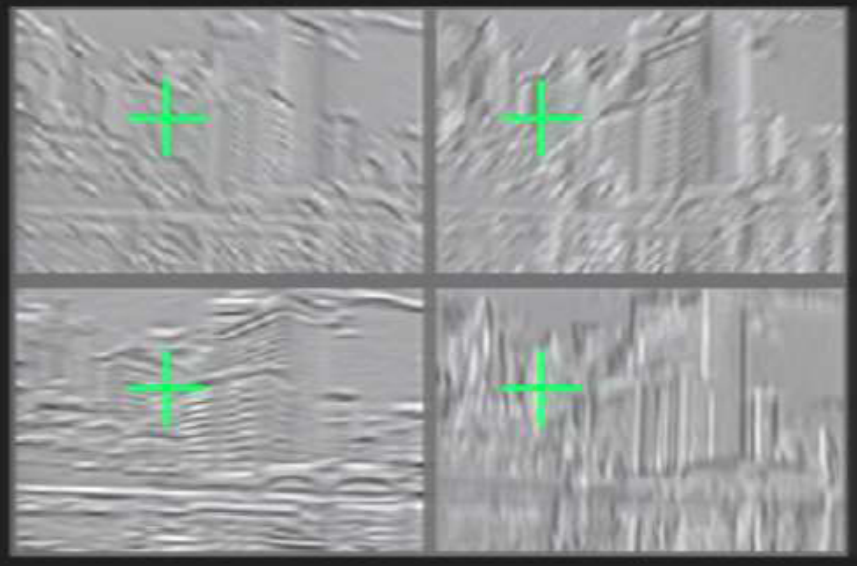
Feature Maps
After Contrast Normalization

Normalization

- Contrast normalization (between/across feature maps)
  - Local mean = 0, local std. = 1, "Local" → 7x7 Gaussian
  - Equalizes the features maps



Feature Maps



Feature Maps
After Contrast Normalization

Role of Normalization

- Introduces local competition between features
  - Poor man's version of "Explaining away" in graphical models
  - Just like top-down models
  - But more local mechanism

- Also helps to scale activations at each layer better for learning
  - Makes energy surface more isotropic
  - So each gradient step makes more progress

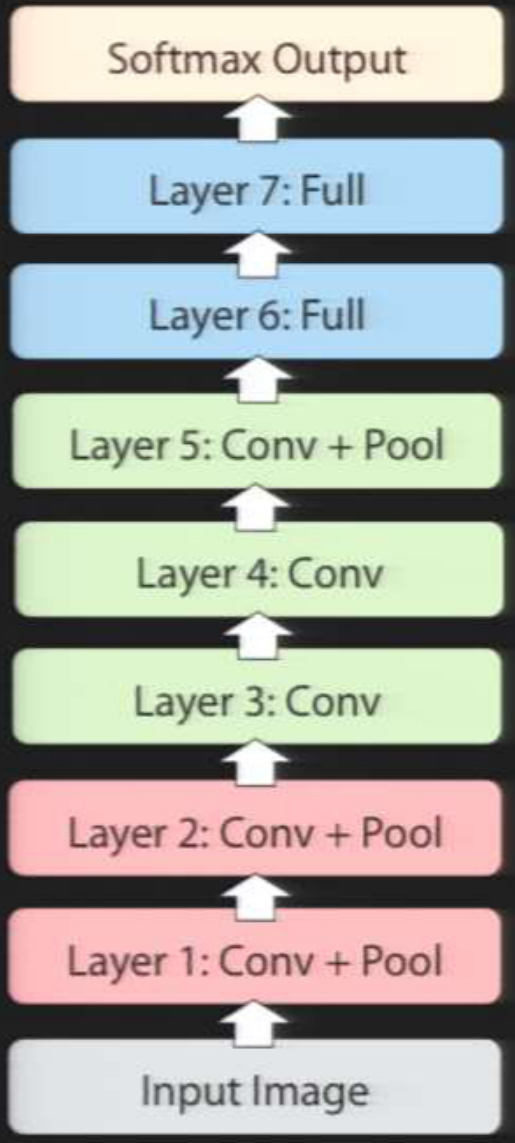
- Empirically, seems to help a bit (1-2%) on ImageNet
  - More on other datasets (see [Jarrett et al. ICCV'09] for interesting analysis)

Architecture

Importance of Depth

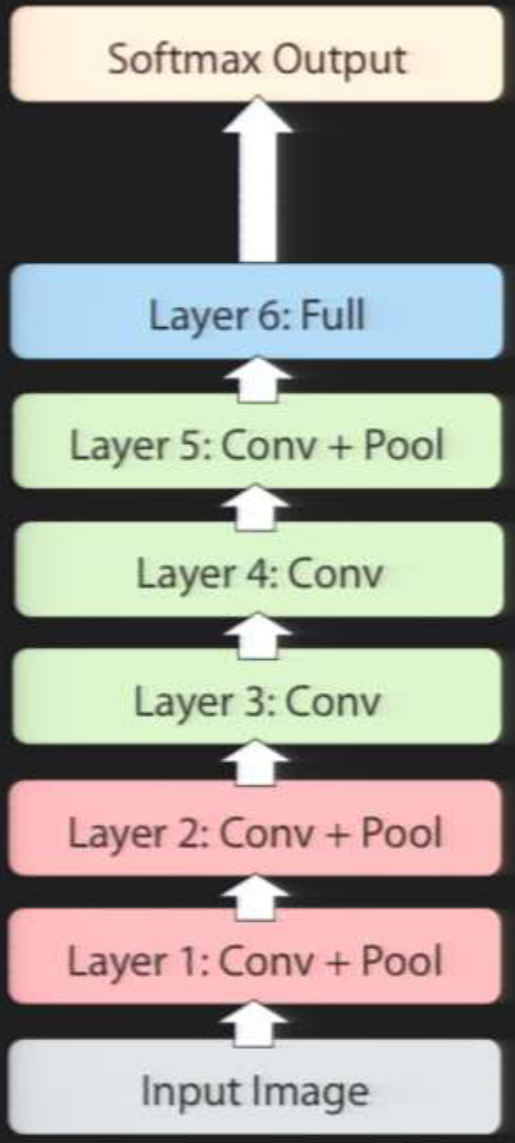
Architecture of Krizhevsky et al.

- 8 layers total
- Trained on Imagenet dataset [Deng et al. CVPR'09]
- 18.2% top-5 error
- Our reimplementation:
  18.1% top-5 error



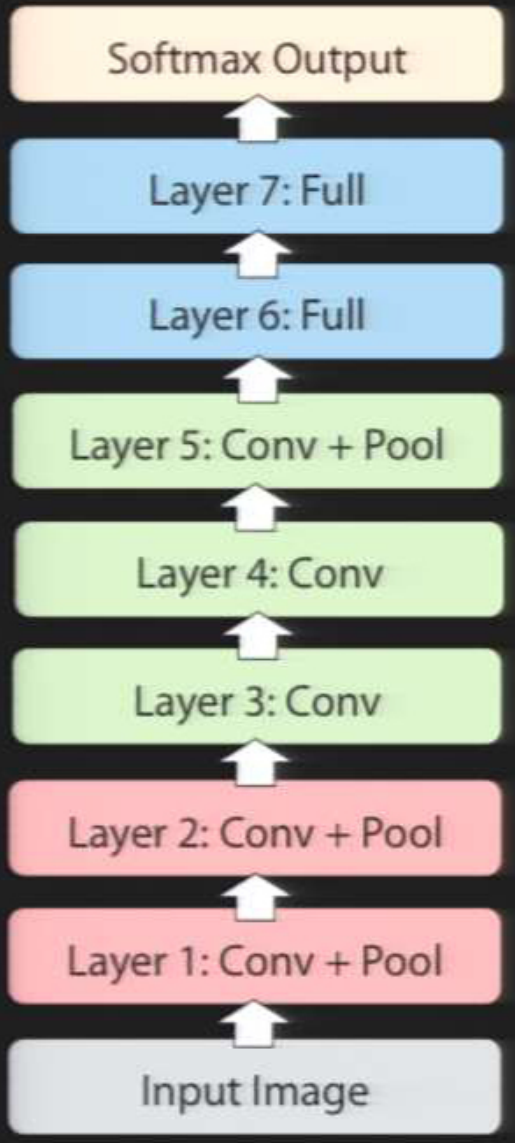
Architecture of Krizhevsky et al.

- Remove top fully connected layer
  - Layer 7
- Drop 16 million parameters
- Only 1.1% drop in performance!



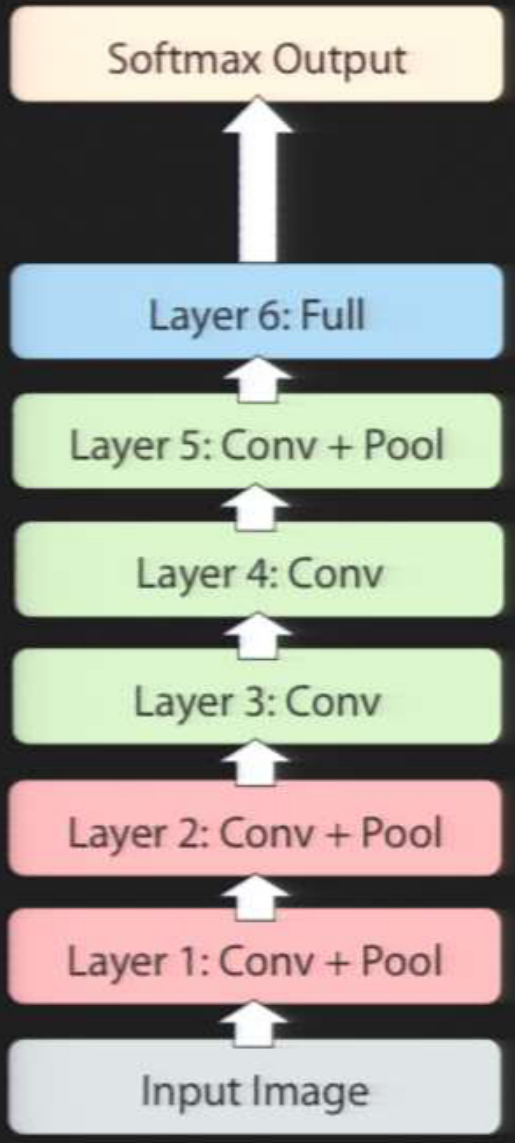
Architecture of Krizhevsky et al.

- 8 layers total
- Trained on Imagenet dataset [Deng et al. CVPR'09]
- 18.2% top-5 error
- Our reimplementation:
  18.1% top-5 error



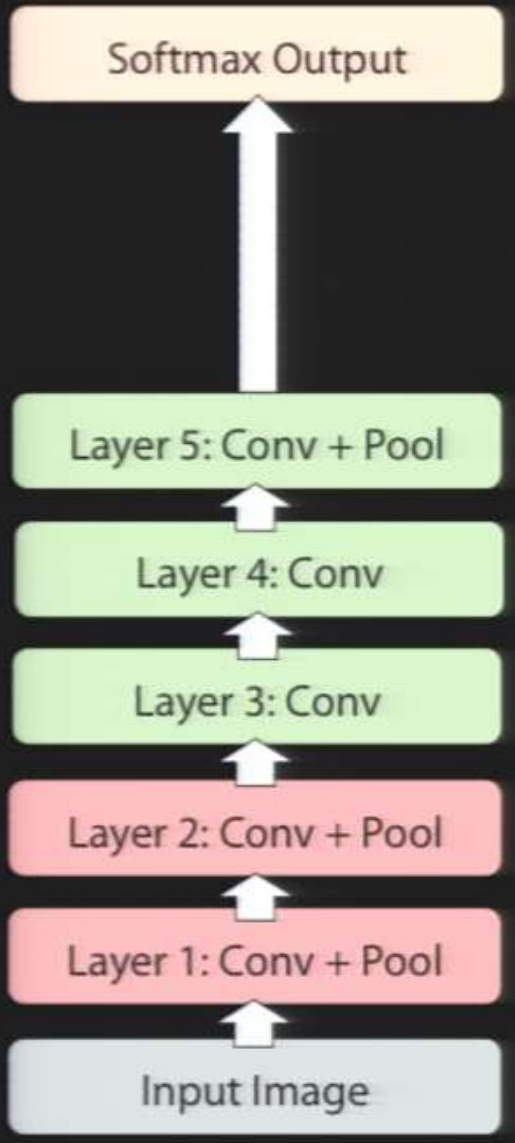
Architecture of Krizhevsky et al.

- Remove top fully connected layer
  - Layer 7
- Drop 16 million parameters
- Only 1.1% drop in performance!



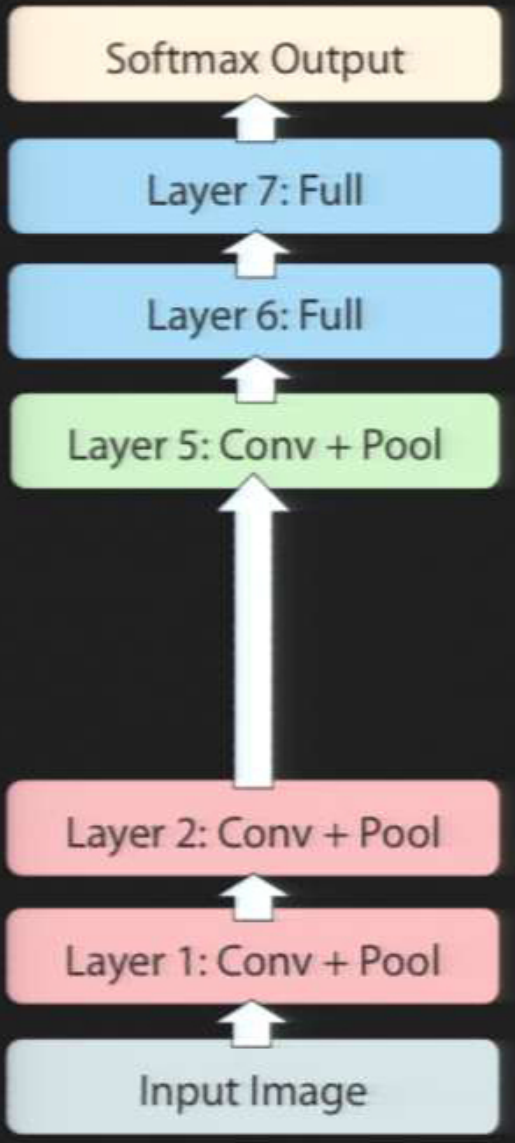
Architecture of Krizhevsky et al.

- Remove both fully connected layers
  - Layer 6 & 7
- Drop ~50 million parameters
- 5.7% drop in performance



Architecture of Krizhevsky et al.

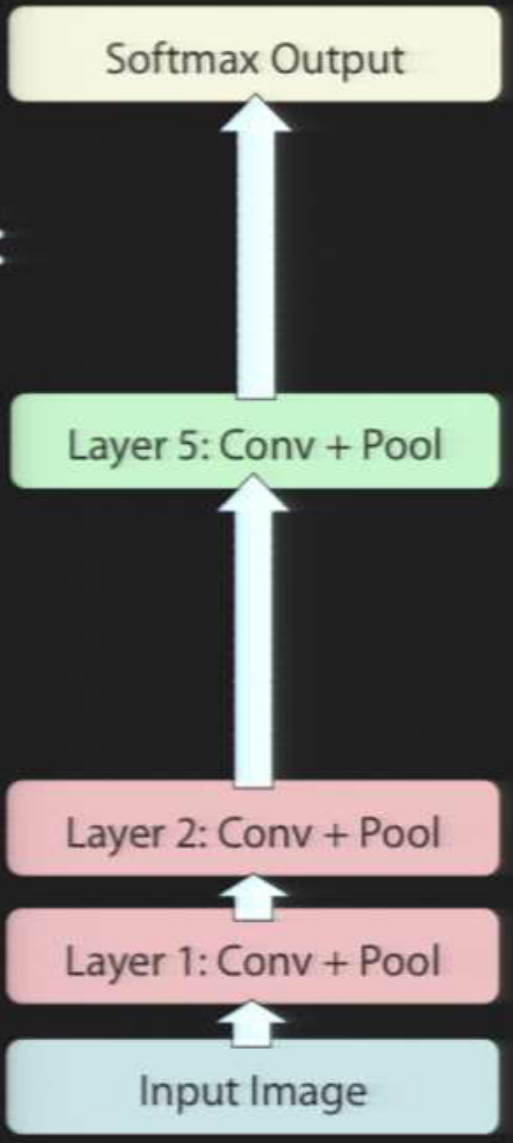
- Now try removing upper feature extractor layers:
  - Layers 3 & 4
- Drop ~1 million parameters
- 3.0% drop in performance



Architecture of Krizhevsky et al.

- Now try removing upper feature extractor layers & fully connected:
  - Layers 3, 4, 6 ,7
- Now only 4 layers
- 33.5% drop in performance

→Depth of network is key



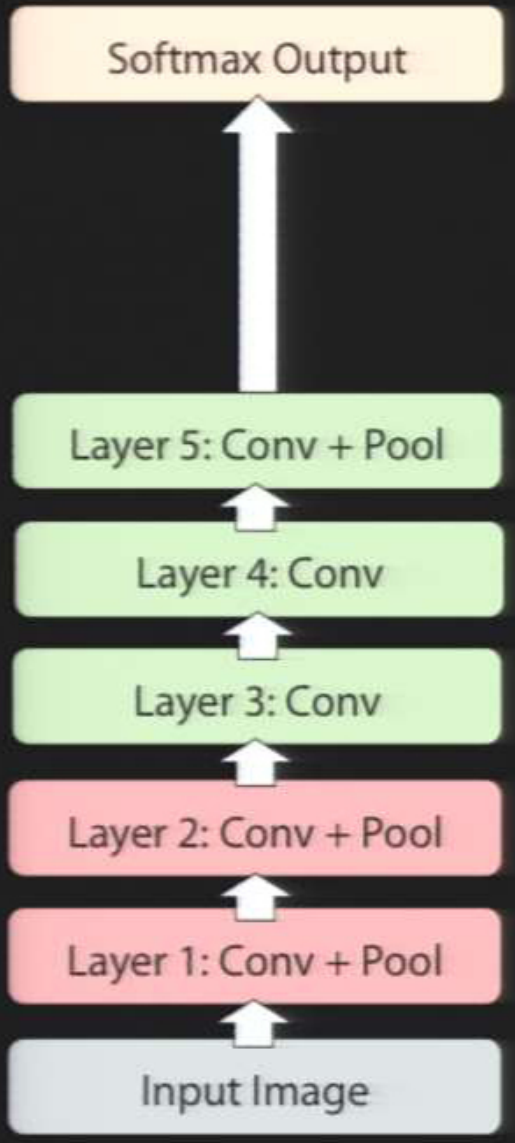
Tapping off Features at each Layer

Plug features from each layer into linear SVM or soft-max

	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

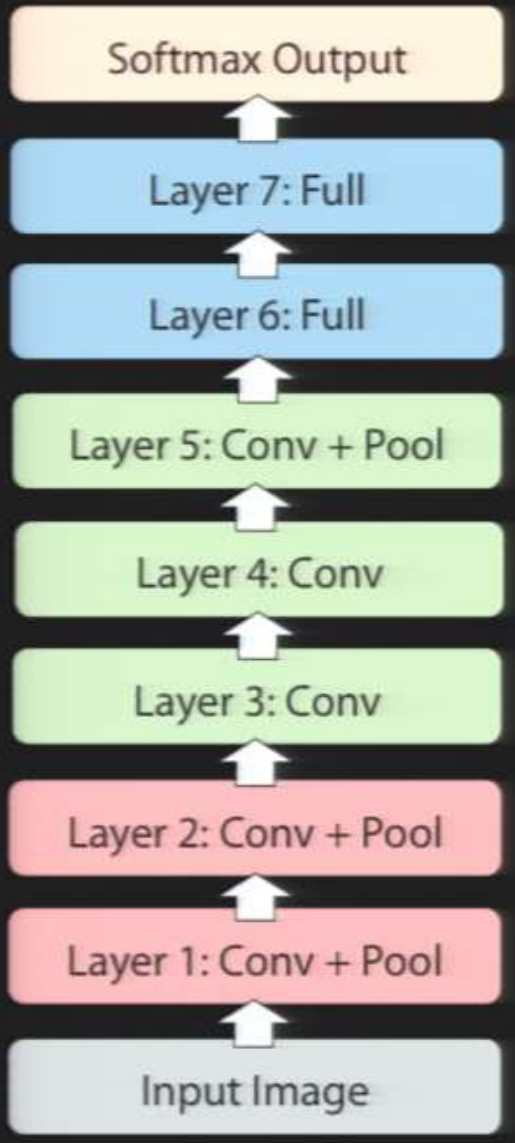
Architecture of Krizhevsky et al.

- Remove both fully connected layers
  - Layer 6 & 7
- Drop ~50 million parameters
- 5.7% drop in performance



Architecture of Krizhevsky et al.

- 8 layers total
- Trained on Imagenet dataset [Deng et al. CVPR'09]
- 18.2% top-5 error
- Our reimplementation:
  18.1% top-5 error



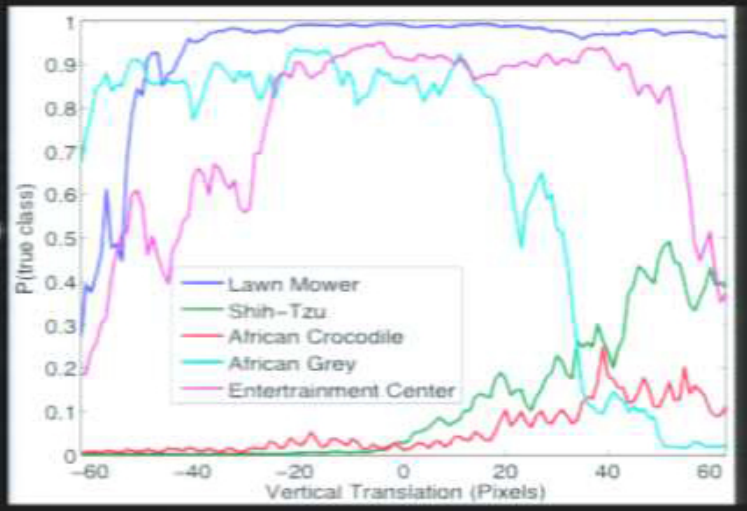
Tapping off Features at each Layer

Plug features from each layer into linear SVM or soft-max

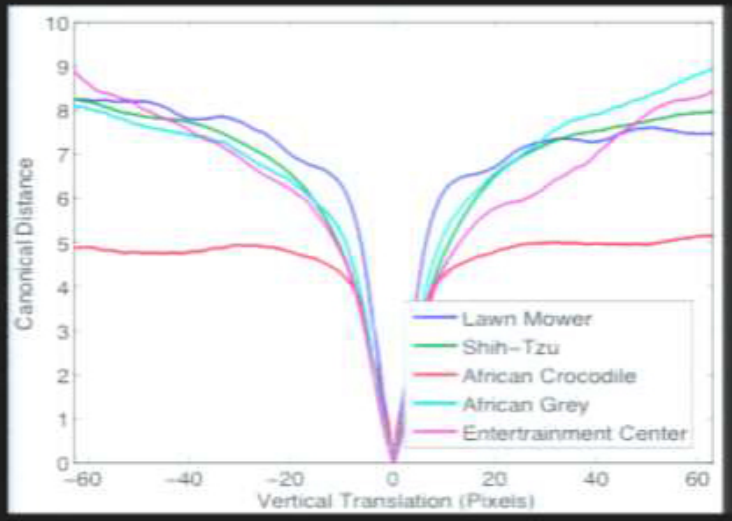
	Cal-101 (30/class)	Cal-256 (60/class)
SVM (1)	44.8 ± 0.7	24.6 ± 0.4
SVM (2)	66.2 ± 0.5	39.6 ± 0.3
SVM (3)	72.3 ± 0.4	46.0 ± 0.3
SVM (4)	76.6 ± 0.4	51.3 ± 0.1
SVM (5)	$\mathbf{86.2 \pm 0.8}$	65.6 ± 0.3
SVM (7)	$\mathbf{85.5 \pm 0.4}$	$\mathbf{71.7 \pm 0.2}$
Softmax (5)	82.9 ± 0.4	65.7 ± 0.5
Softmax (7)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

Translation (Vertical)

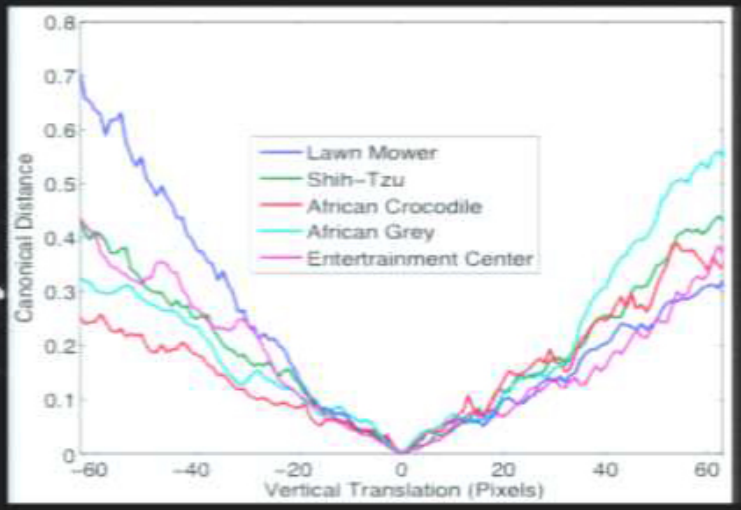
Output



Layer 1

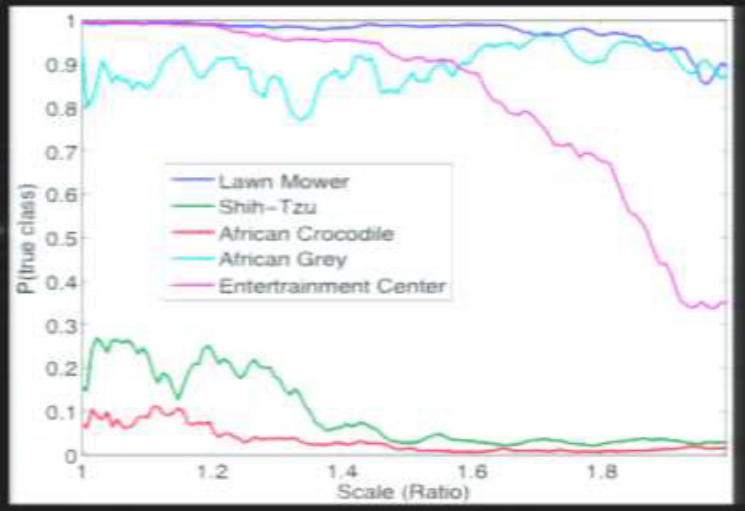


Layer 7

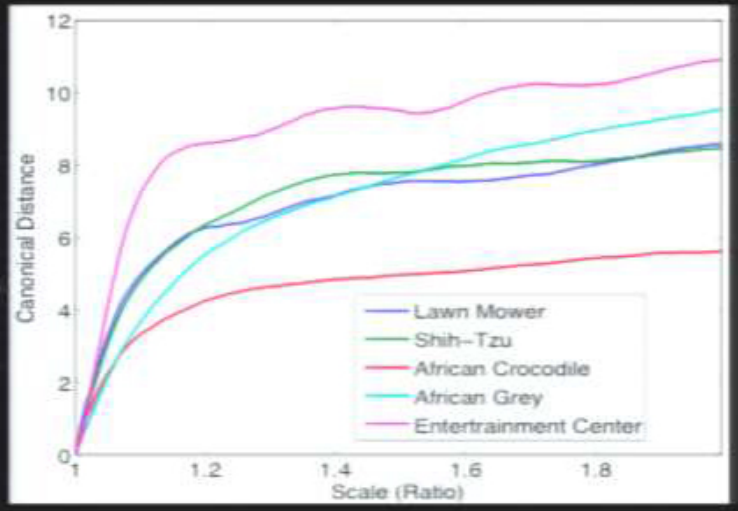


Scale Invariance

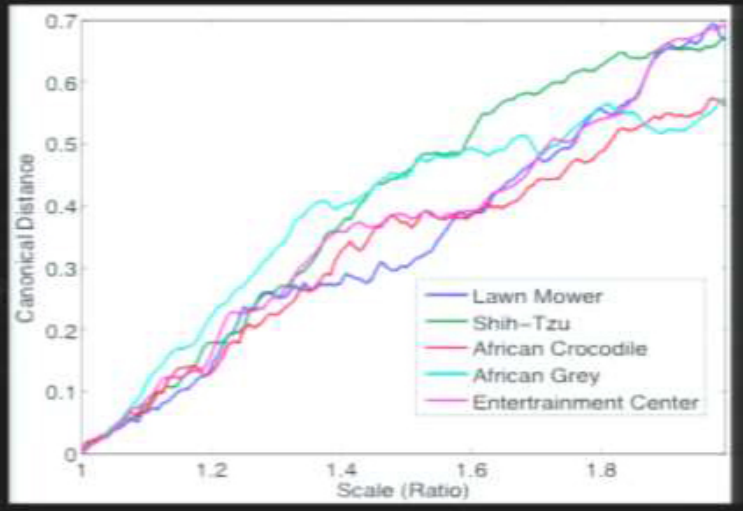
Output



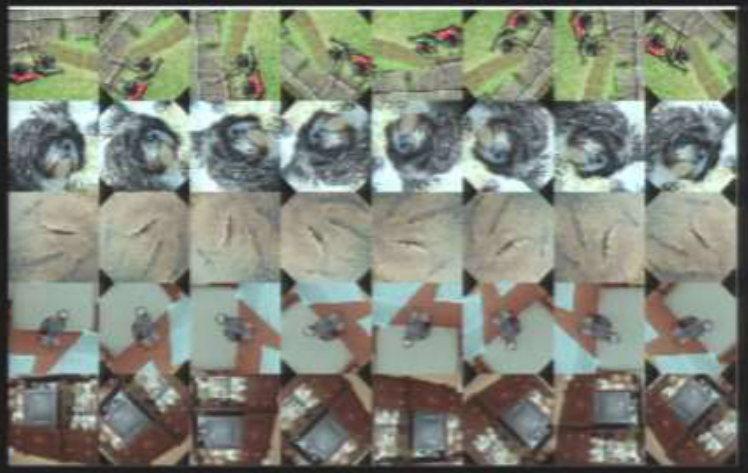
Layer 1



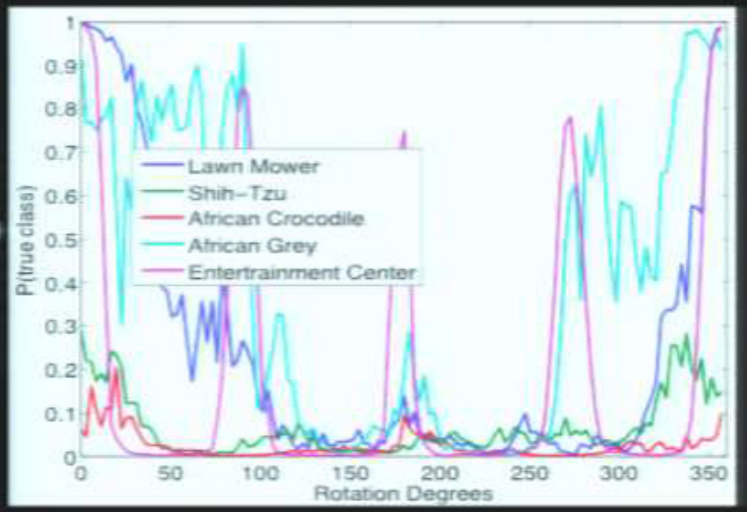
Layer 7



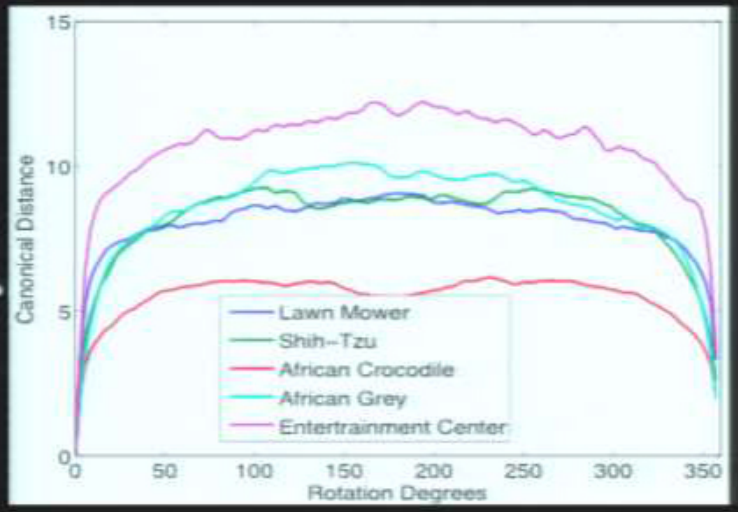
Rotation Invariance



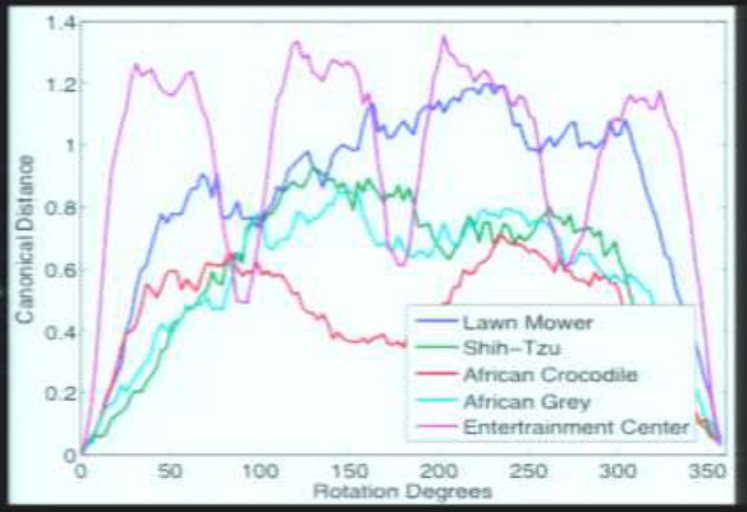
Output



Layer 1



Layer 7



Visualizing ConvNets

Visualizing Convnets

- Raw coefficients of learned filters in higher layers difficult to interpret

- Several approaches look to optimize input to maximize activity in a high-level feature
  - Erhan et al. [Tech Report 2009]
  - Le et al. [NIPS 2010]
  - Depend on initialization
  - Model invariance with Hessian about (locally) optimal stimulus

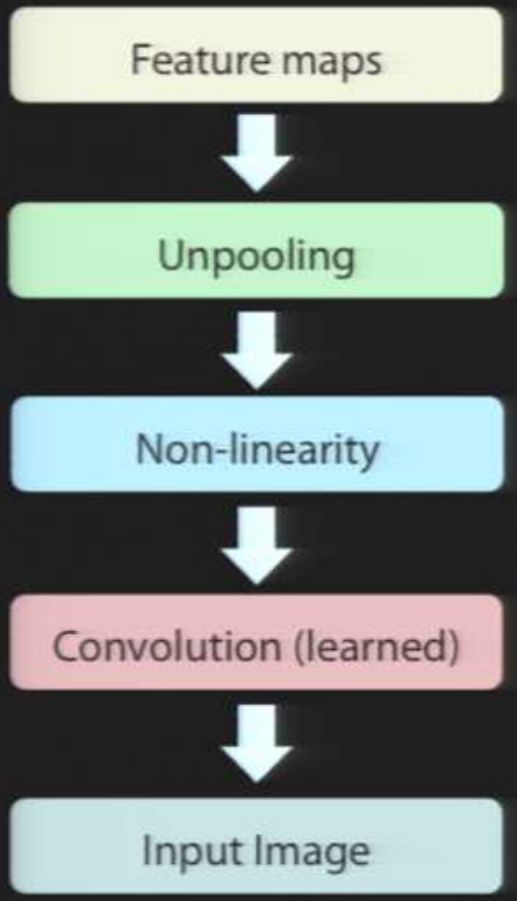
Visualization using Deconvolutional Networks

[Zeiler et al. CVPR'10, ICCV'11, arXiv'13]

- Provides way to map activations at high layers back to the input

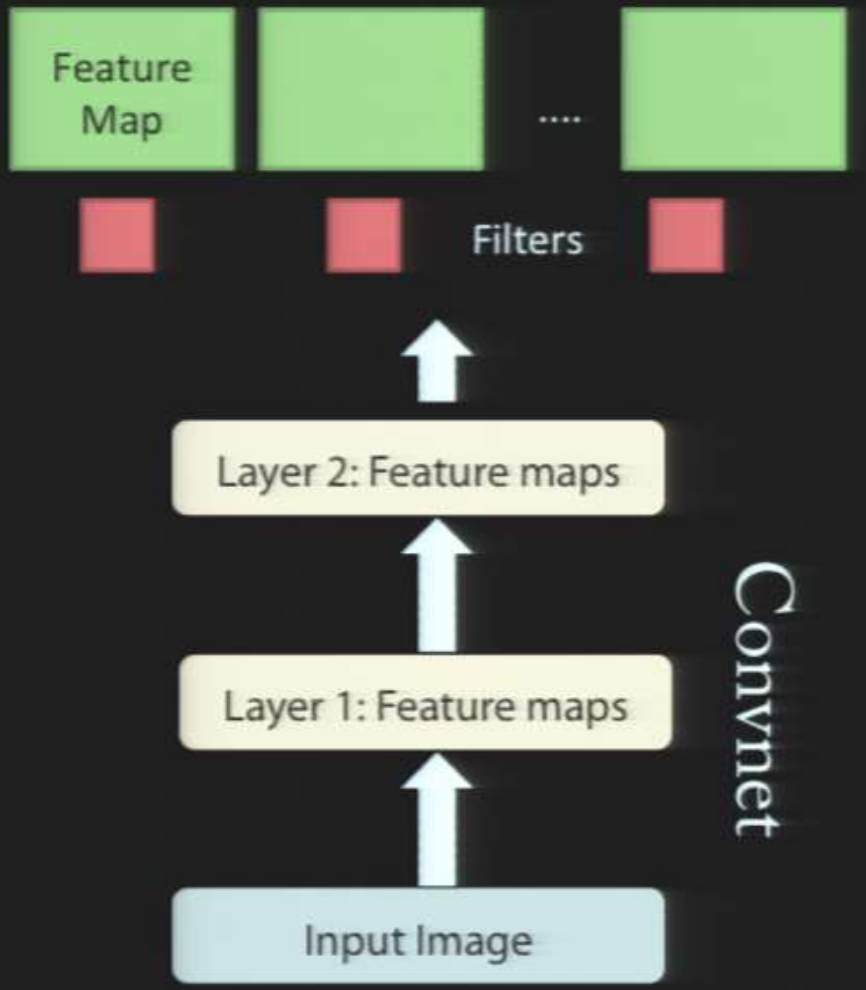
- Same operations as Convnet, but in reverse:
  - Unpool feature maps
  - Convolve unpooled maps
    - Filters copied from Convnet

- Used here purely as a probe
  - Originally proposed as unsupervised learning method
  - No inference, no learning



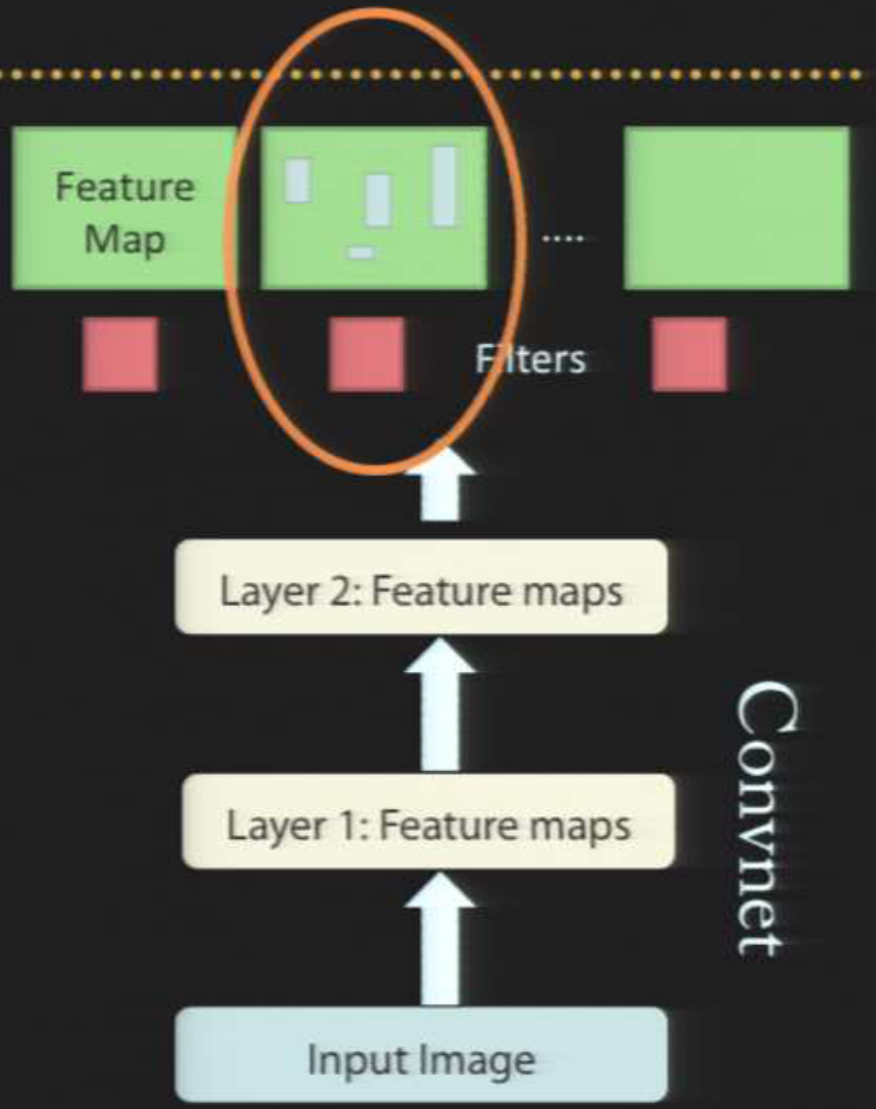
Deconvnet Projection from Higher Layers

[Zeiler and Fergus. arXiv'13]



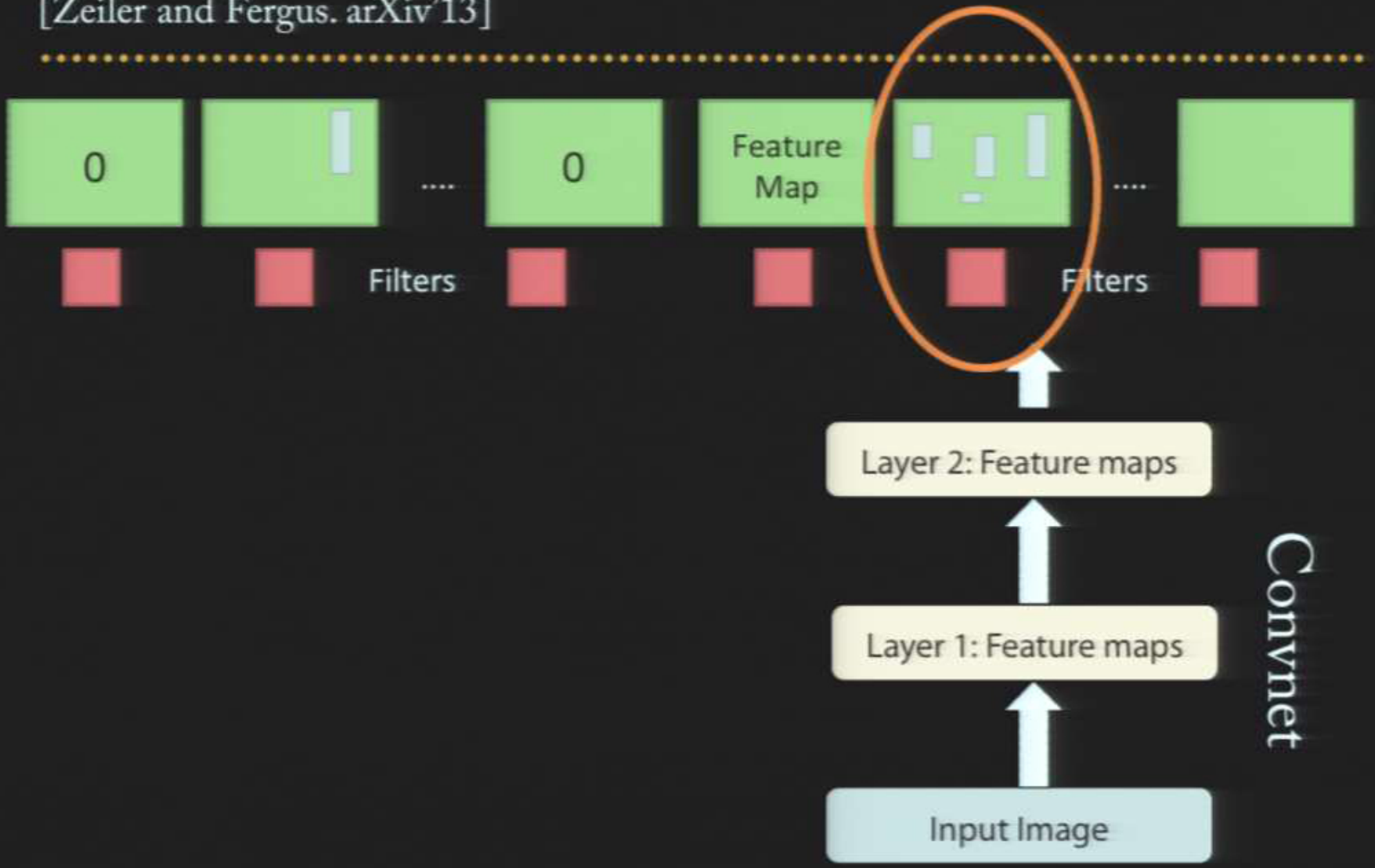
Deconvnet Projection from Higher Layers

[Zeiler and Fergus. arXiv'13]



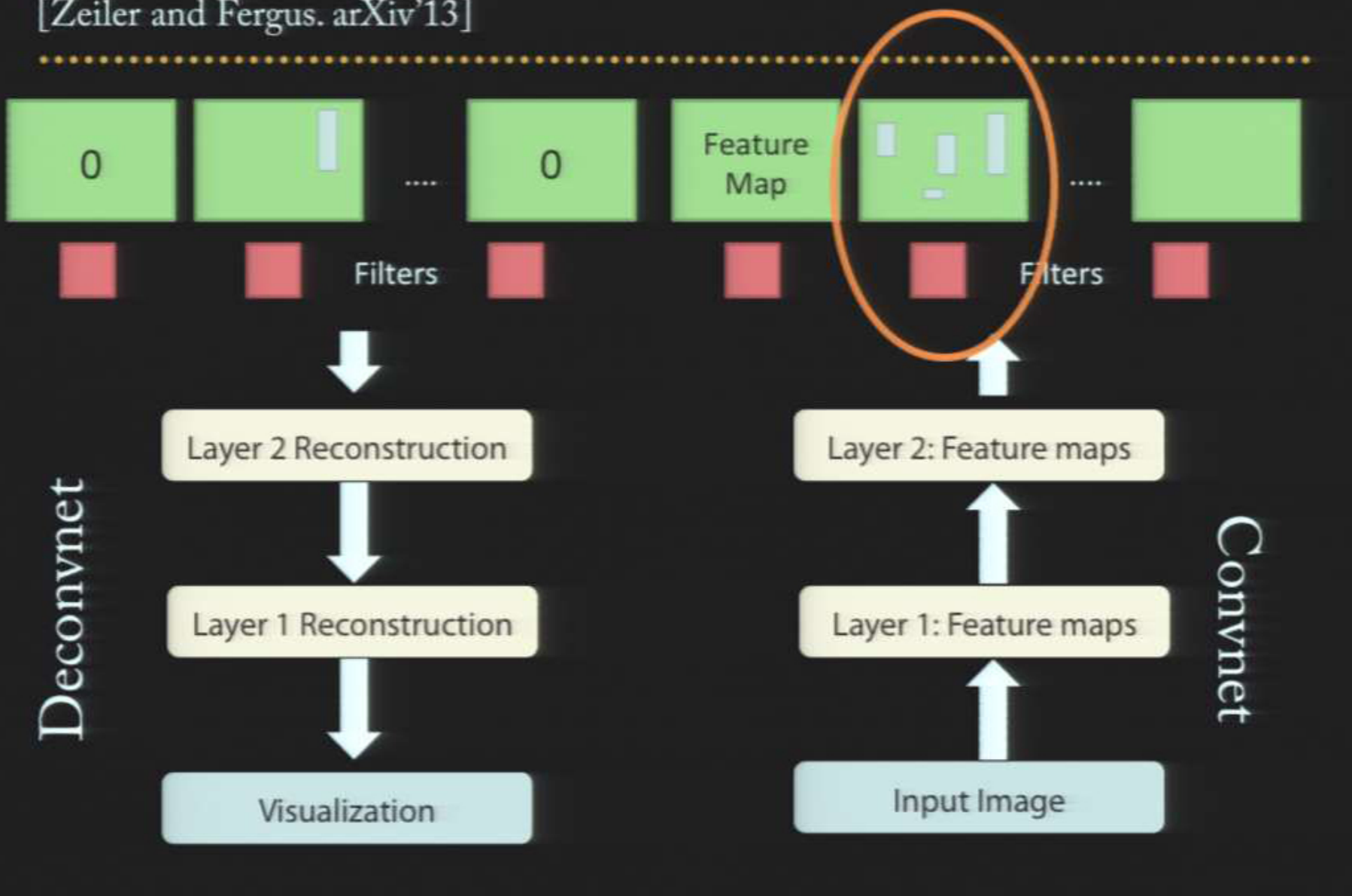
Deconvnet Projection from Higher Layers

[Zeiler and Fergus. arXiv'13]



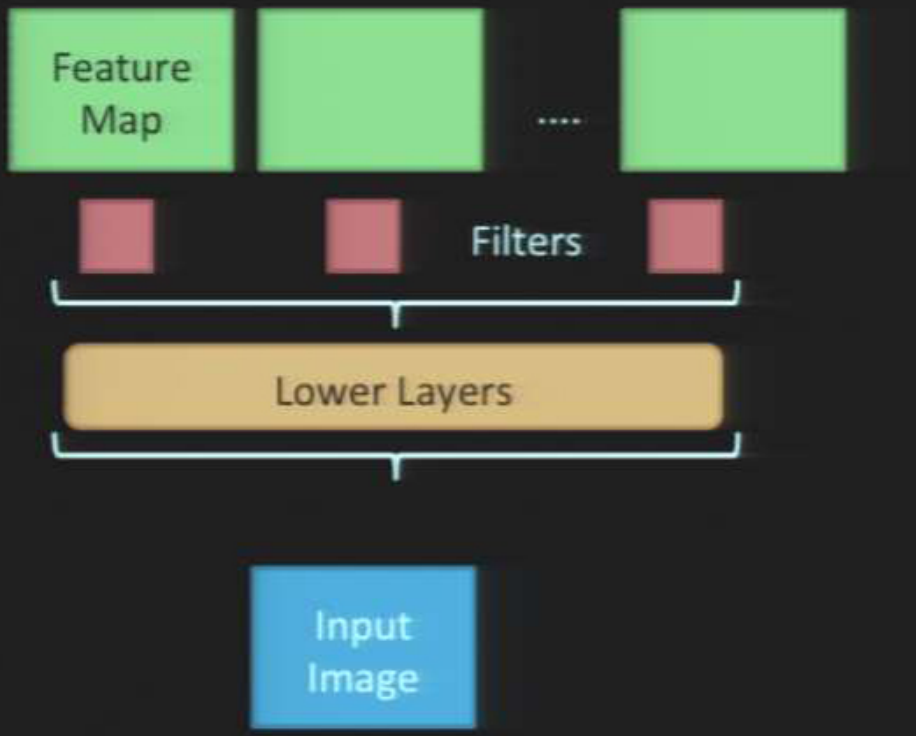
Deconvnet Projection from Higher Layers

[Zeiler and Fergus. arXiv'13]

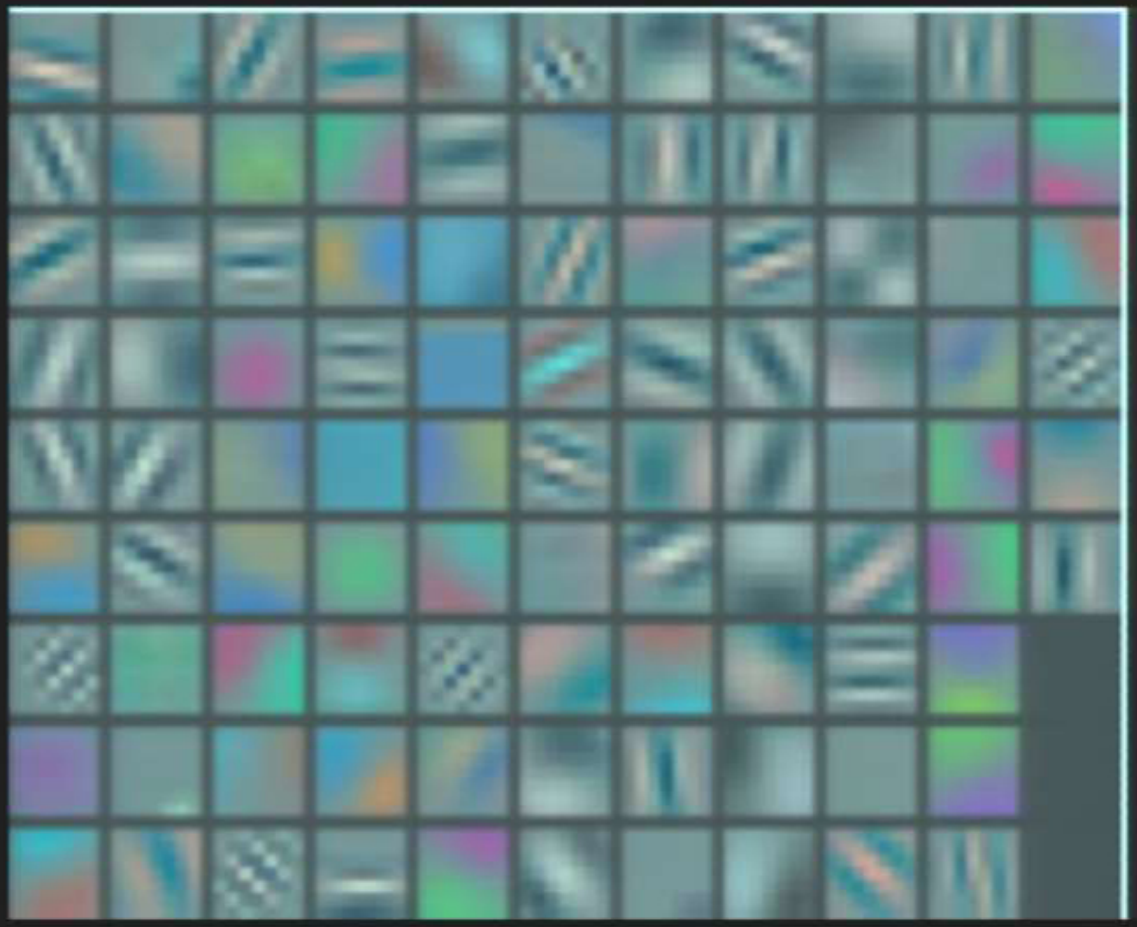


Visualizations of Higher Layers

- Use ImageNet 2012 validation set [Zeiler and Fergus. arXiv'13]
- Push each image through network

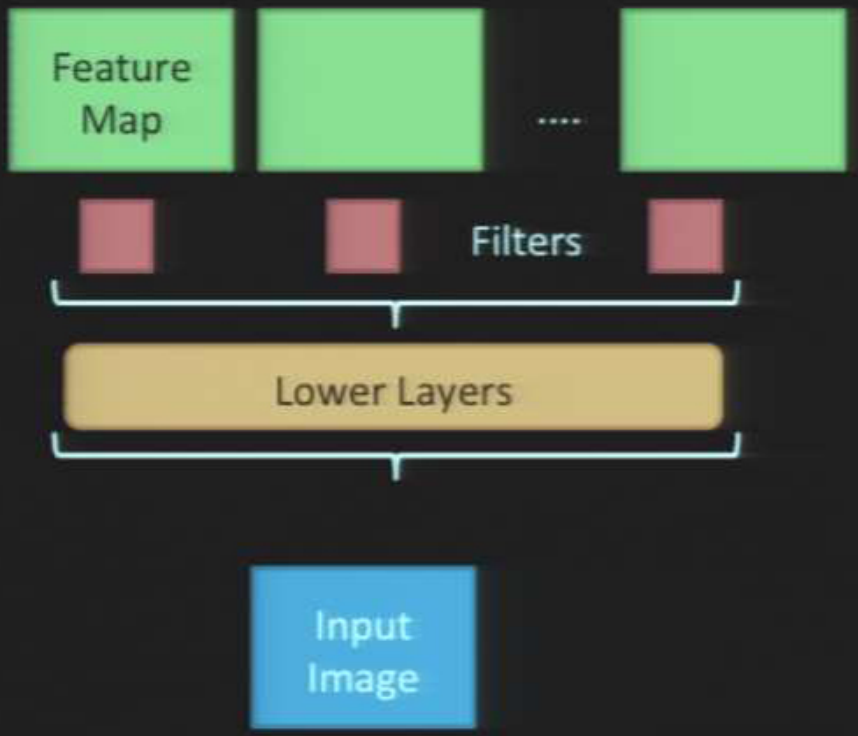


Layer 1 Filters



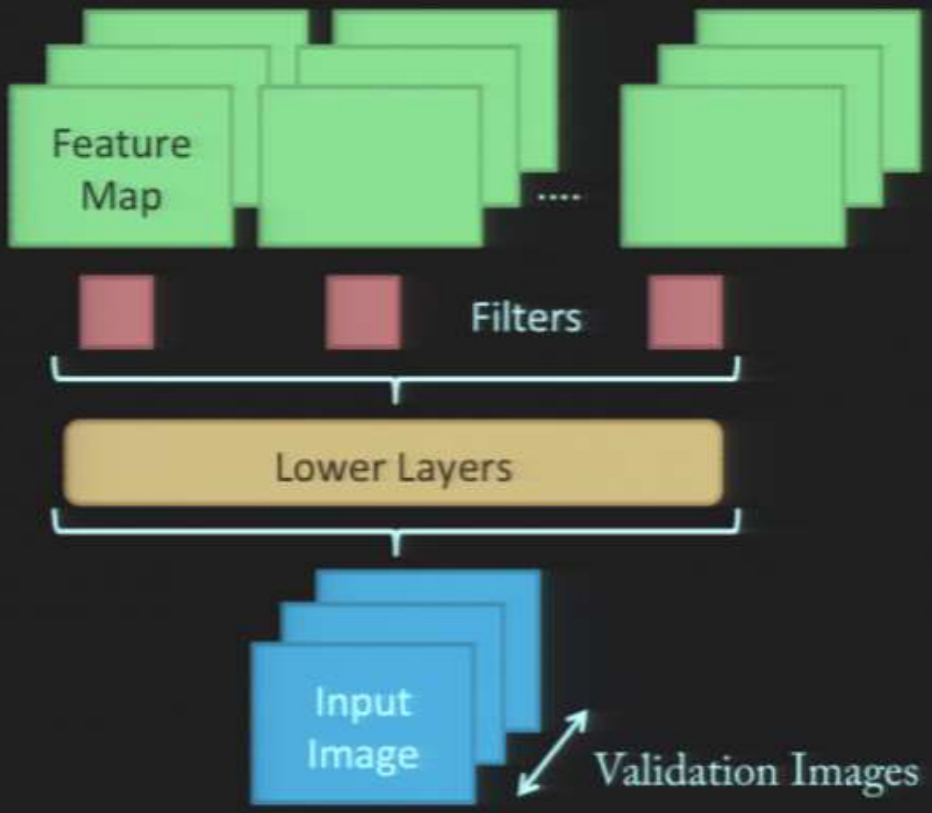
Visualizations of Higher Layers

- Use ImageNet 2012 validation set [Zeiler and Fergus. arXiv'13]
- Push each image through network



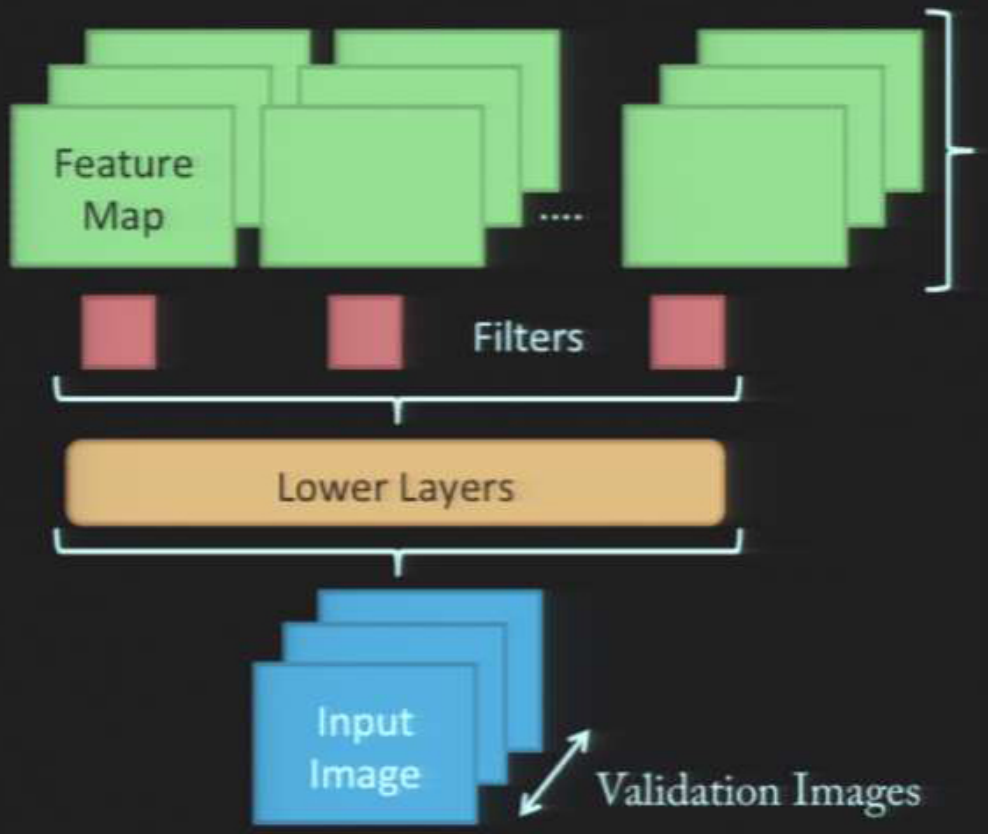
Visualizations of Higher Layers

- Use ImageNet 2012 validation set [Zeiler and Fergus. arXiv'13]
- Push each image through network



Visualizations of Higher Layers

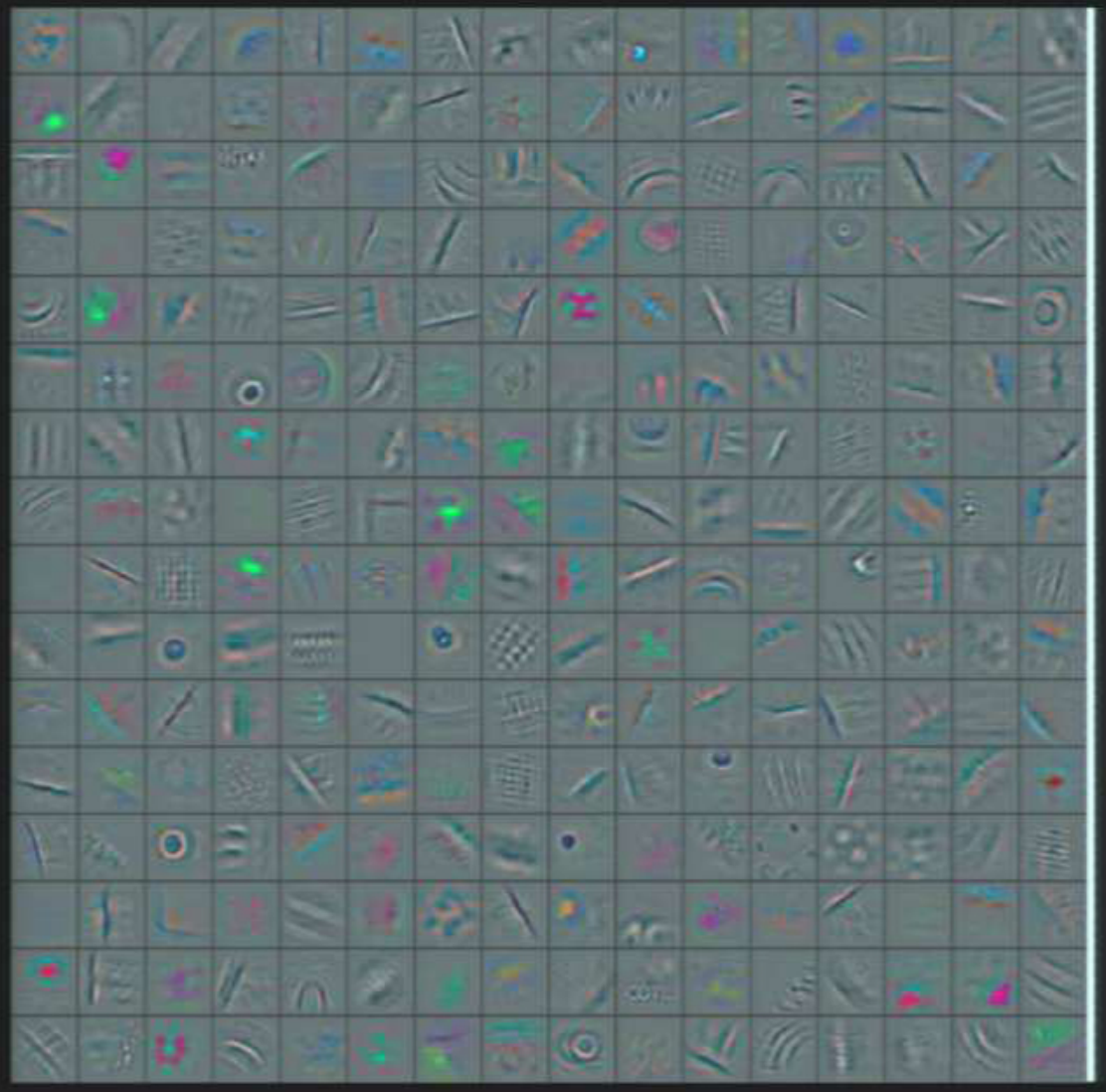
- Use ImageNet 2012 validation set [Zeiler and Fergus. arXiv'13]
- Push each image through network

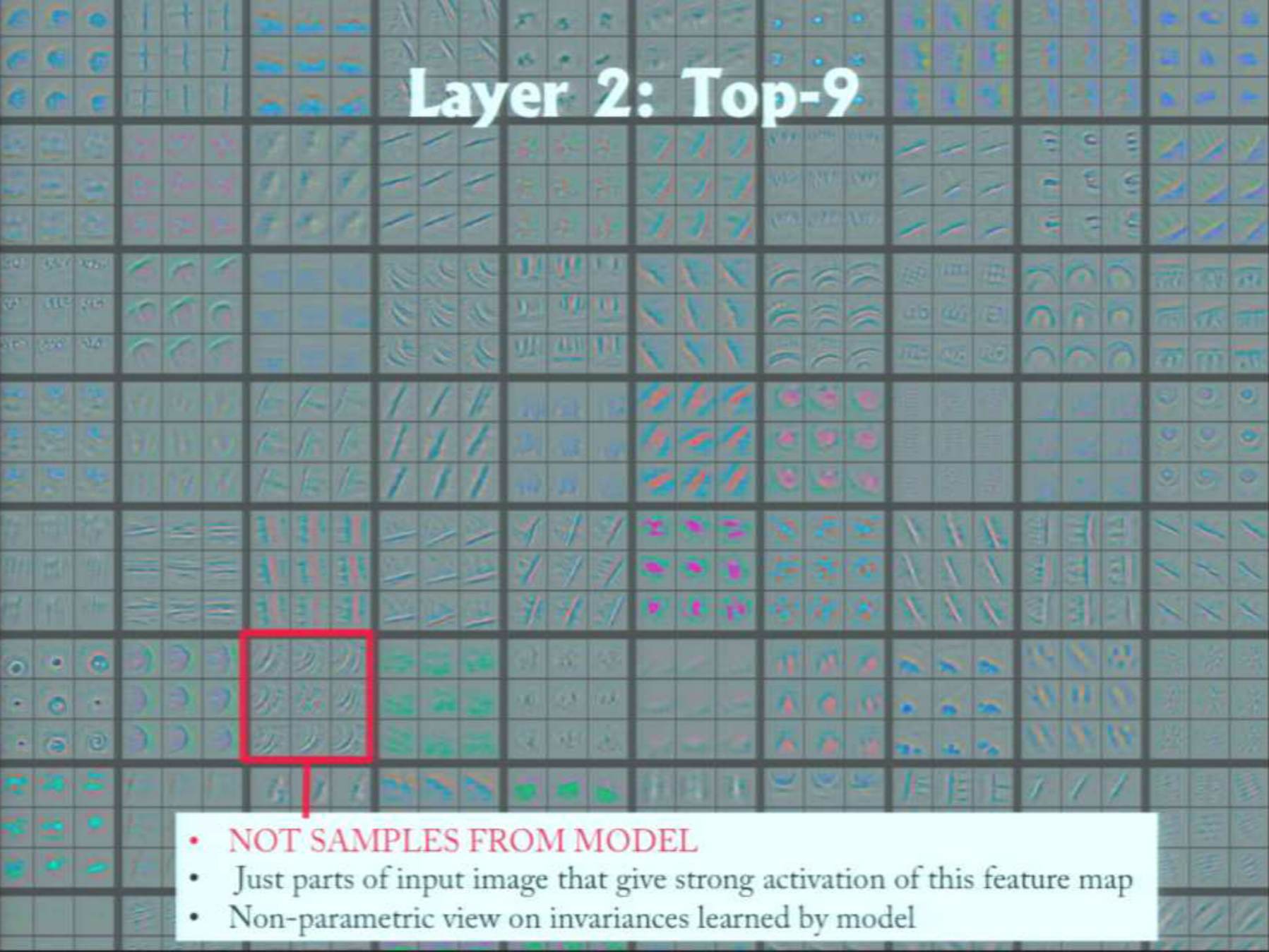


- Take max activation from feature map associated with each filter
- Use Deconvnet to project back to pixel space
- Use pooling "switches" peculiar to that activation

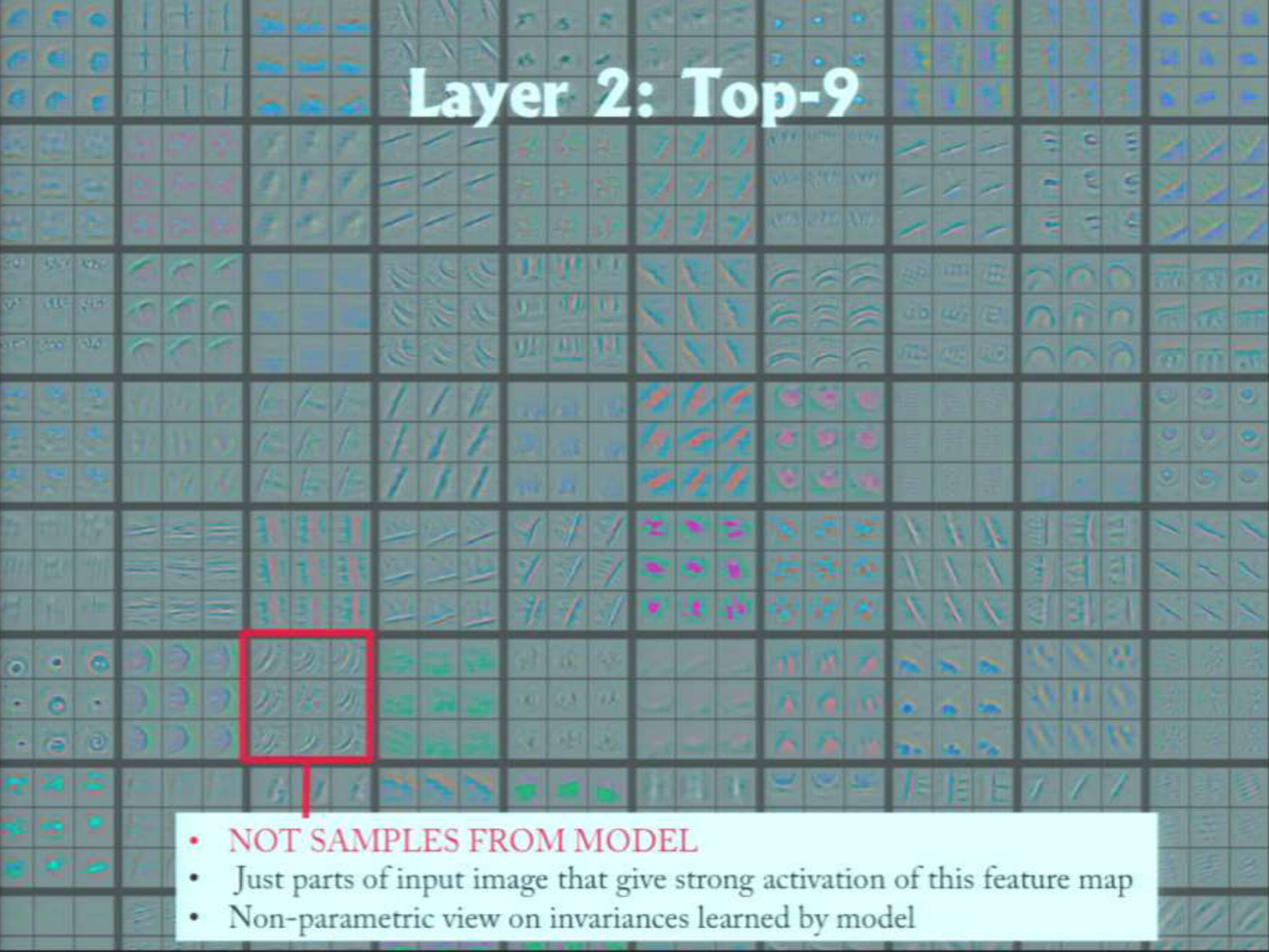
Layer 1: Top-9 Patches

Layer 2: Top-1

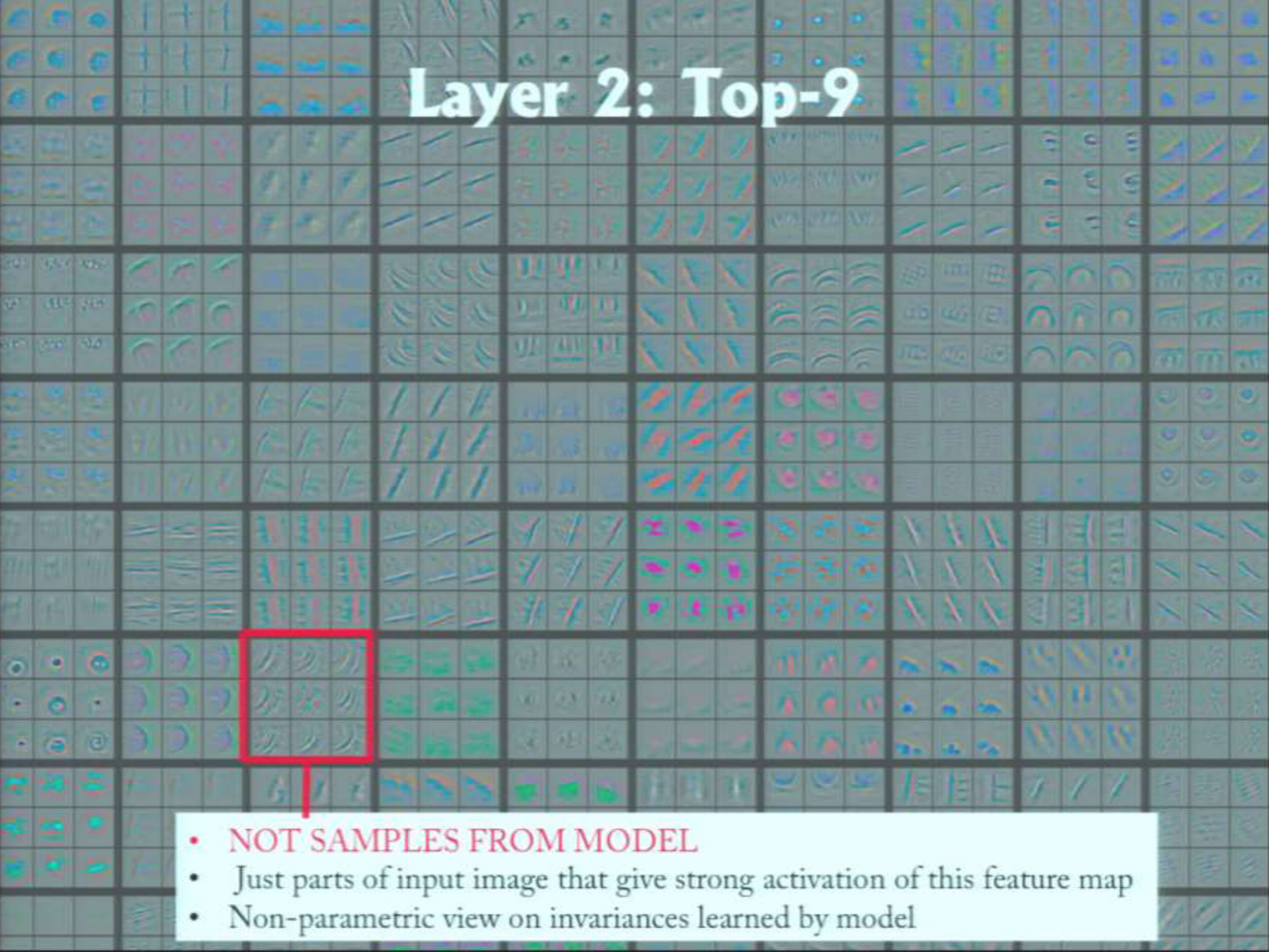



Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model

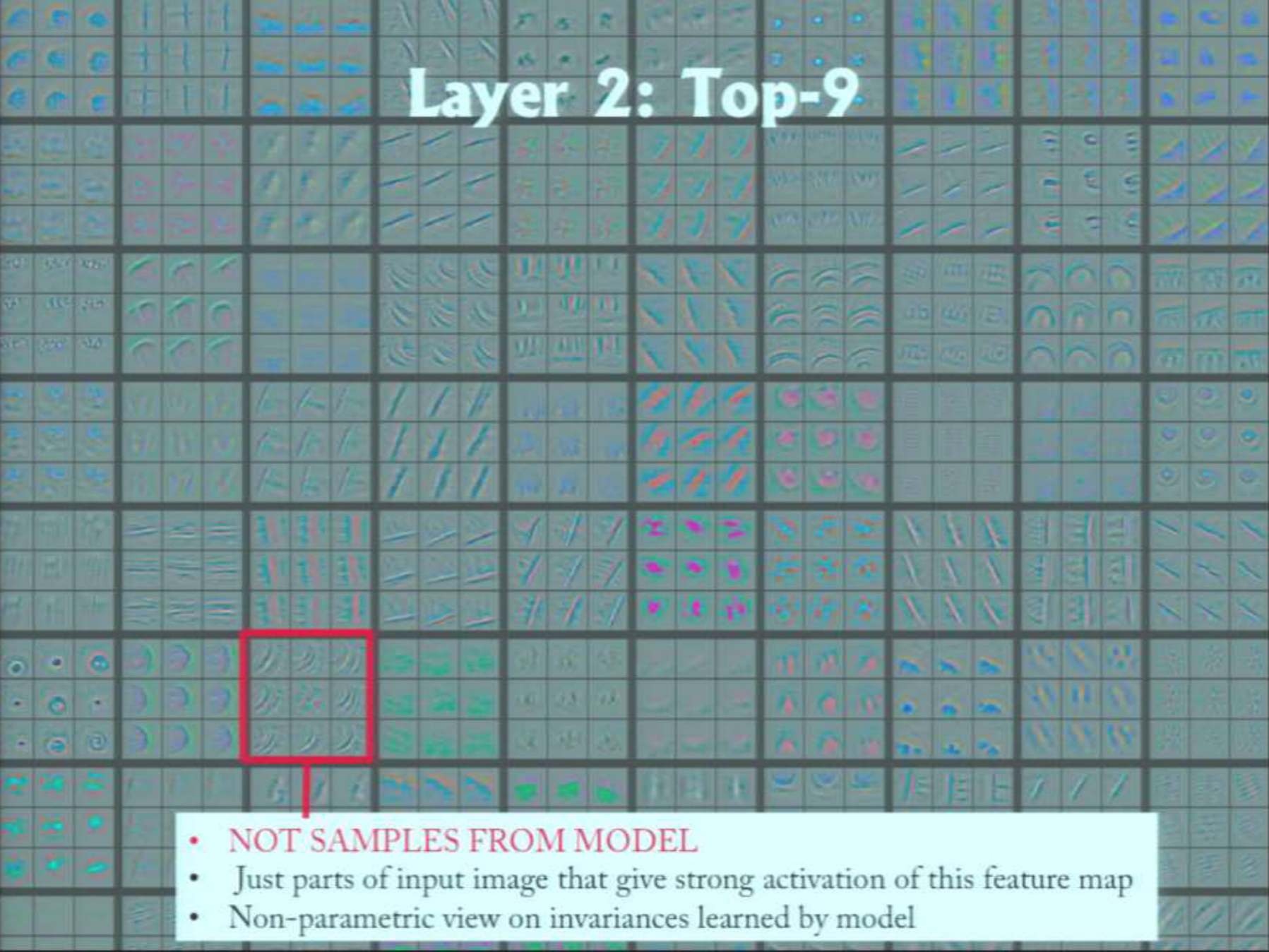

Layer 2: Top-9 Patches
• Patches from validation images that give maximal activation of a given feature map


Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model

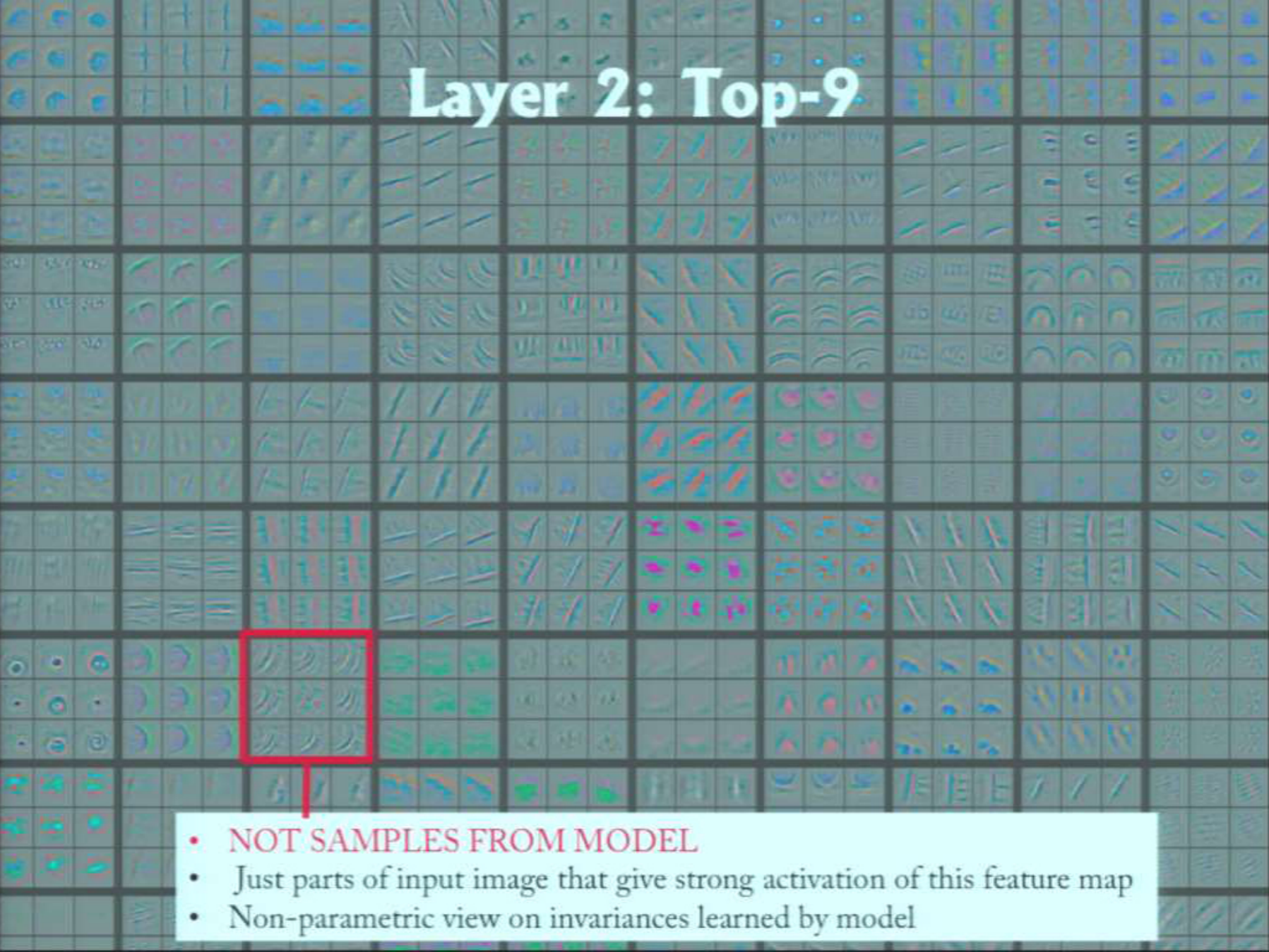

Layer 2: Top-9 Patches
• Patches from validation images that give maximal activation of a given feature map


Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model


Layer 2: Top-9 Patches
• Patches from validation images that give maximal activation of a given feature map

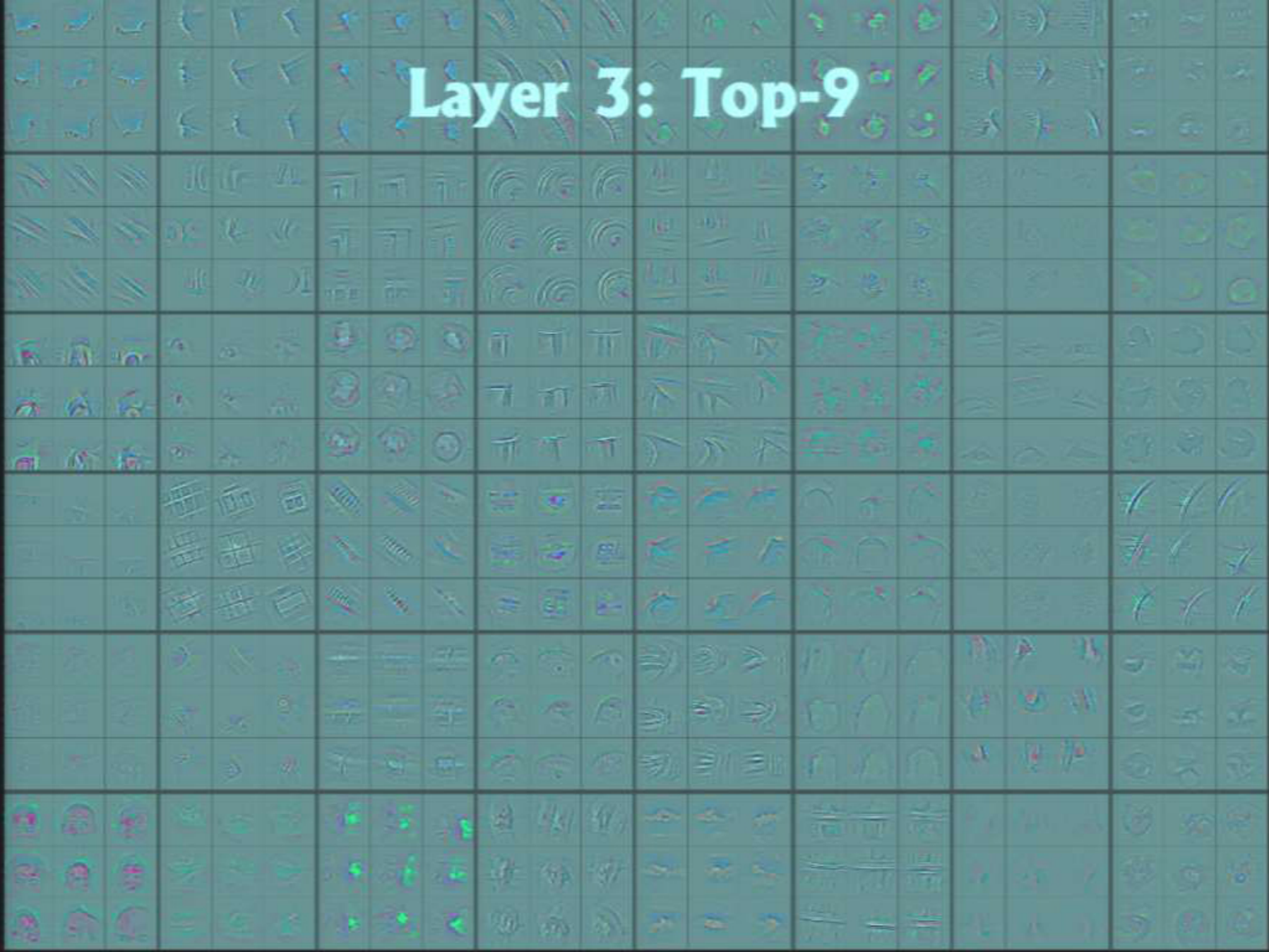

Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model

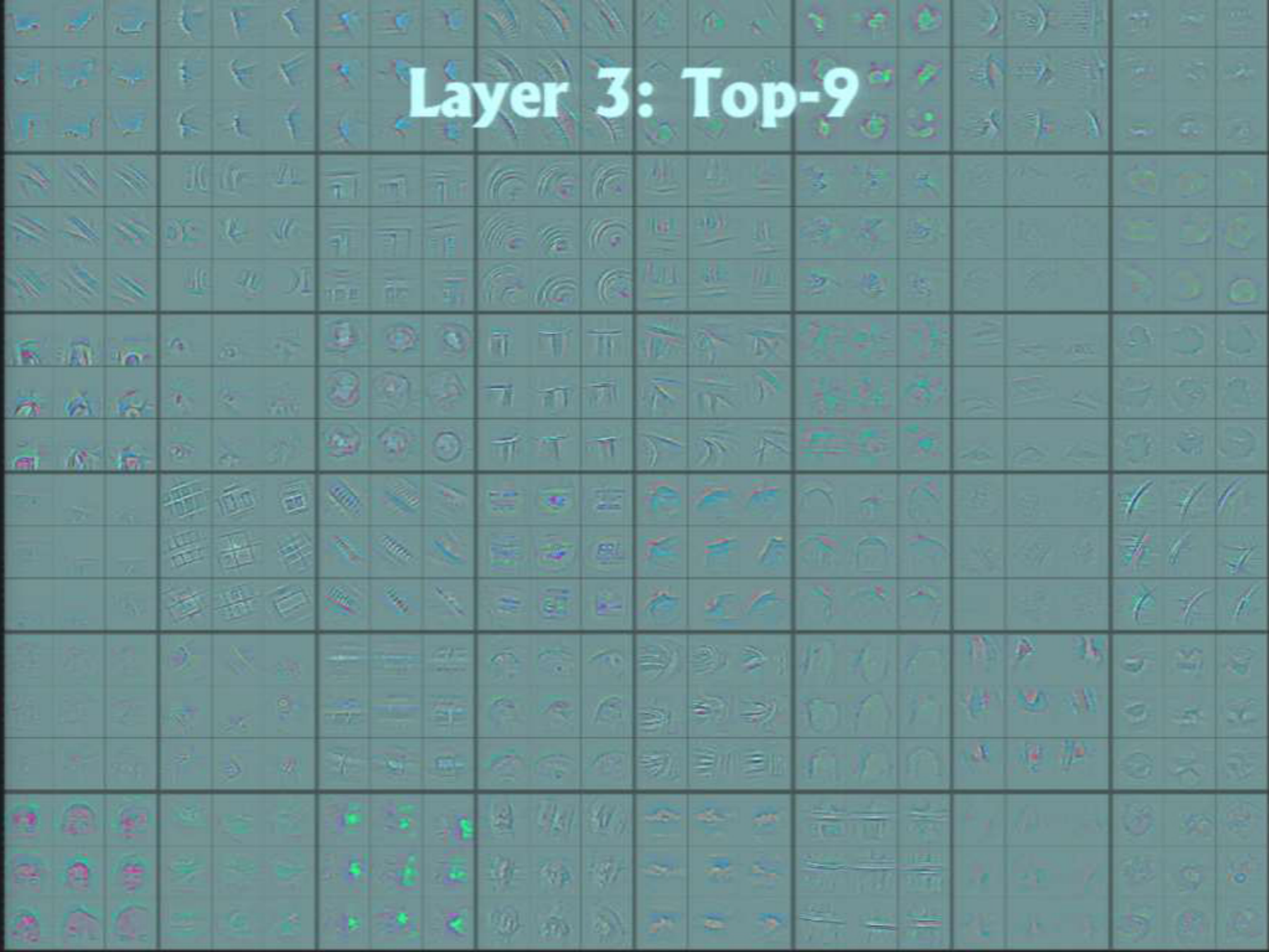

Layer 2: Top-9 Patches
• Patches from validation images that give maximal activation of a given feature map

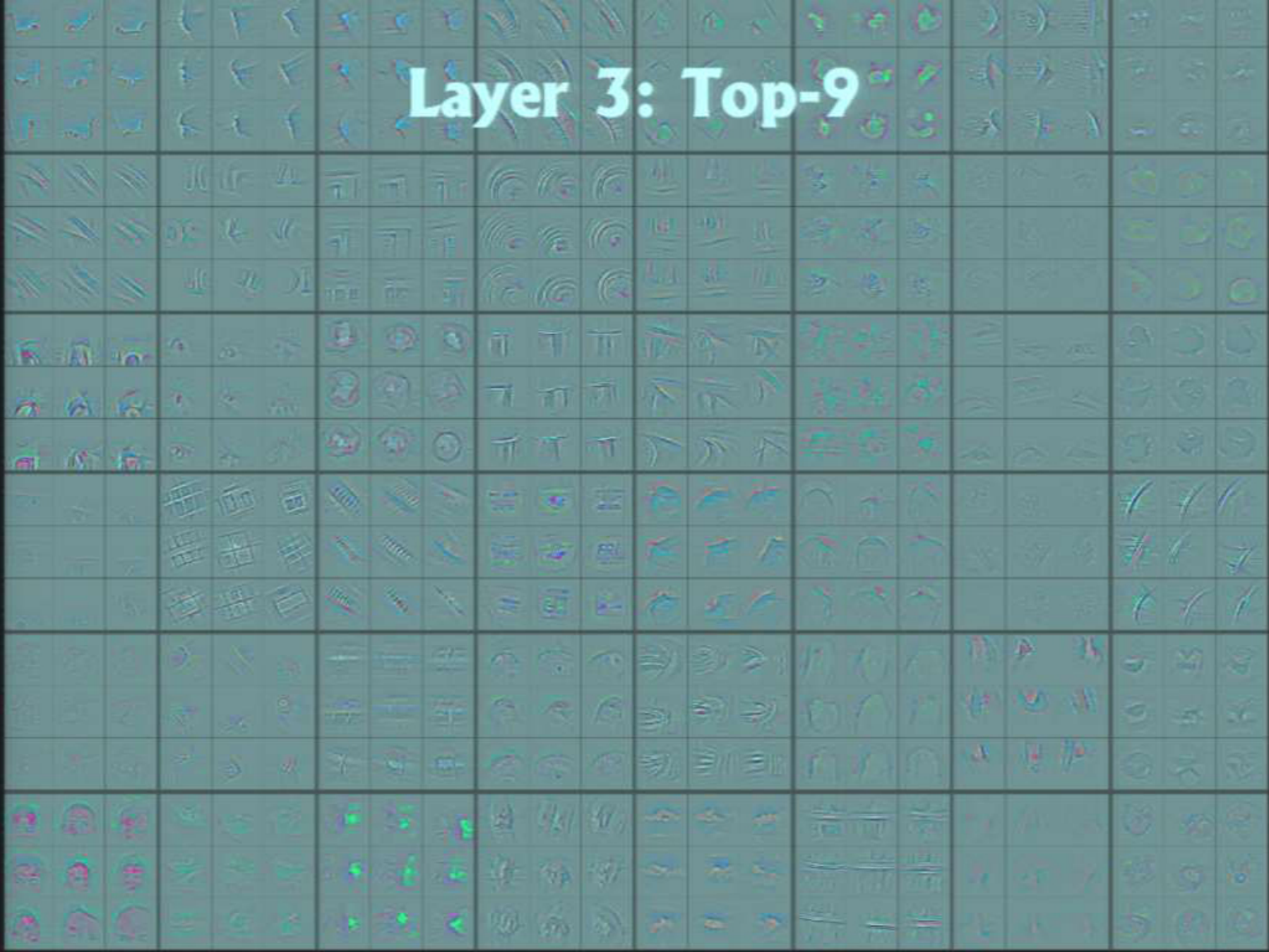

Layer 2: Top-9
• NOT SAMPLES FROM MODEL
• Just parts of input image that give strong activation of this feature map
• Non-parametric view on invariances learned by model

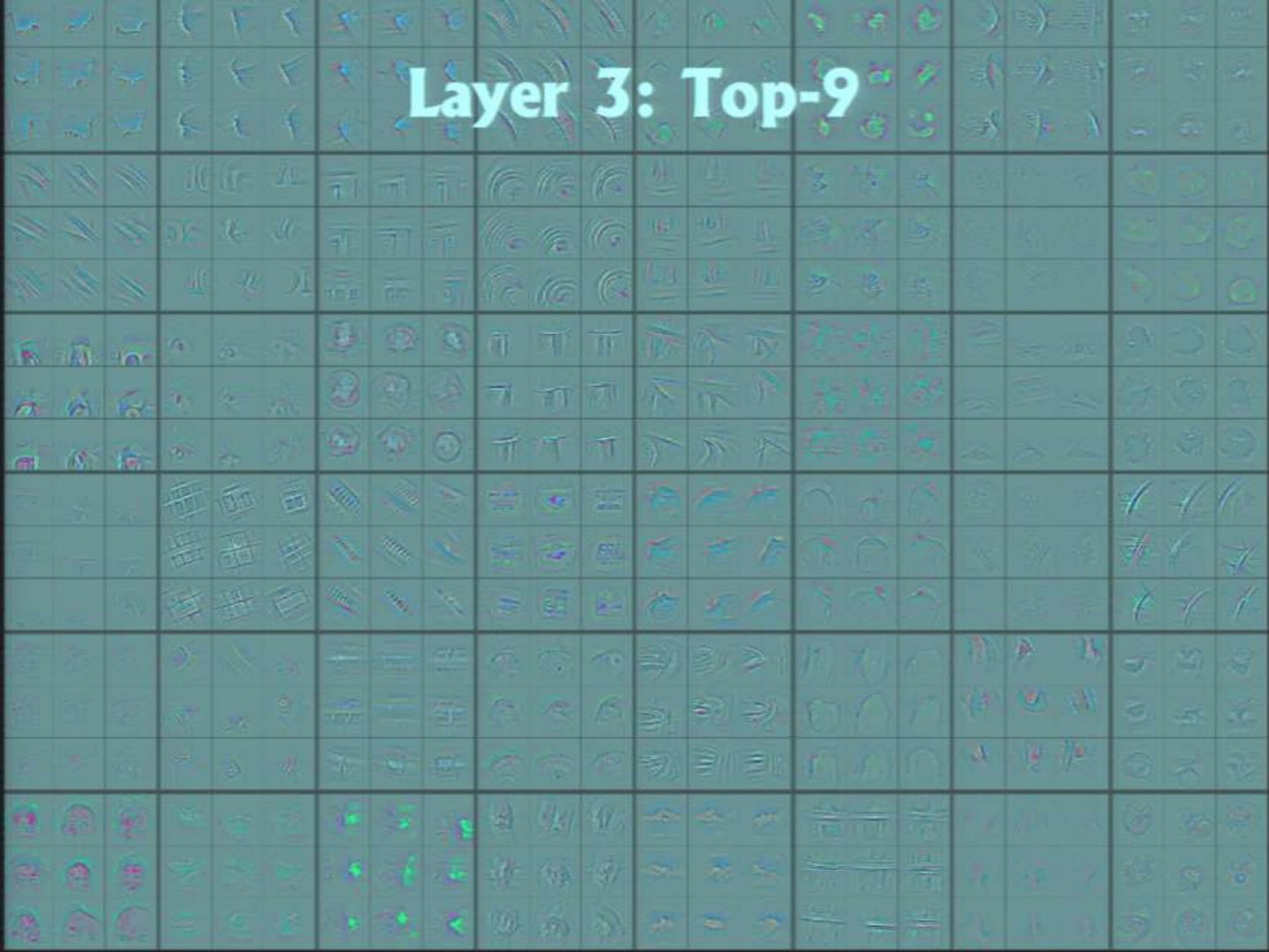

Layer 2: Top-9 Patches
• Patches from validation images that give maximal activation of a given feature map

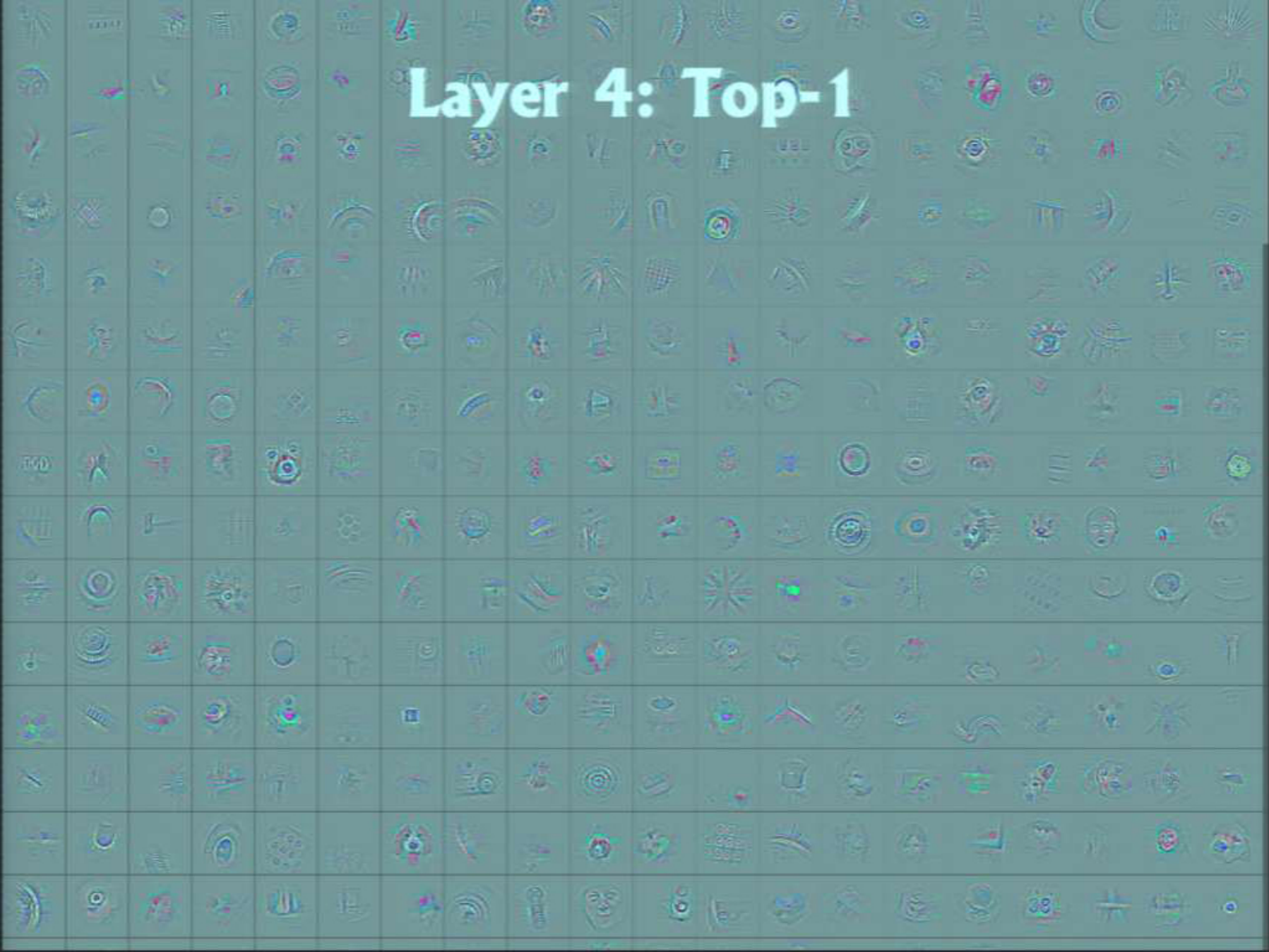

Layer 3: Top-1

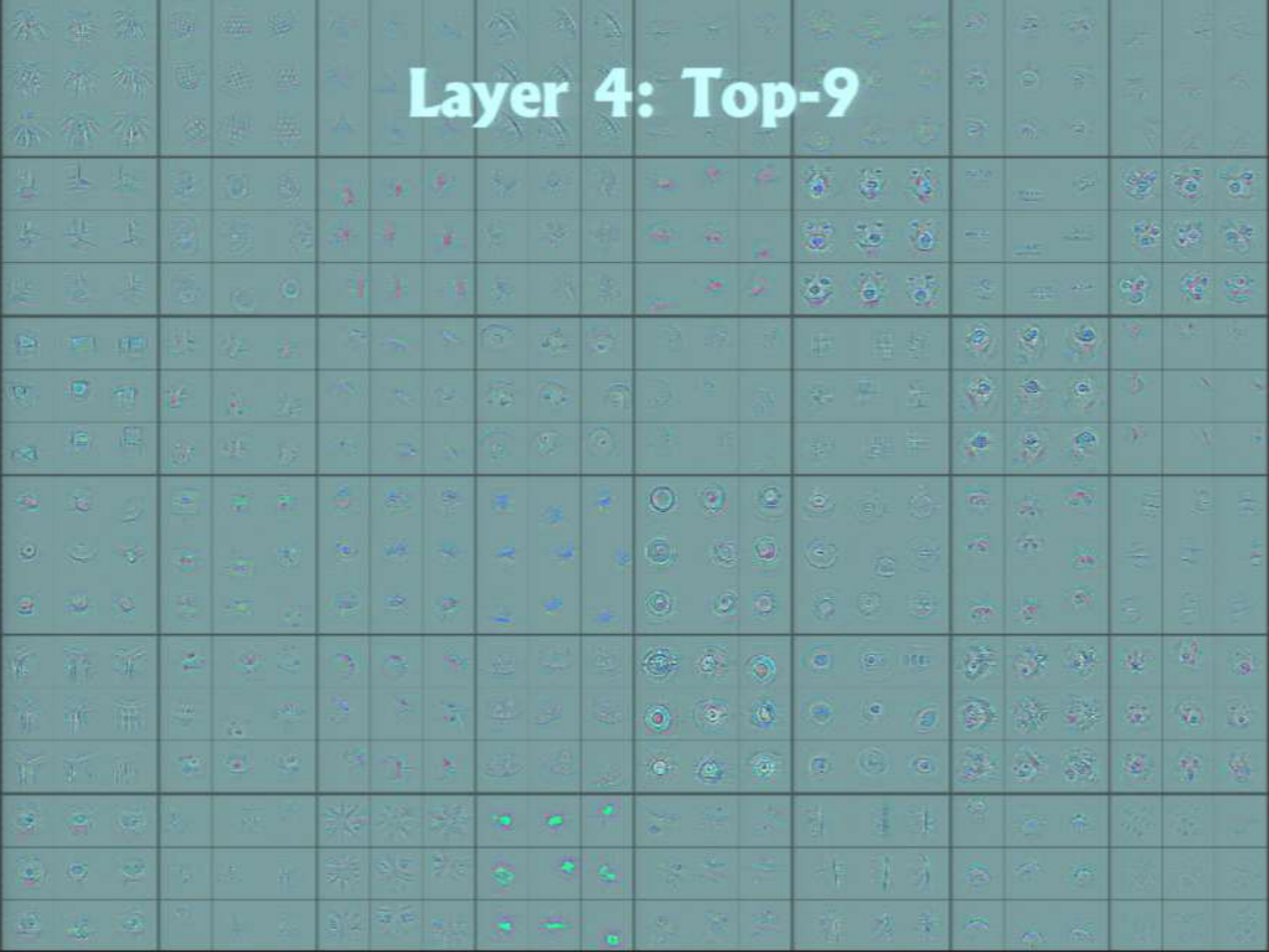

Layer 3: Top-9


Layer 3: Top-9

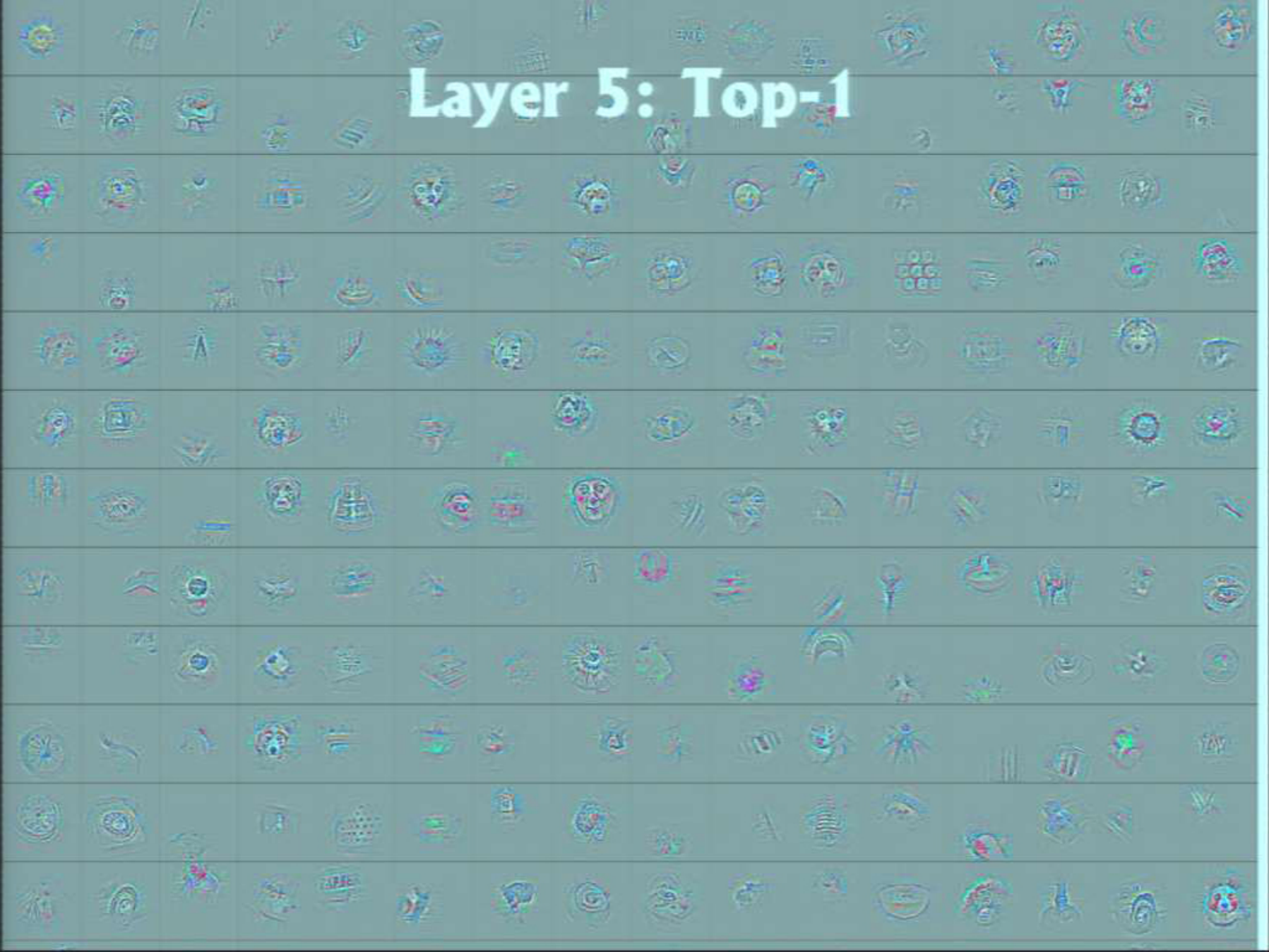

Layer 3: Top-9


Layer 3: Top-9


Layer 4: Top-1


Layer 4: Top-9


Layer 4: Top-9 Patches


Layer 5: Top-1


Layer 5: Top-9


Layer 5: Top-9 Patches


Layer 5: Top-9


Layer 5: Top-9 Patches


Layer 5: Top-9


Layer 5: Top-9 Patches


Layer 5: Top-9


Layer 5: Top-9 Patches


Layer 5: Top-9


Layer 5: Top-9 Patches

Diagnosing Problems

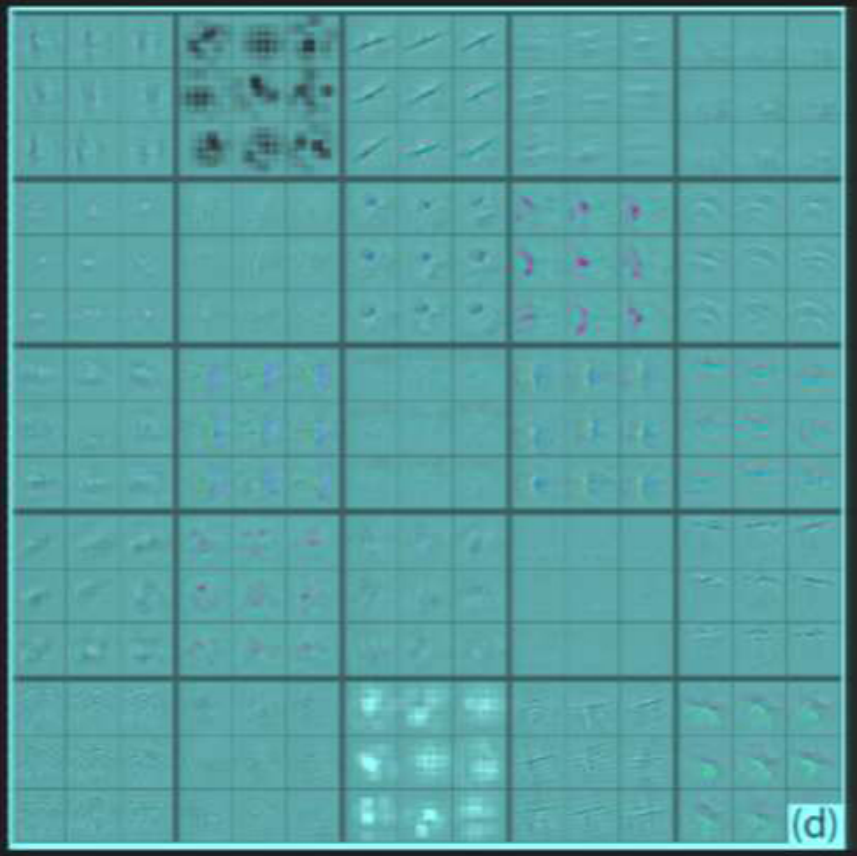
- Visualization of Krizhevsky et al.'s architecture showed some problems with layers 1 and 2
  - Large stride of 4 used

- Alter architecture: smaller stride & filter size
  - Visualizations look better
  - Performance improves

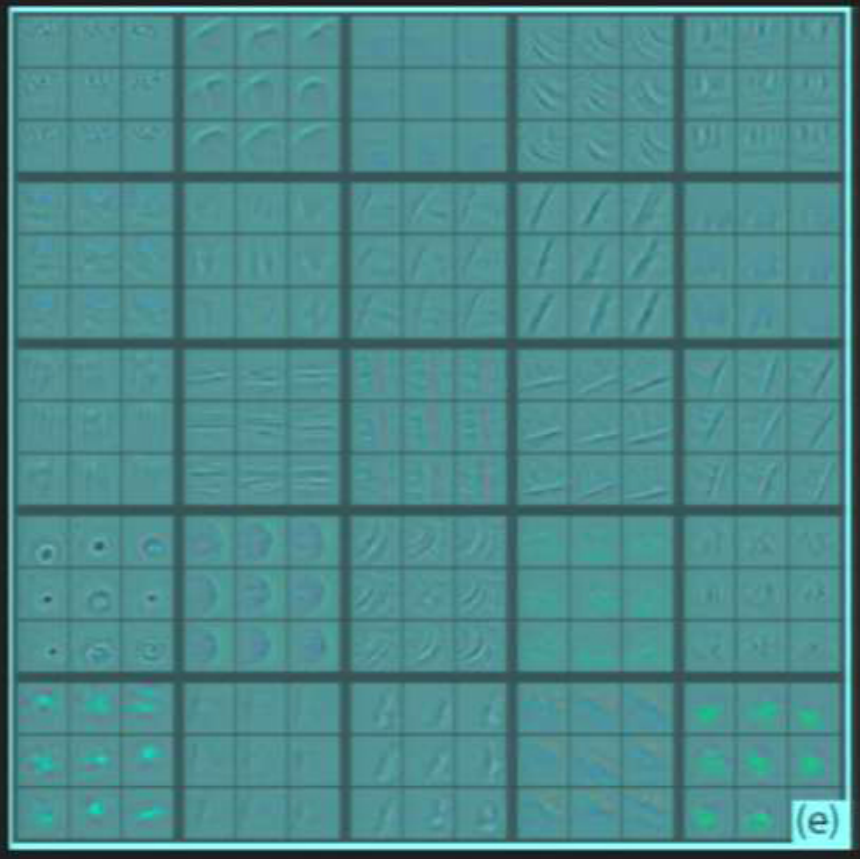
Comparison with Krizhevsky et al.

- Layer 2 visualizations

Krizhevsky et al.



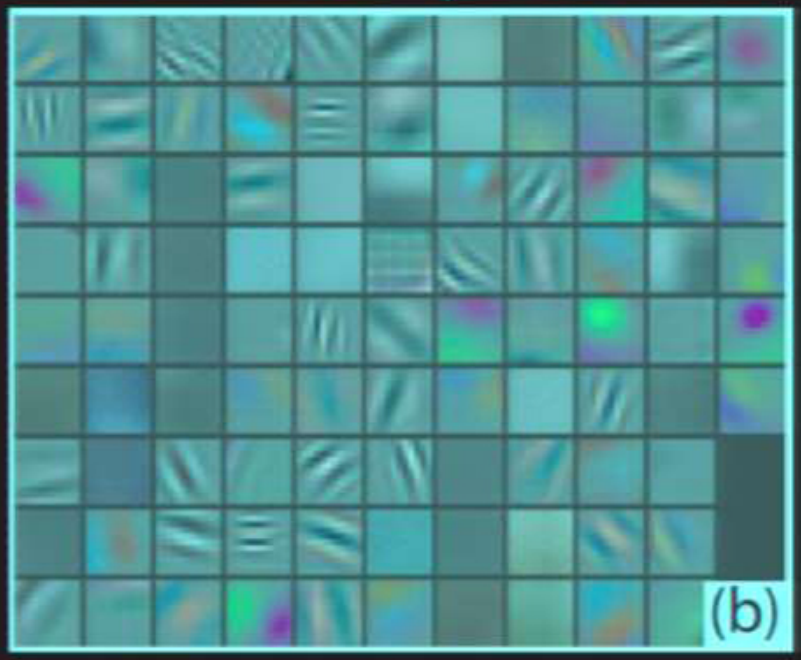
Ours



Comparison with Krizhevsky et al.

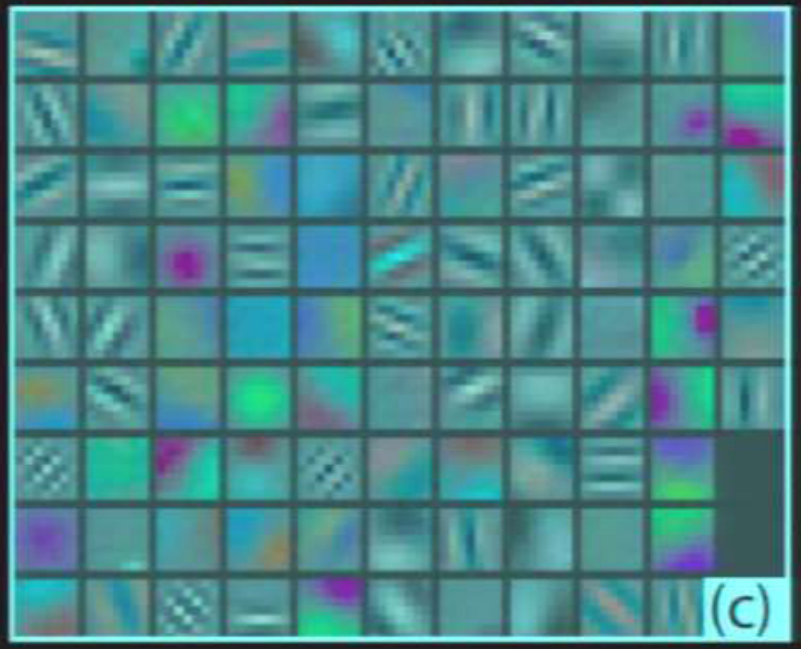
- Layer 1 filters

Krizhevsky et al.



11x11 filters, stride 4

Ours



7x7 filters, stride 2

ImageNet Classification 2012

[Zeiler and Fergus. arXiv'13]

Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	--
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	--

* Trained using Imagnet 2011 and 2012 training sets.

ImageNet Classification 2012

[Zeiler and Fergus. arXiv'13]

Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	--
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	--
1 convnet as per Fig. 3	38.4	16.5	--

* Trained using Imagnet 2011 and 2012 training sets.

ImageNet Classification 2012

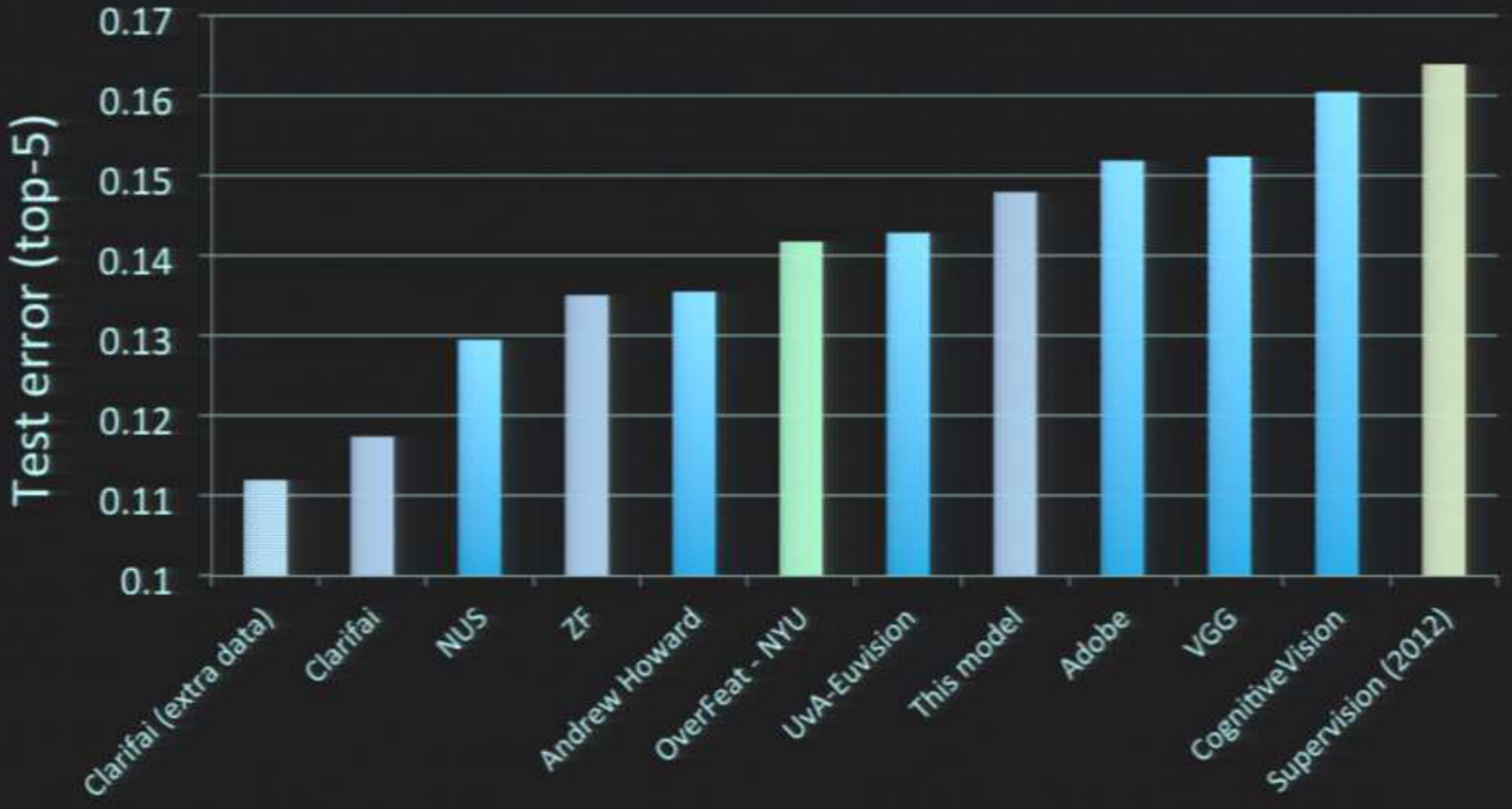
[Zeiler and Fergus. arXiv'13]

Error %	Val Top-1	Val Top-5	Test Top-5
(Gunji et al., 2012)	-	-	26.2
(Krizhevsky et al., 2012), 1 convnet	40.7	18.2	--
(Krizhevsky et al., 2012), 5 convnets	38.1	16.4	16.4
(Krizhevsky et al., 2012)*, 1 convnets	39.0	16.6	--
(Krizhevsky et al., 2012)*, 7 convnets	36.7	15.4	15.3
Our replication of (Krizhevsky et al., 2012), 1 convnet	40.5	18.1	--
1 convnet as per Fig. 3	38.4	16.5	--
5 convnets as per Fig. 3 – (a)	36.7	15.3	15.3
1 convnet as per Fig. 3 but with layers 3,4,5: 512,1024,512 maps – (b)	37.5	16.0	16.1
6 convnets, (a) & (b) combined	**36.0**	**14.7**	**14.8**

* Trained using Imagnet 2011 and 2012 training sets.

ImageNet Classification 2013 Results

- http://www.image-net.org/challenges/LSVRC/2013/results.php



- Pre-2012: 26.2% error → 2012: 16.5% error → 2013: 11.2% error

How to Choose Architecture

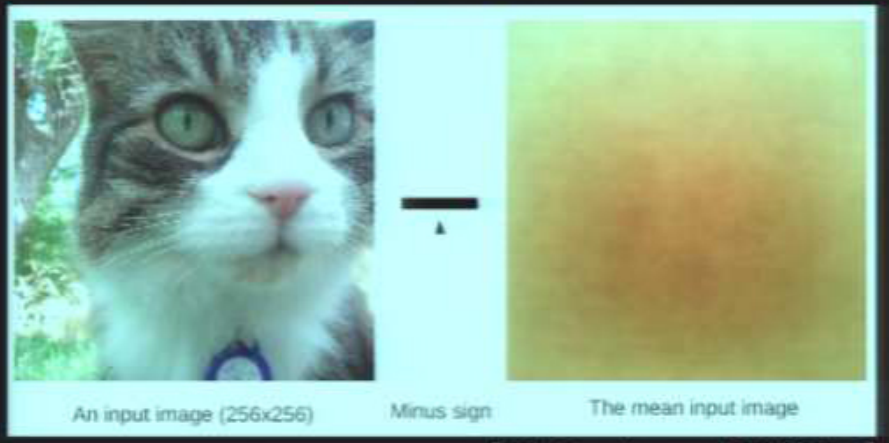
- Task-dependent but many hyper-parameters:
  - # layers, # feature maps, strides in conv/pool
  - limited by amount of labeled data & GPUs
- Cross-validation
- Grid search (need lots of GPUs)
- Smarter strategies:
  - Random [Bergstra & Bengio JMLR 2012]
  - Bayesian optimization [Snoek et al. NIPS 2012]
  - Using visualizations [Zeiler & Fergus, arXiv 1311.2901]

Training Big ConvNets

- Back-propagation of error
  - [Rumelhart, Hinton & Williams 1986] + many others
  - Chain rule
- Stochastic Gradient Descent
  - 2nd order methods expensive
  - L. Bottou "Stochastic Gradient Tricks" Neural Networks 2012
- Momentum
  - Nesterov variant [Sutskever et al. ICML 2012]
- Classification loss: cross-entropy
- GPU implementation

Pre-Processing

- Mean removal

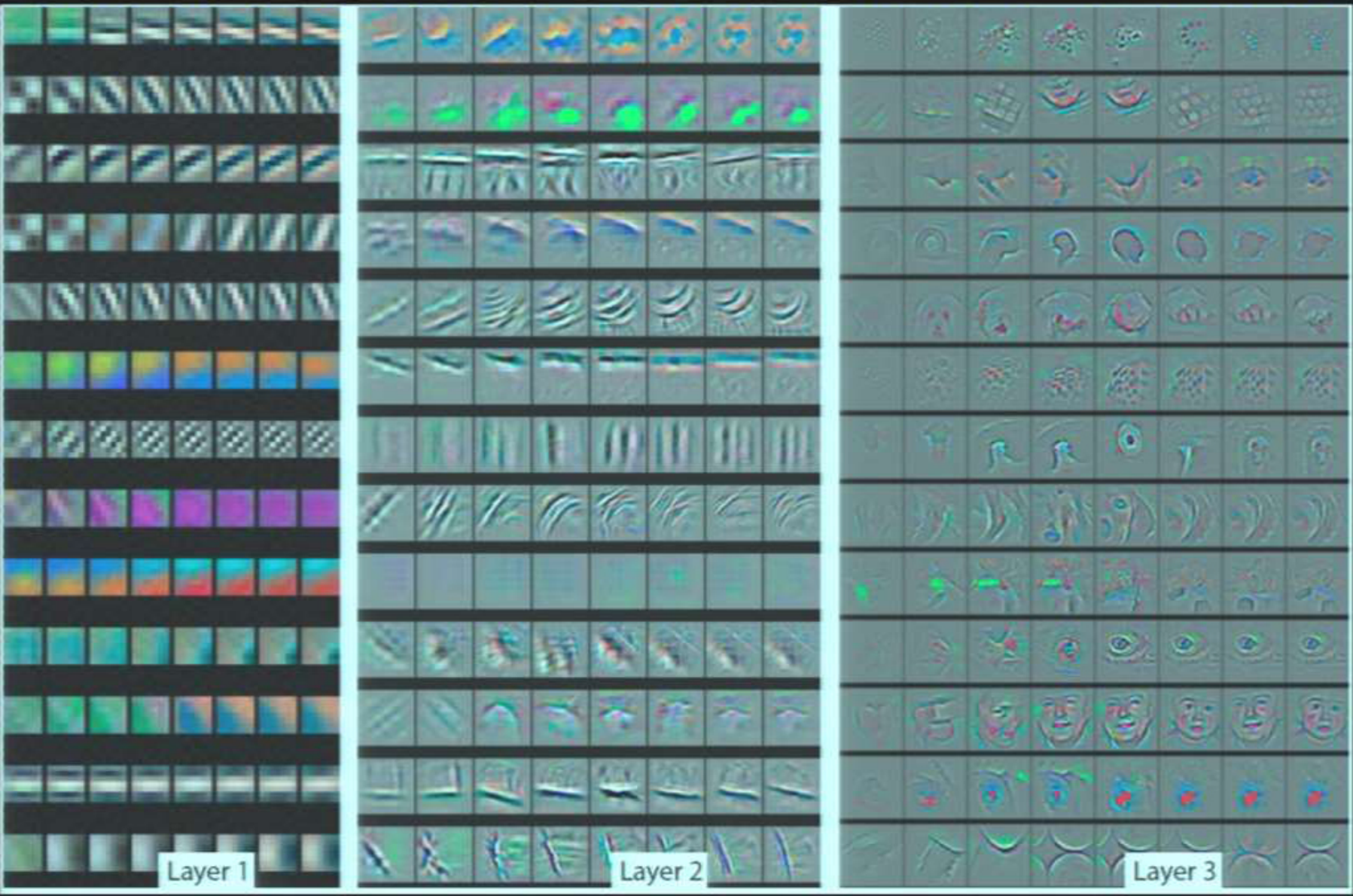


[Krizhevsky et al. NIPS'12]

- Whitening (ZCA)
  - Form of PCA
  - Removes correlations
  - But too expensive for entire image

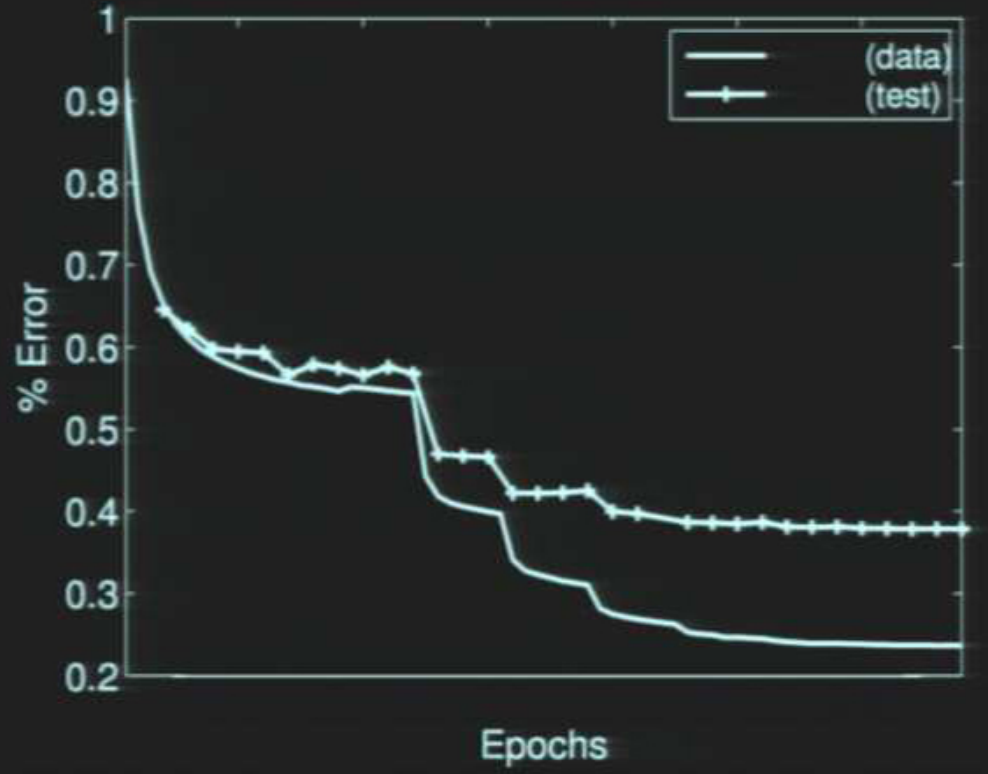
- Contrast normalization

Evolution of Features During Training

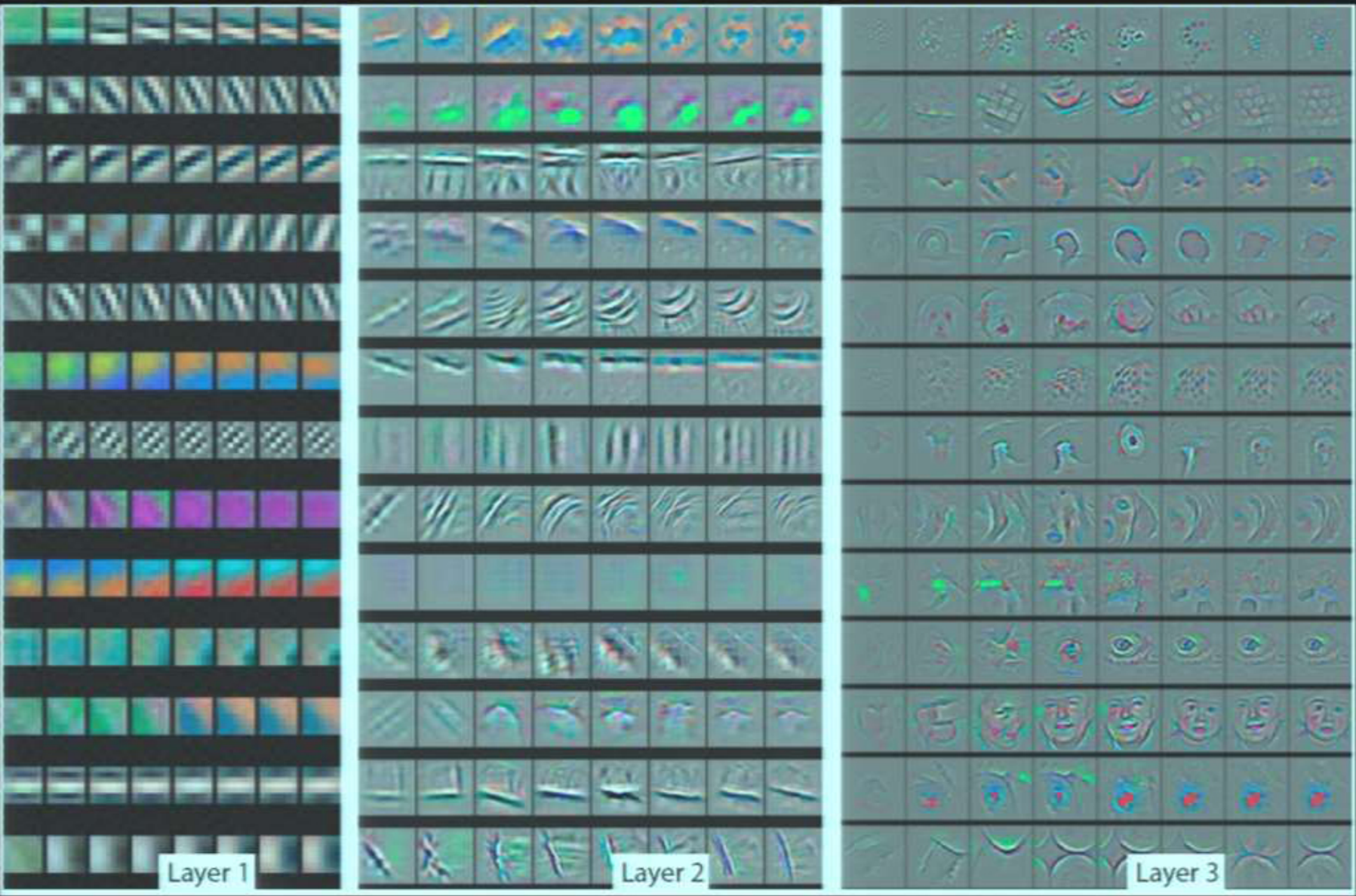


Annealing of Learning Rate

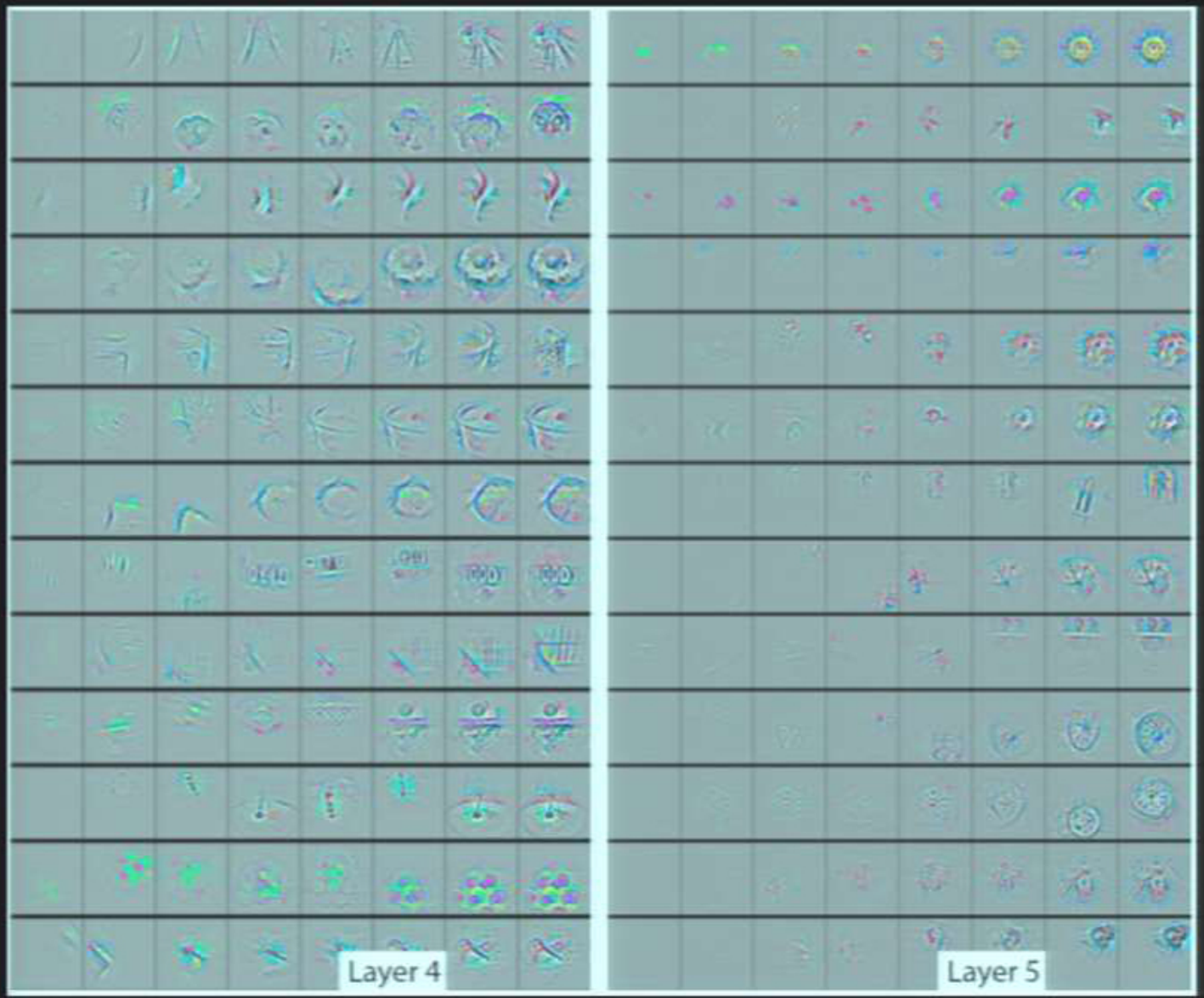
- Start large & slowly reduce, to 100-1000x smaller by end
- Explore different scales of energy surface



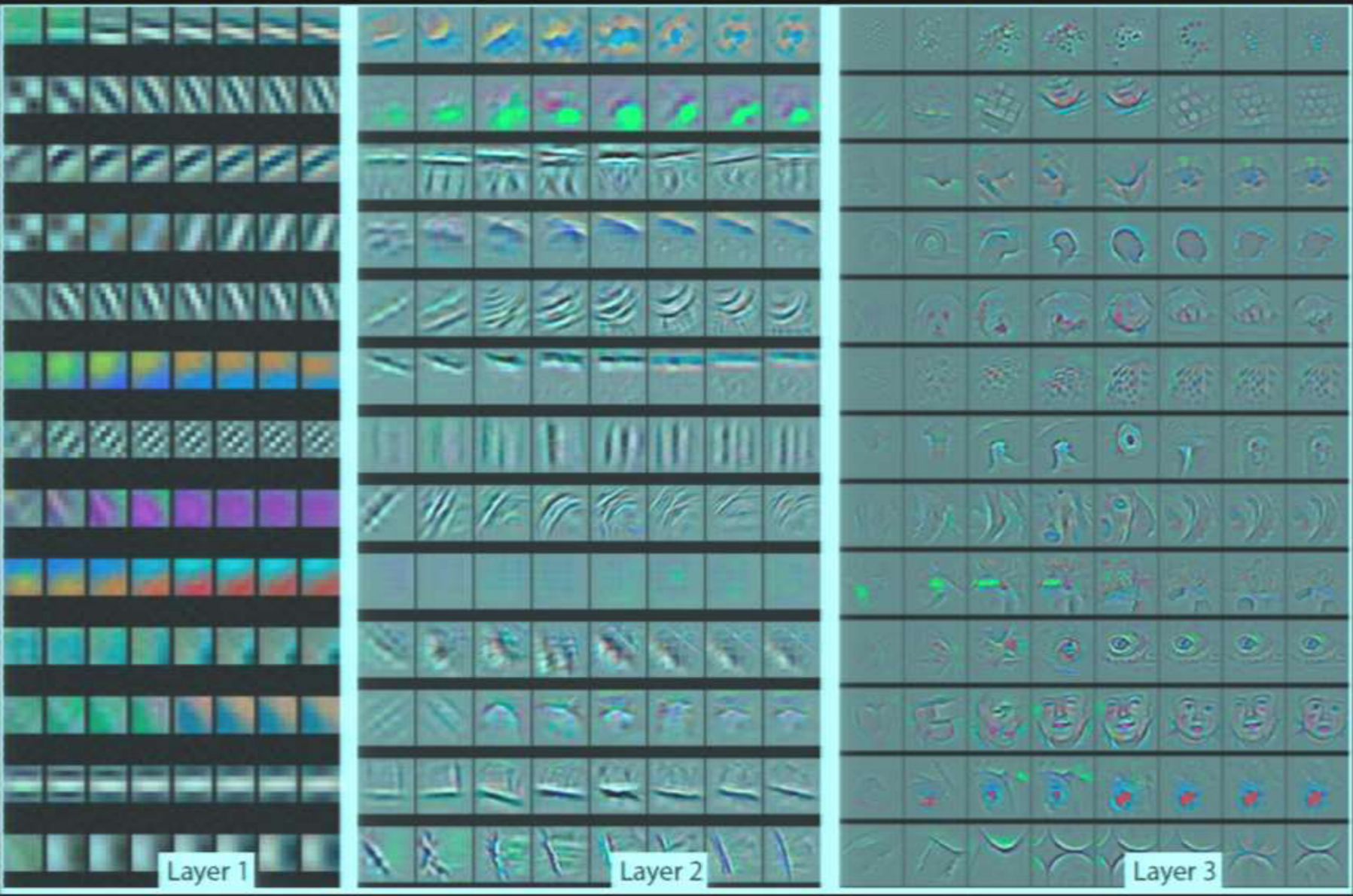
Evolution of Features During Training



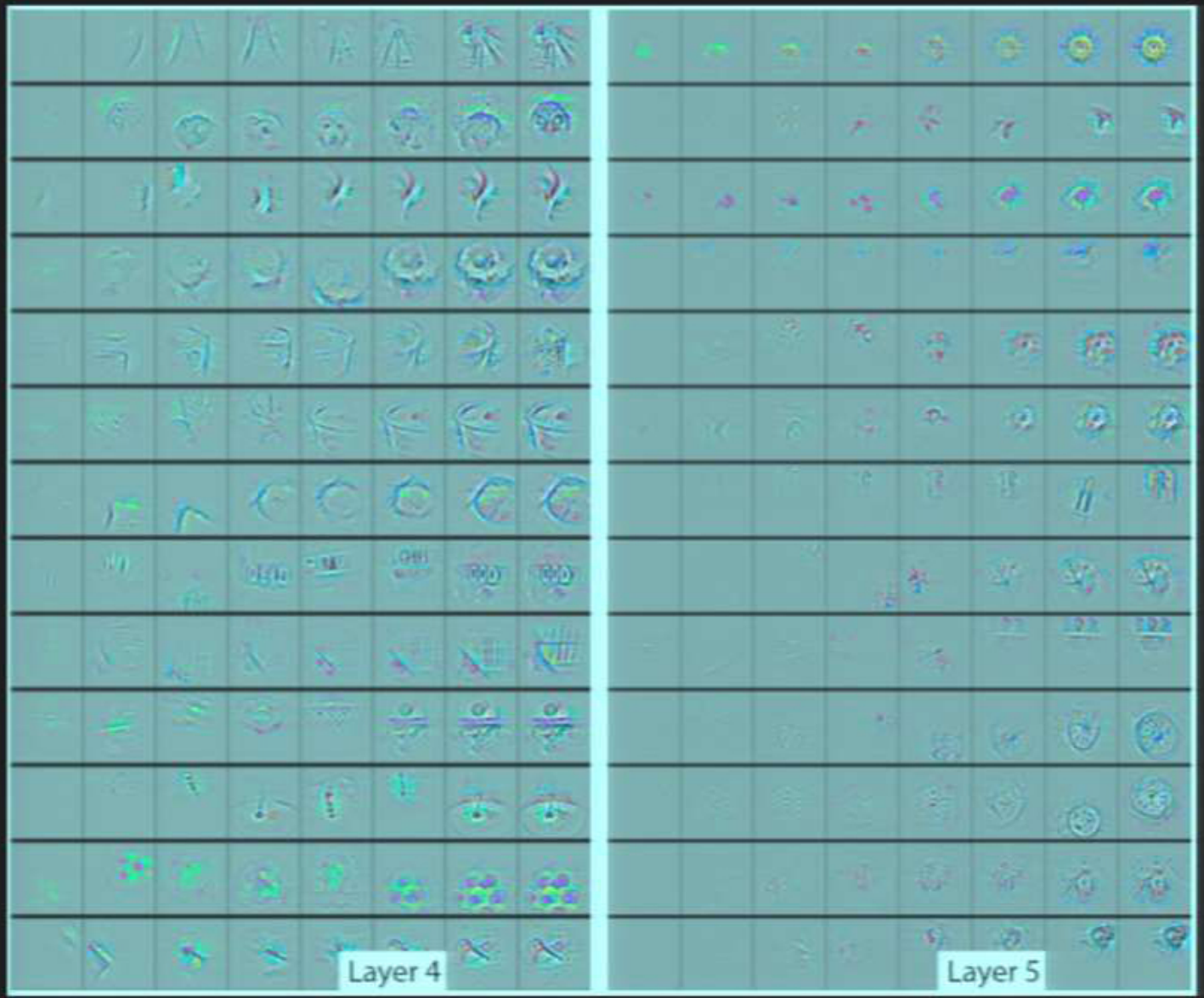
Evolution of Features During Training



Evolution of Features During Training



Evolution of Features During Training



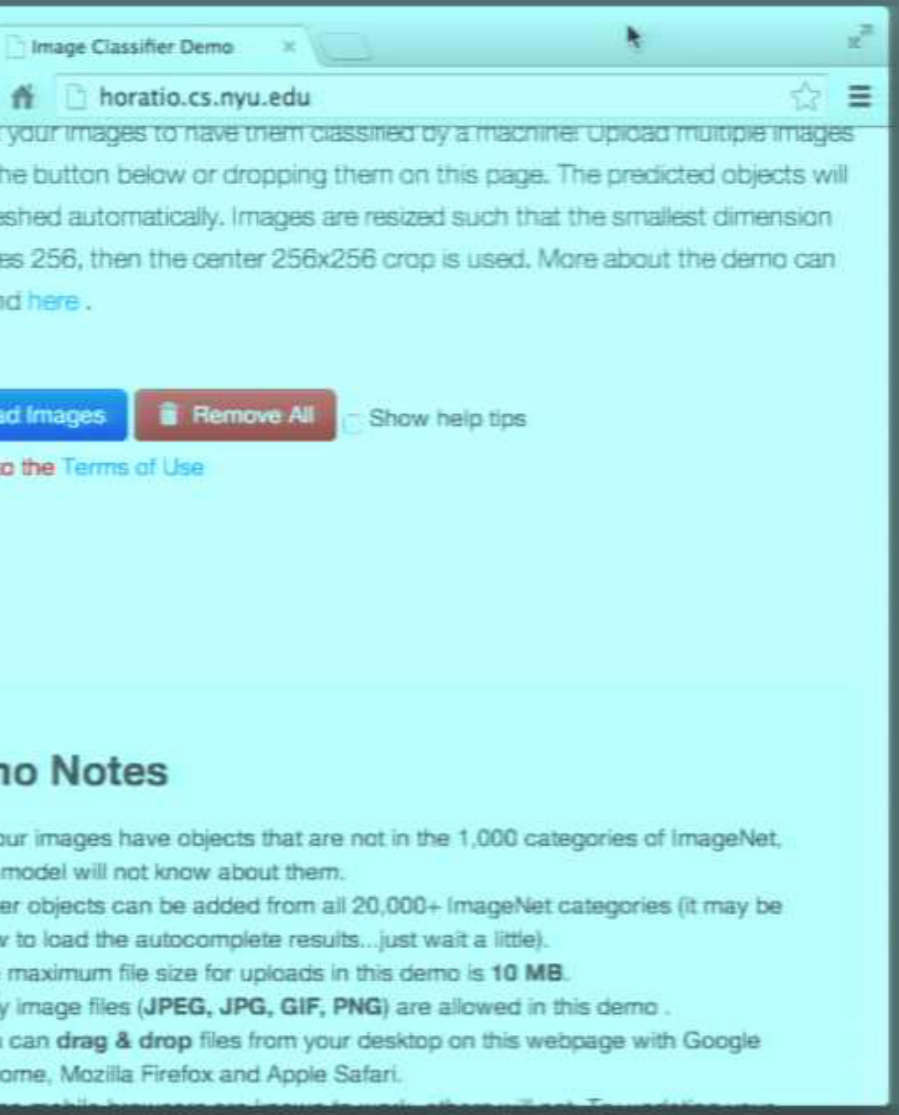


Image Classifier Demo
horatio.cs.nyu.edu
your images to have them classified by a machine! Upload multiple images
he button below or dropping them on this page. The predicted objects will
eshed automatically. Images are resized such that the smallest dimension
es 256, then the center 256x256 crop is used. More about the demo can
nd here .
d Images
Remove All
Show help tips
o the Terms of Use
no Notes
our images have objects that are not in the 1,000 categories of ImageNet,
model will not know about them.
er objects can be added from all 20,000+ ImageNet categories (it may be
v to load the autocomplete results...just wait a little).
maximum file size for uploads in this demo is 10 MB.
y image files (JPEG, JPG, GIF, PNG) are allowed in this demo .
can drag & drop files from your desktop on this webpage with Google
ome, Mozilla Firefox and Apple Safari.


Image Classifier Demo
horatio.cs.nyu.edu
assifier Demo
ge Classifier Demo
NYU
your images to have them classified by a machine! Upload multiple images
he button below or dropping them on this page. The predicted objects will
eshed automatically. Images are resized such that the smallest dimension
es 256, then the center 256x256 crop is used. More about the demo can
d here .
d Images
Remove All
Show help tips
o the Terms of Use
no Notes

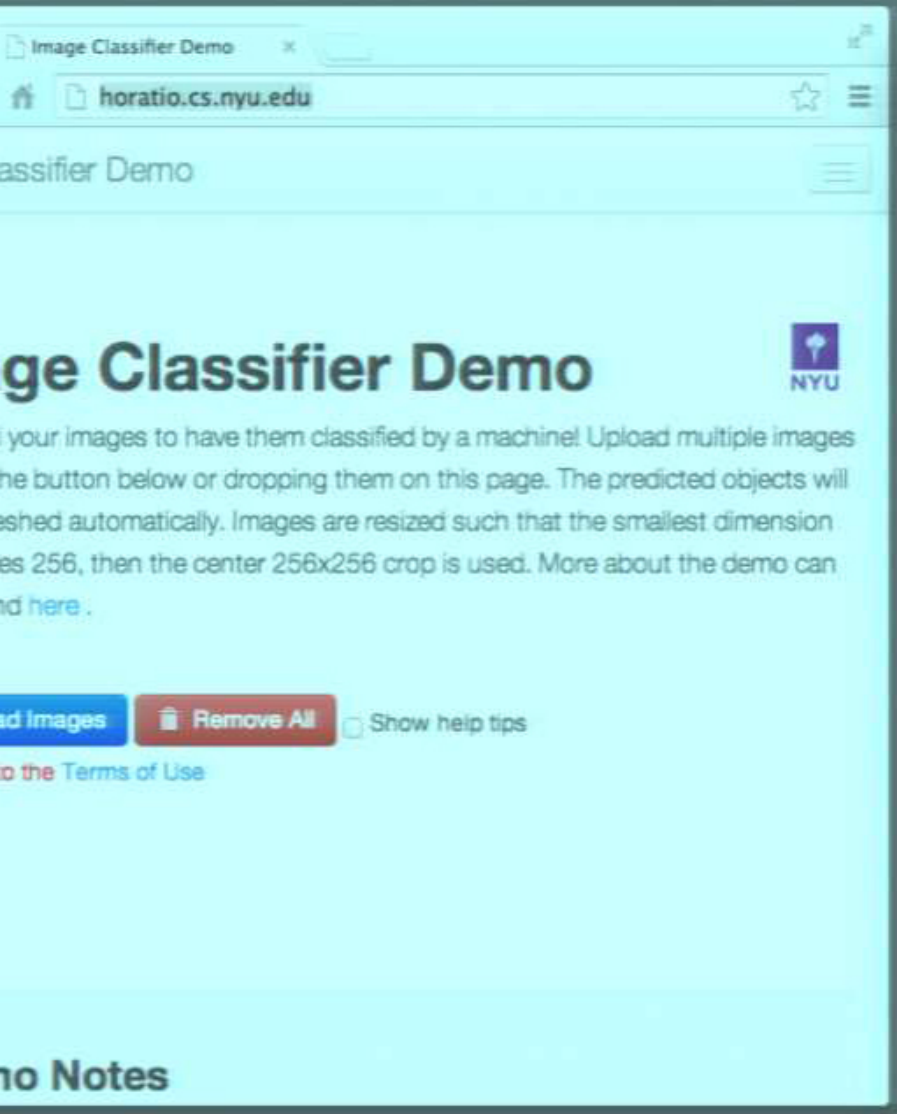
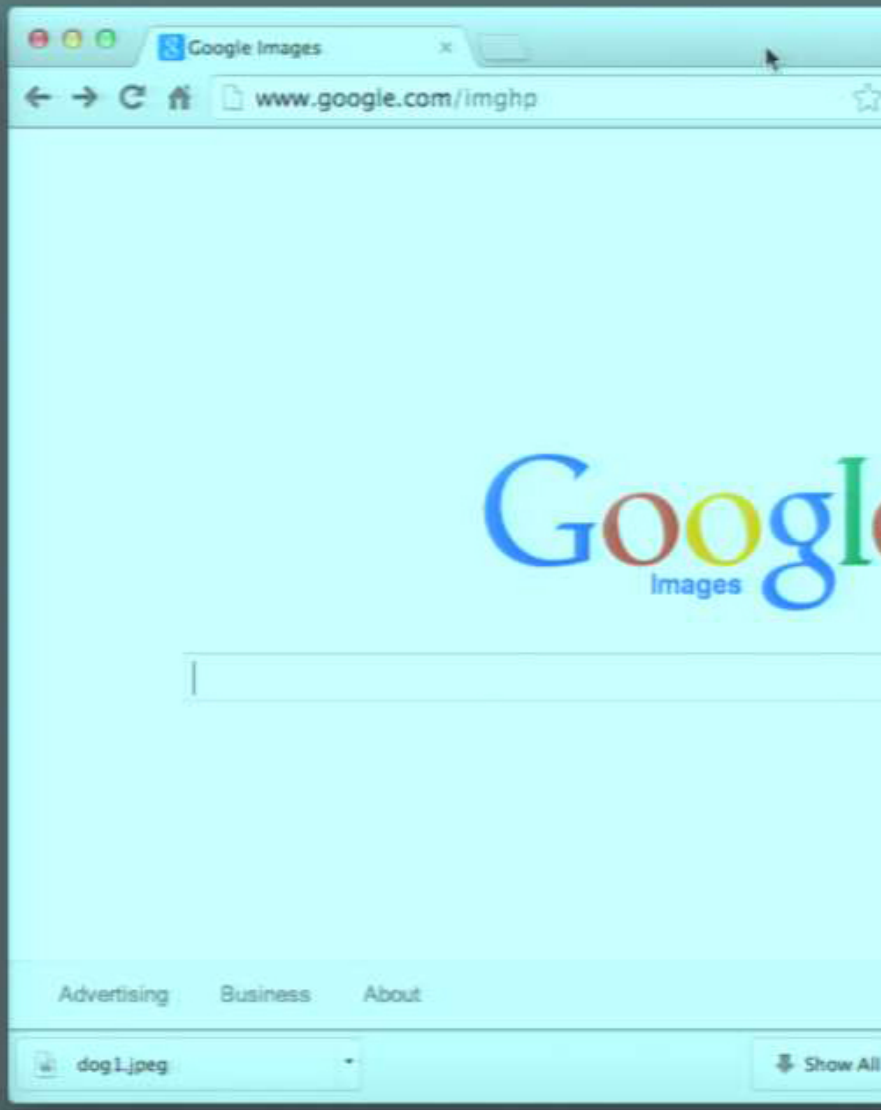


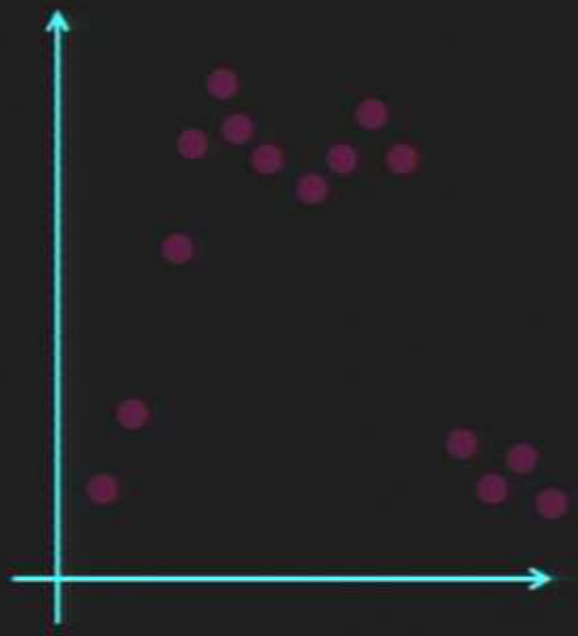
Image Classifier Demo
horatio.cs.nyu.edu
assifier Demo
ge Classifier Demo
NYU
your images to have them classified by a machine! Upload multiple images
he button below or dropping them on this page. The predicted objects will
eshed automatically. Images are resized such that the smallest dimension
es 256, then the center 256x256 crop is used. More about the demo can
d here .
d Images
Remove All
Show help tips
o the Terms of Use
no Notes


Google Images
www.google.com/imghp
Google
Images
Advertising
Business
About
dog1.jpeg
Show All

Improving Generalization

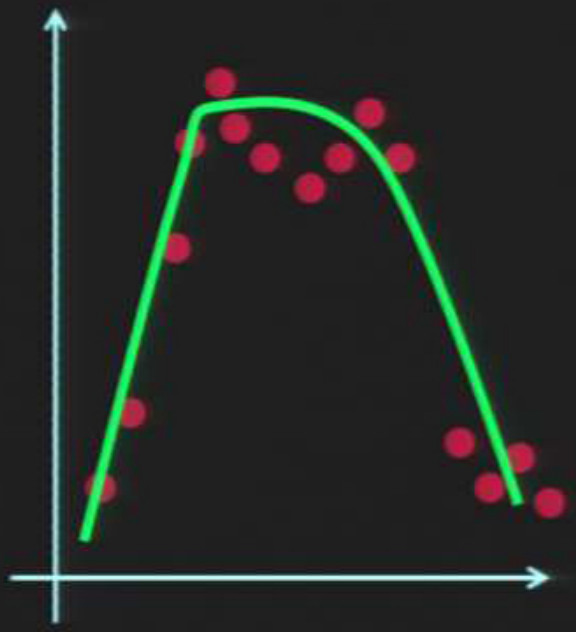
- Data augmentation (crops/flips etc. of images)
- Weight decay (L1 or L2 penalty on weights)
- Inject Noise into network
  - DropOut [Hinton et al. 2012]
  - DropConnect [Wan et al. ICML 2012]
  - Stochastic Pooling [Zeiler & Fergus ICLR'13]

Big Model + Regularize vs Small Model



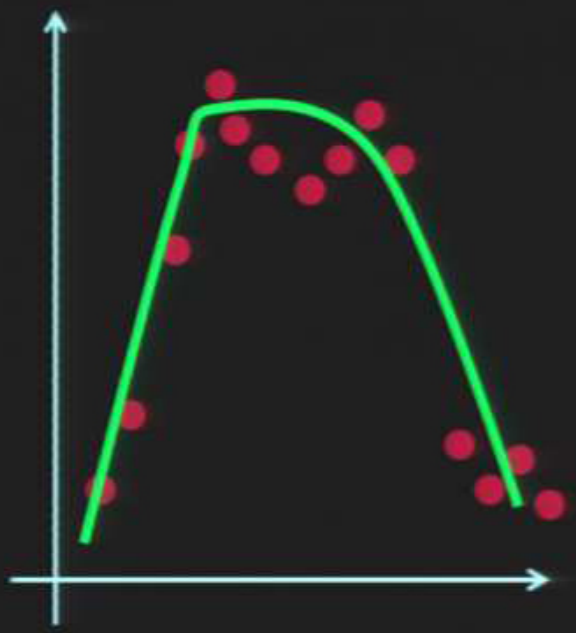
Big Model + Regularize vs Small Model

Small model

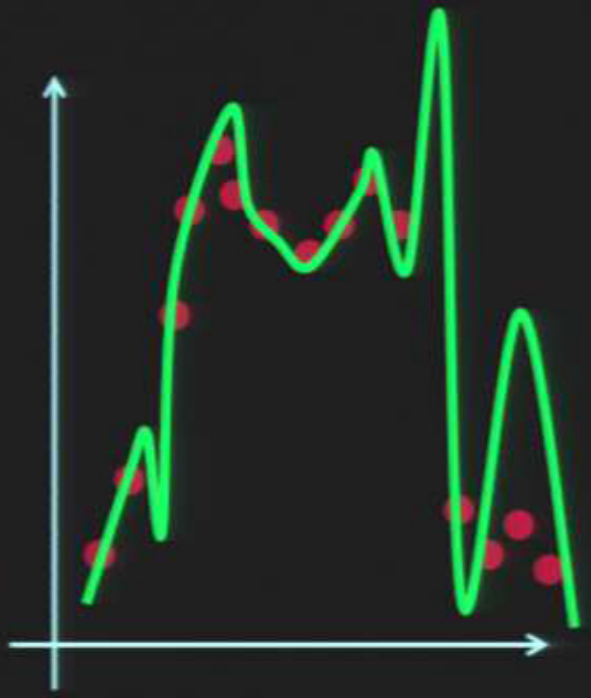


Big Model + Regularize vs Small Model

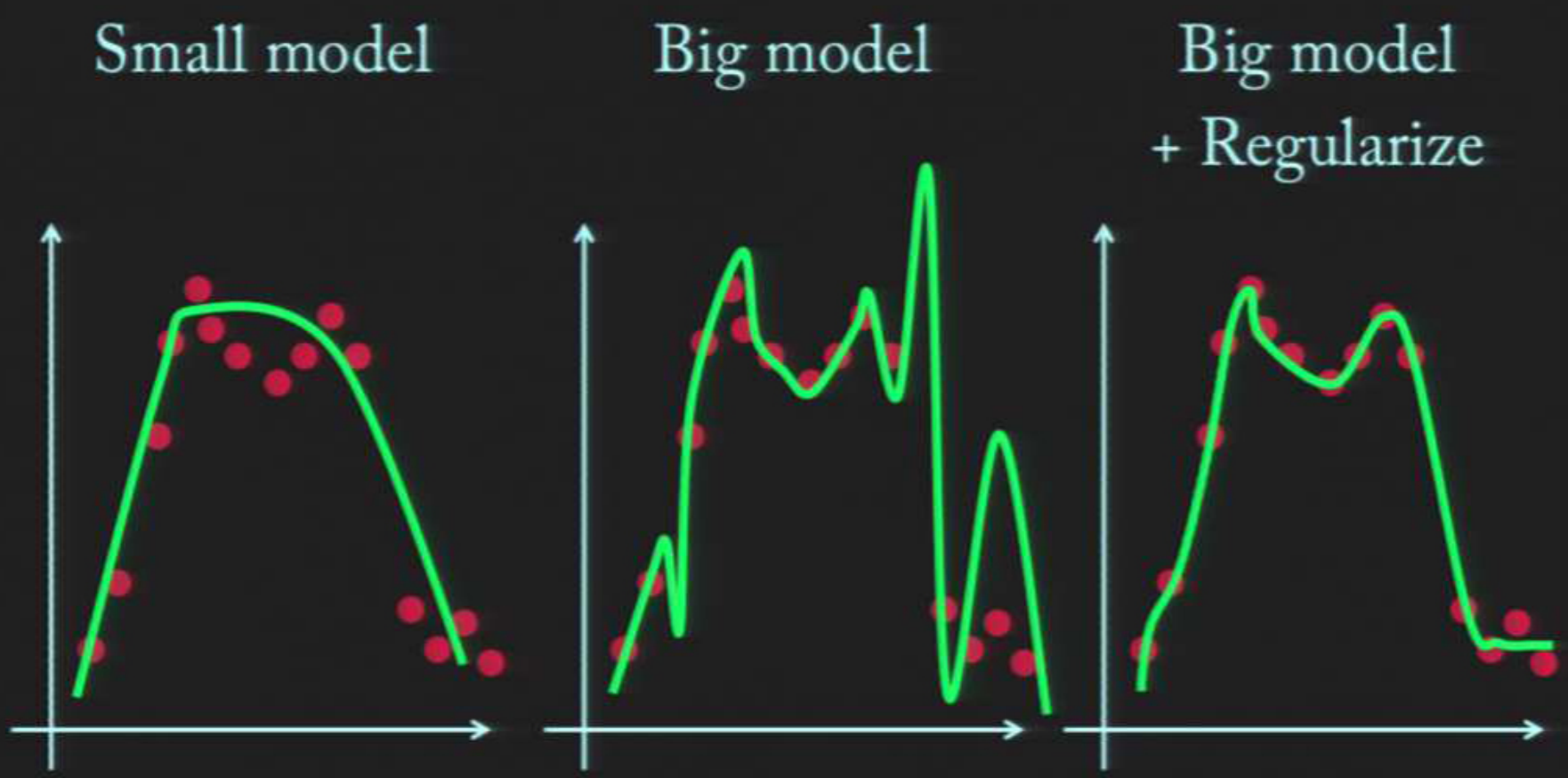
Small model



Big model



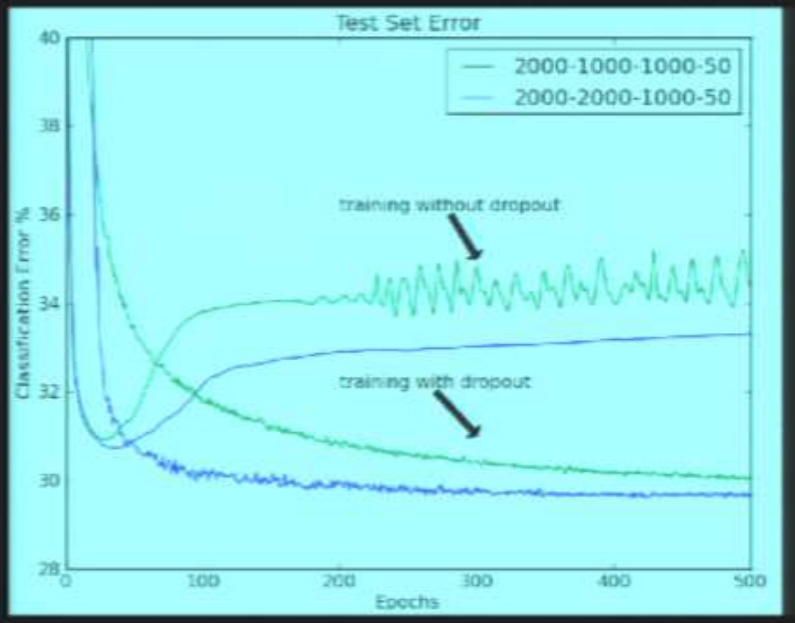
Big Model + Regularize vs Small Model



DropOut

- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, arXiv:1207.0580 2012

- Fully connected layers only
- Randomly set activations in layer to zero
- Gives ensemble of models
- Similar to bagging [Breiman'94], but differs in that parameters are shared.

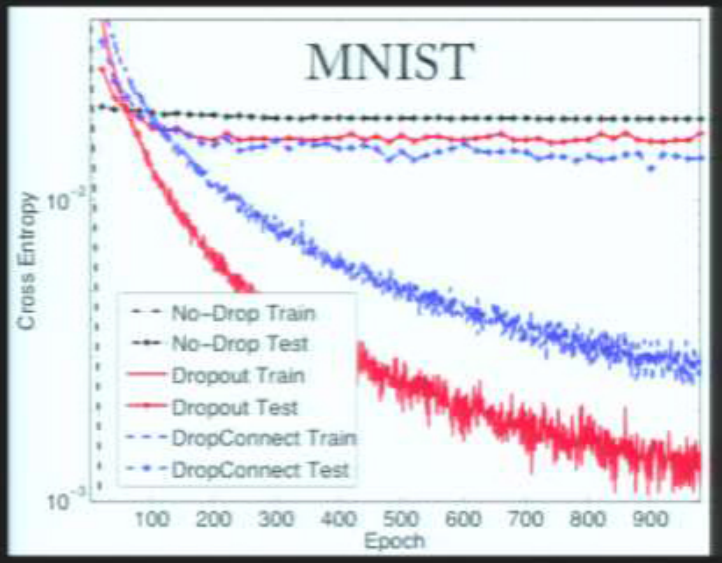


DropConnect

- Wan et al. ICML 2013
- Fully-connected layers only
- Random binary mask on weights

b) DropConnect mask M

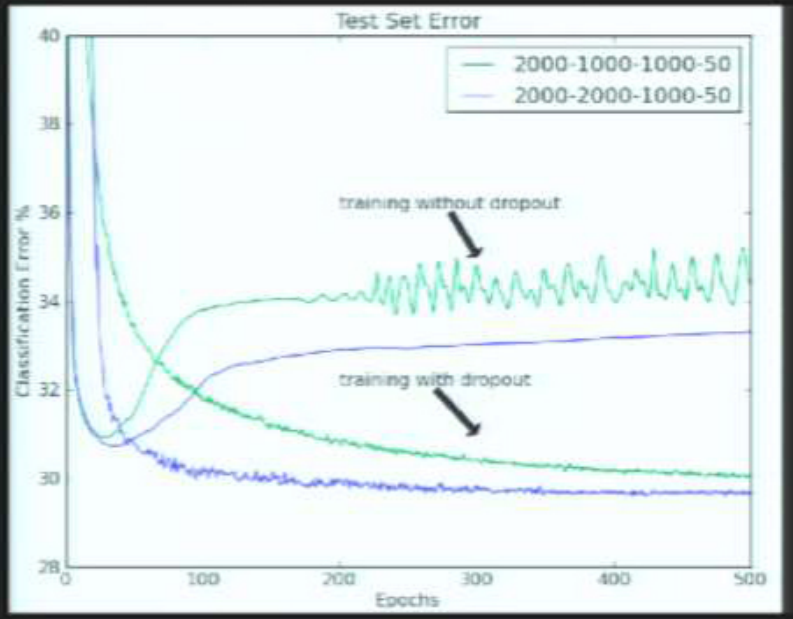
c) Effective Dropout mask M'



DropOut

- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, arXiv:1207.0580 2012

- Fully connected layers only
- Randomly set activations in layer to zero
- Gives ensemble of models
- Similar to bagging [Breiman'94], but differs in that parameters are shared.

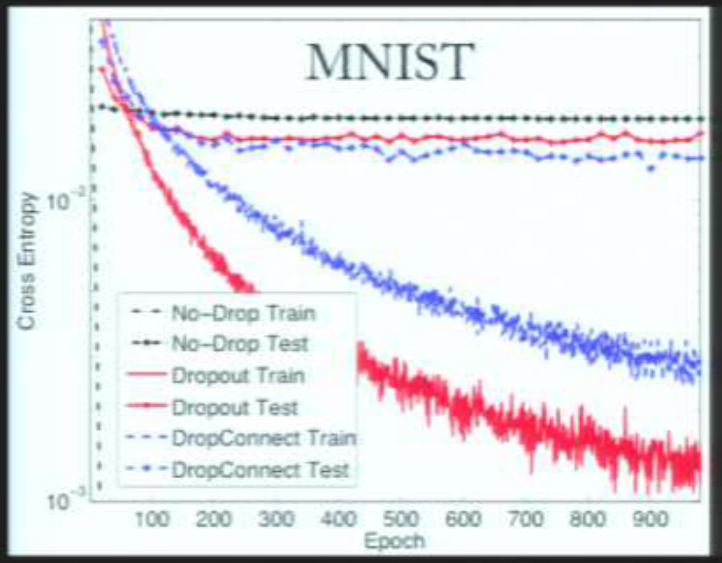


DropConnect

- Wan et al. ICML 2013
- Fully-connected layers only
- Random binary mask on weights

b) DropConnect mask M

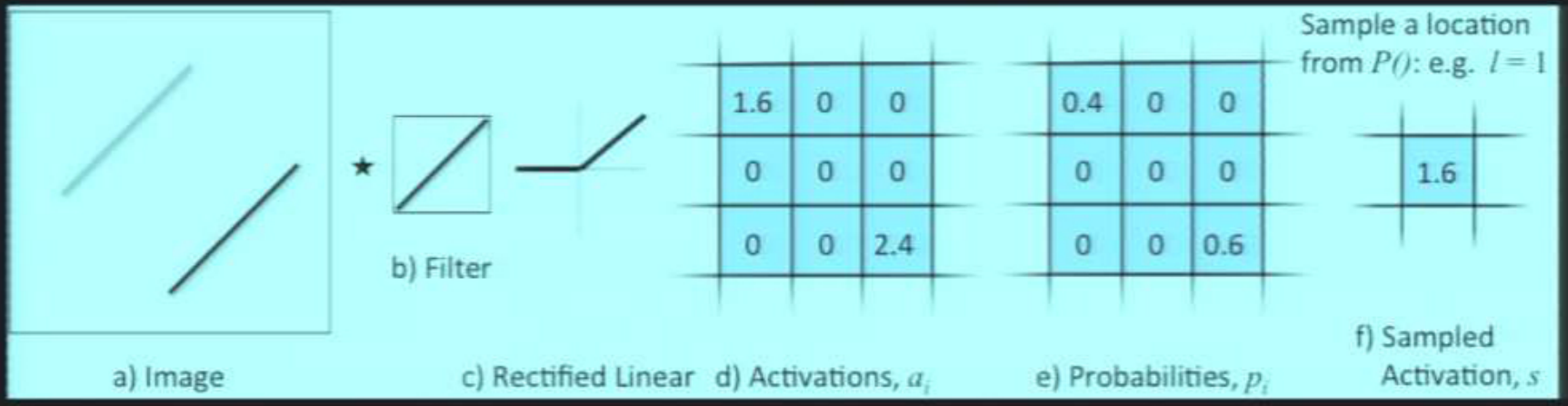
c) Effective Dropout mask M'



Stochastic Pooling

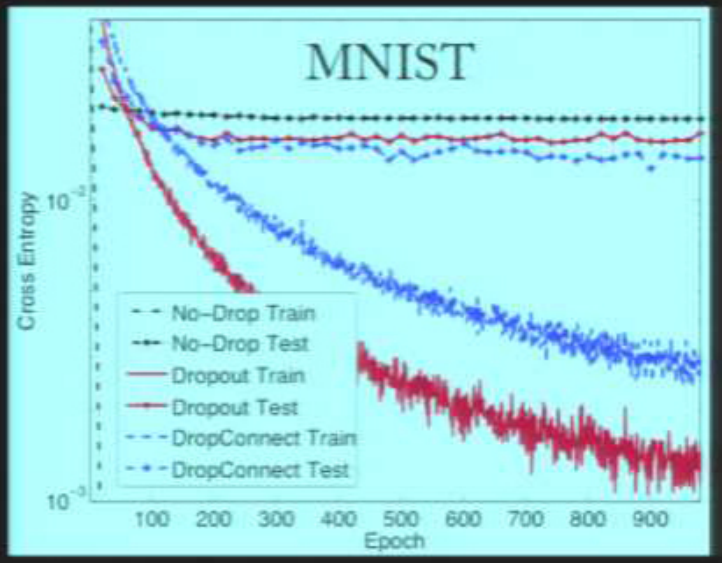
[Zeiler and Fergus, ICLR 2013]

- For conv layers
- Compute activations a_i: (≥ 0)
- Normalize to sum to 1 -> $p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$
- Sample location, l, from multinomial
- Use activation from the location: $s = a_l$



DropConnect

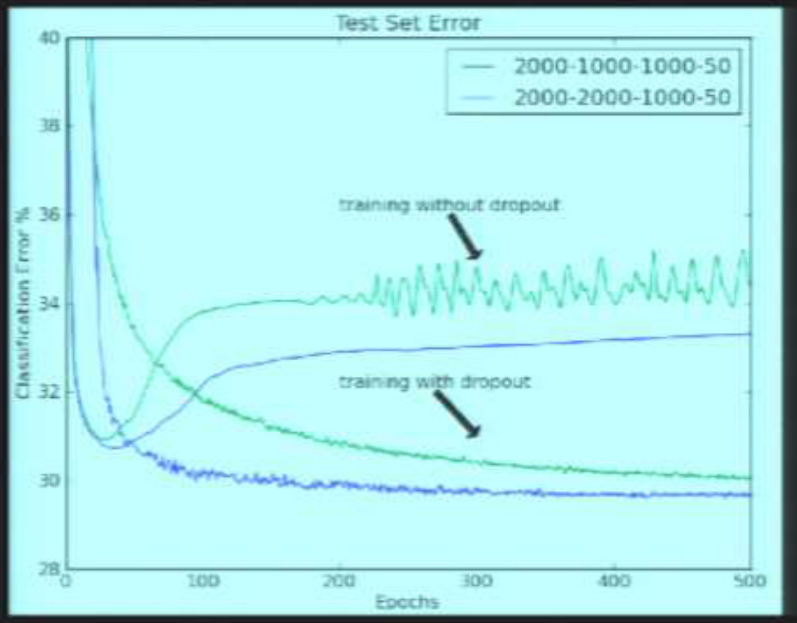
- Wan et al. ICML 2013
- Fully-connected layers only
- Random binary mask on weights



DropOut

- G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever and R. R. Salakhutdinov, *Improving neural networks by preventing co-adaptation of feature detectors*, arXiv:1207.0580 2012

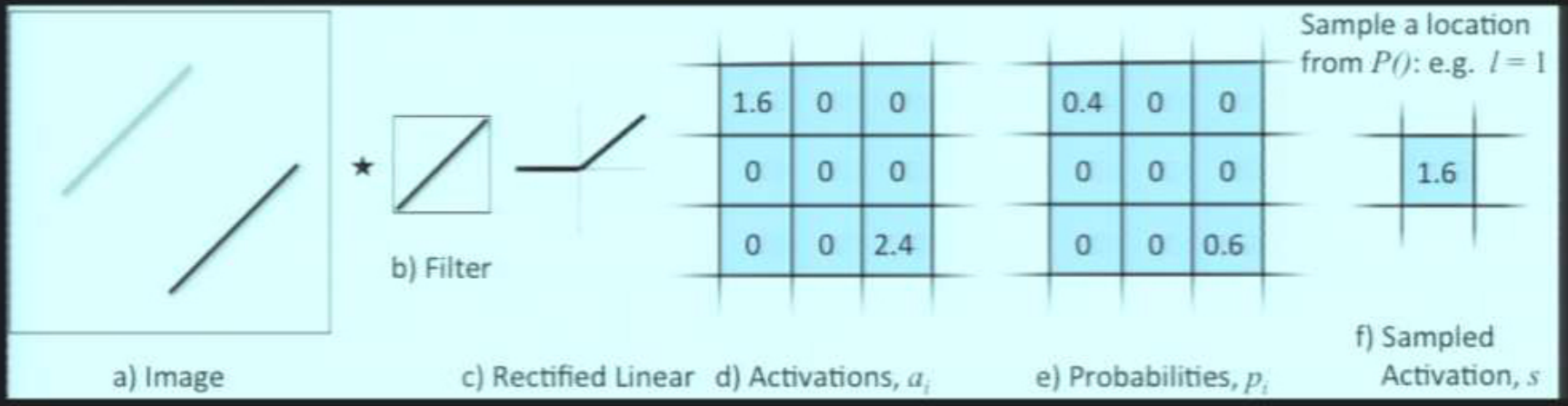
- Fully connected layers only
- Randomly set activations in layer to zero
- Gives ensemble of models
- Similar to bagging [Breiman'94], but differs in that parameters are shared.



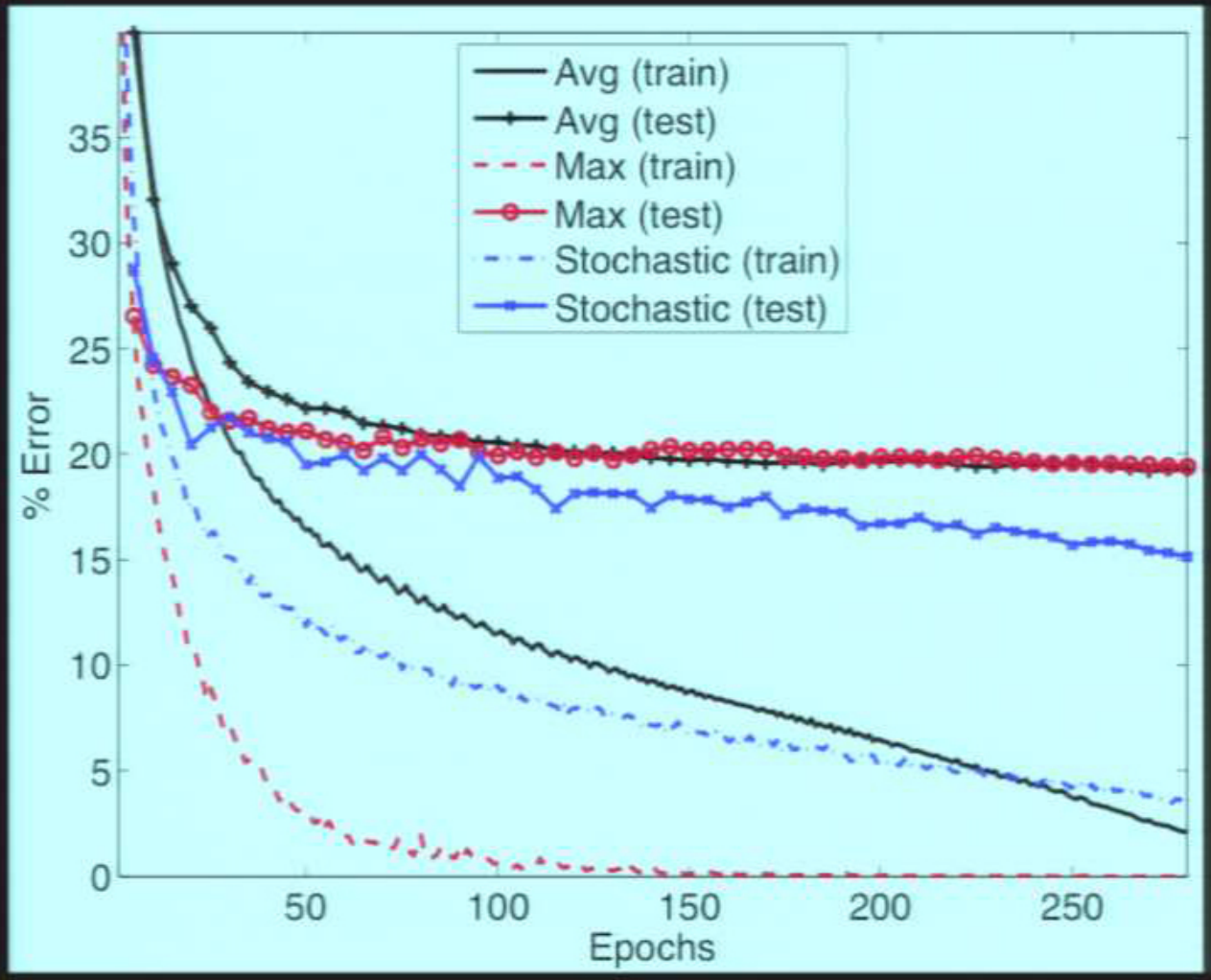
Stochastic Pooling

[Zeiler and Fergus, ICLR 2013]

- For conv layers
- Compute activations a_i: (≥ 0)
- Normalize to sum to 1 -> $p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$
- Sample location, l, from multinomial
- Use activation from the location: $s = a_l$



Stochastic Pooling: CIFAR-10



Other Good Things to Know

- Check gradients numerically by finite differences
- Plot feature maps: should be uncorrelated & high variance

Good training: hidden units are sparse across samples and across features.

Slide credit: M. Ranzato

Other Good Things to Know

- Check gradients numerically by finite differences
- Plot feature maps: should be uncorrelated & high variance

Bad training: many hidden units ignore the input and/or exhibit strong correlations.

Slide credit: M. Ranzato

It's not working – what do I do?

- Training diverges:
  - Learning rate may be too large
    → decrease learning rate
  - BackProp is buggy
    → numerical gradient checking

- Parameters collapse / loss is minimized but accuracy is low
  - Check loss function:
    - Is it appropriate for the task you want to solve?
    - Does it have degenerate solutions?

Slide credit:
M. Ranzato

It's not working – what do I do?

- Network is underperforming
  - Compute flops and # parameters
    → if too small, make net larger
  - Visualize hidden units/parameters
    → fix optimization

- Network is too slow
  - Compute flops and # parameters
    → GPU, distributed framework, make net smaller

Slide credit:
M. Ranzato

It's not working – what do I do?

- Training diverges:
  - Learning rate may be too large
    → decrease learning rate
  - BackProp is buggy
    → numerical gradient checking

- Parameters collapse / loss is minimized but accuracy is low
  - Check loss function:
    - Is it appropriate for the task you want to solve?
    - Does it have degenerate solutions?

Slide credit:
M. Ranzato

It's not working – what do I do?

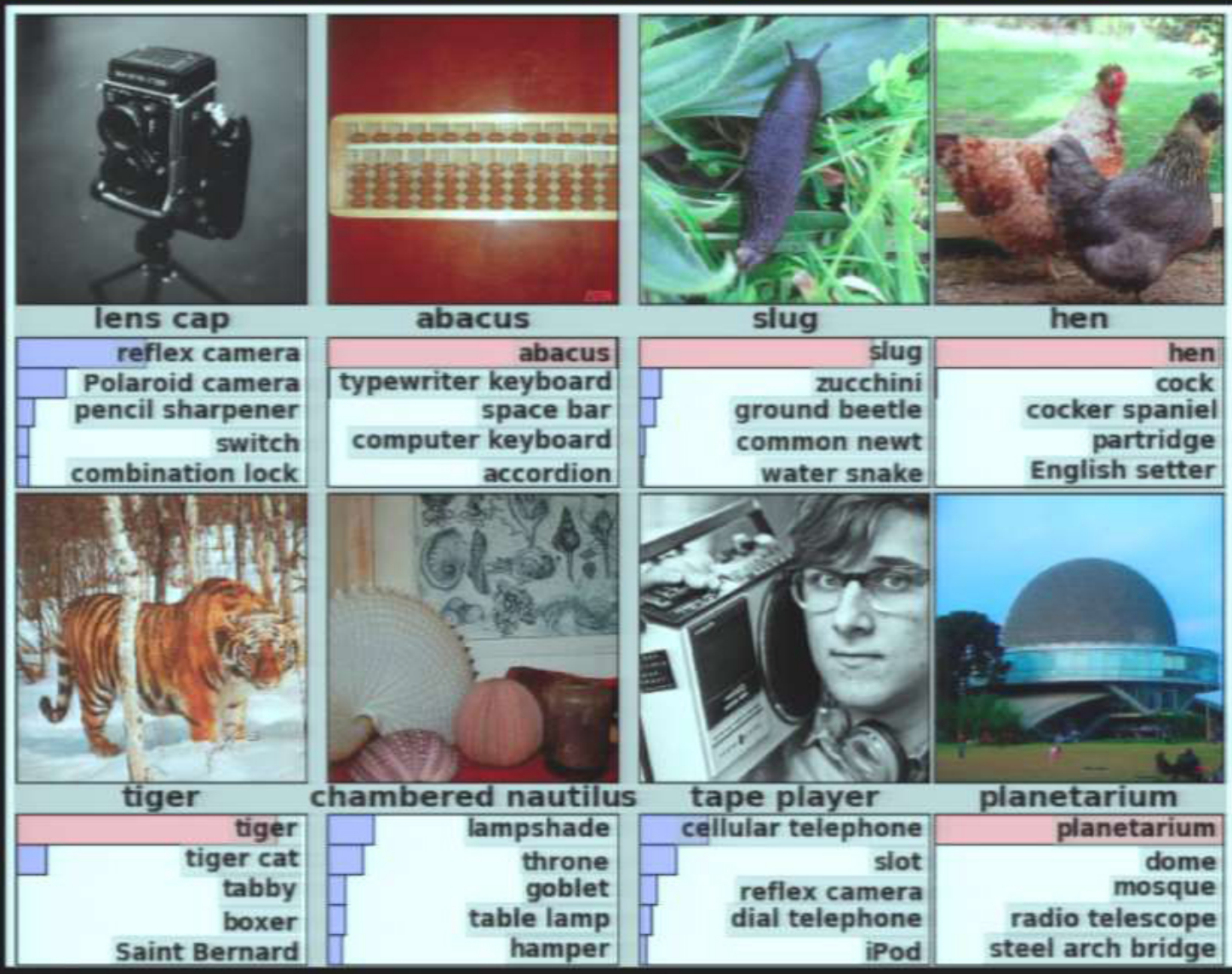
- Network is underperforming
  - Compute flops and # parameters
    → if too small, make net larger
  - Visualize hidden units/parameters
    → fix optimization

- Network is too slow
  - Compute flops and # parameters
    → GPU, distributed framework, make net smaller

Slide credit:
M. Ranzato

Sample Classification Results

[Krizhevsky et al. NIPS'12]



Sample Classification Results

[Krizhevsky et al. NIPS'12]

Object Detection

Detection with ConvNets

- So far, all about classification
- What about localizing objects within the scene?

Groundtruth:
tv or monitor
tv or monitor (2)
tv or monitor (3)
person
remote control
remote control (2)

Occlusion Experiment

- Mask parts of input with occluding square
- Monitor output of classification network

- Perhaps network using scene context?

Occlusion Experiment

- Mask parts of input with occluding square

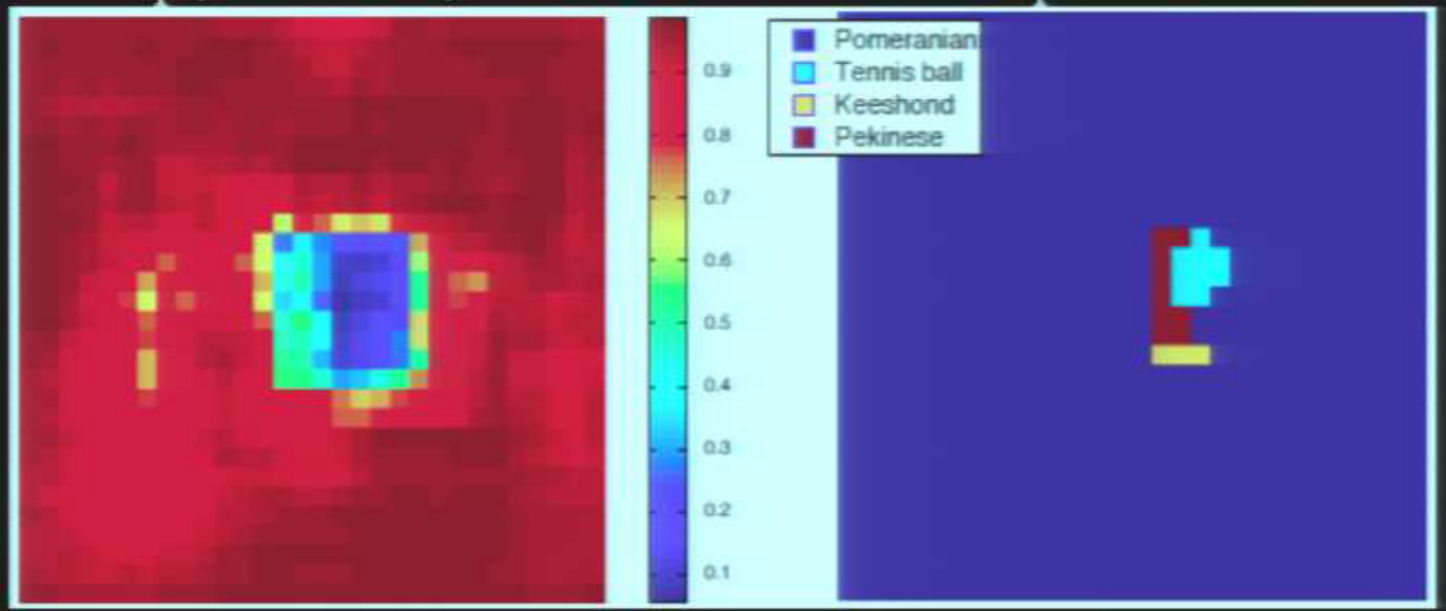
- Monitor output of classification network

- Perhaps network using scene context?

Input image

p(True class)

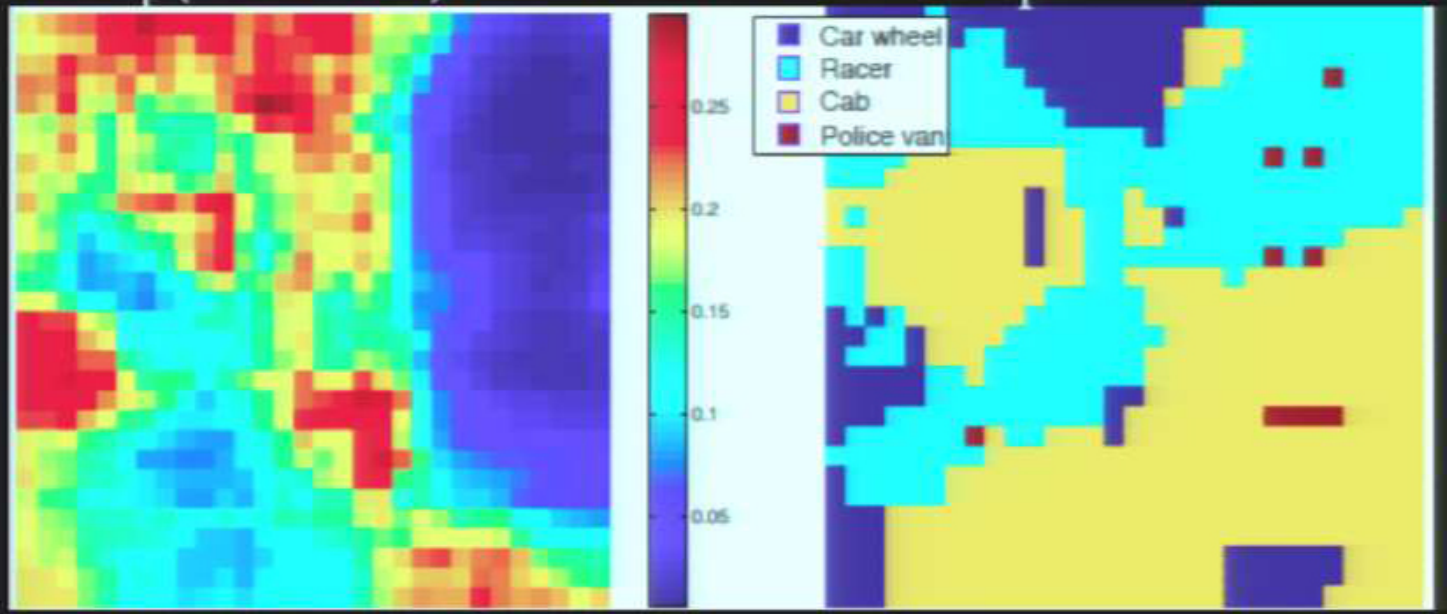
Most probable class



Input image

p(True class)

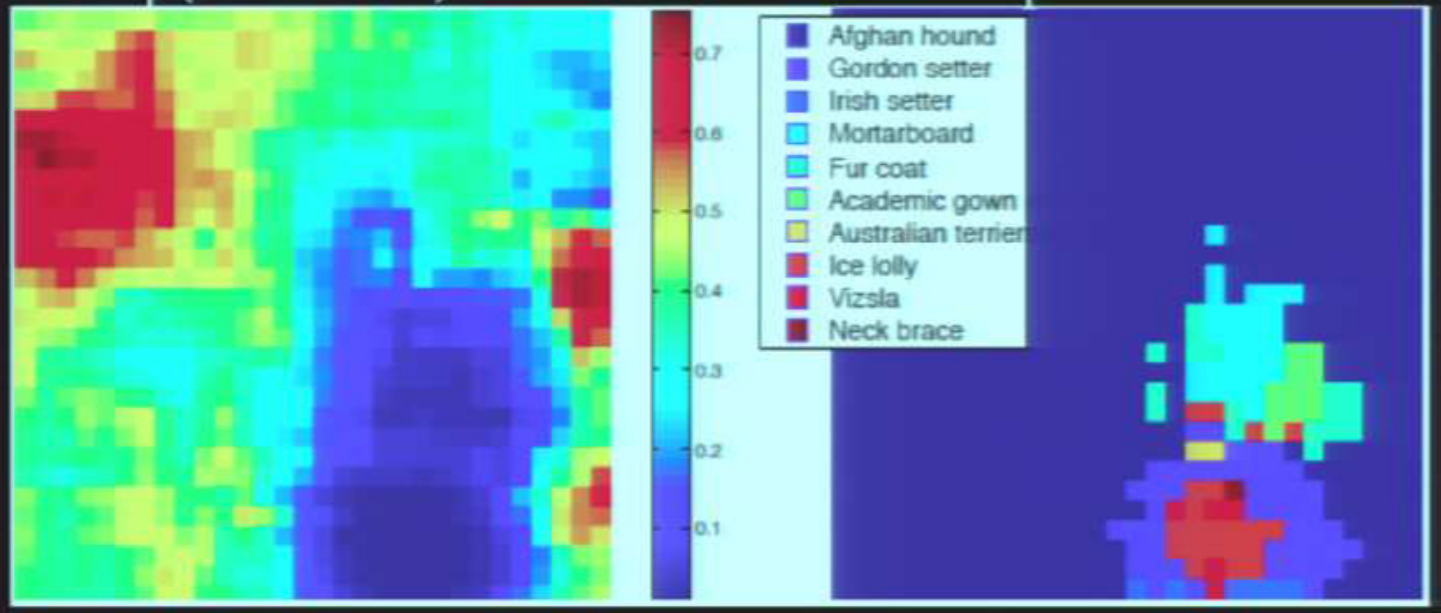
Most probable class



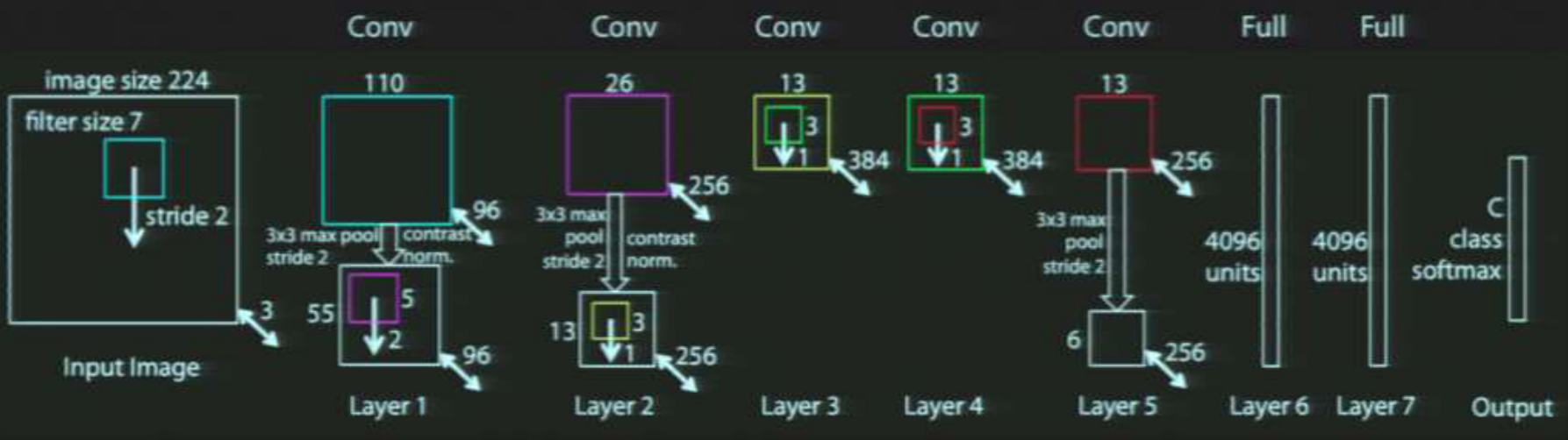
Input image

p(True class)

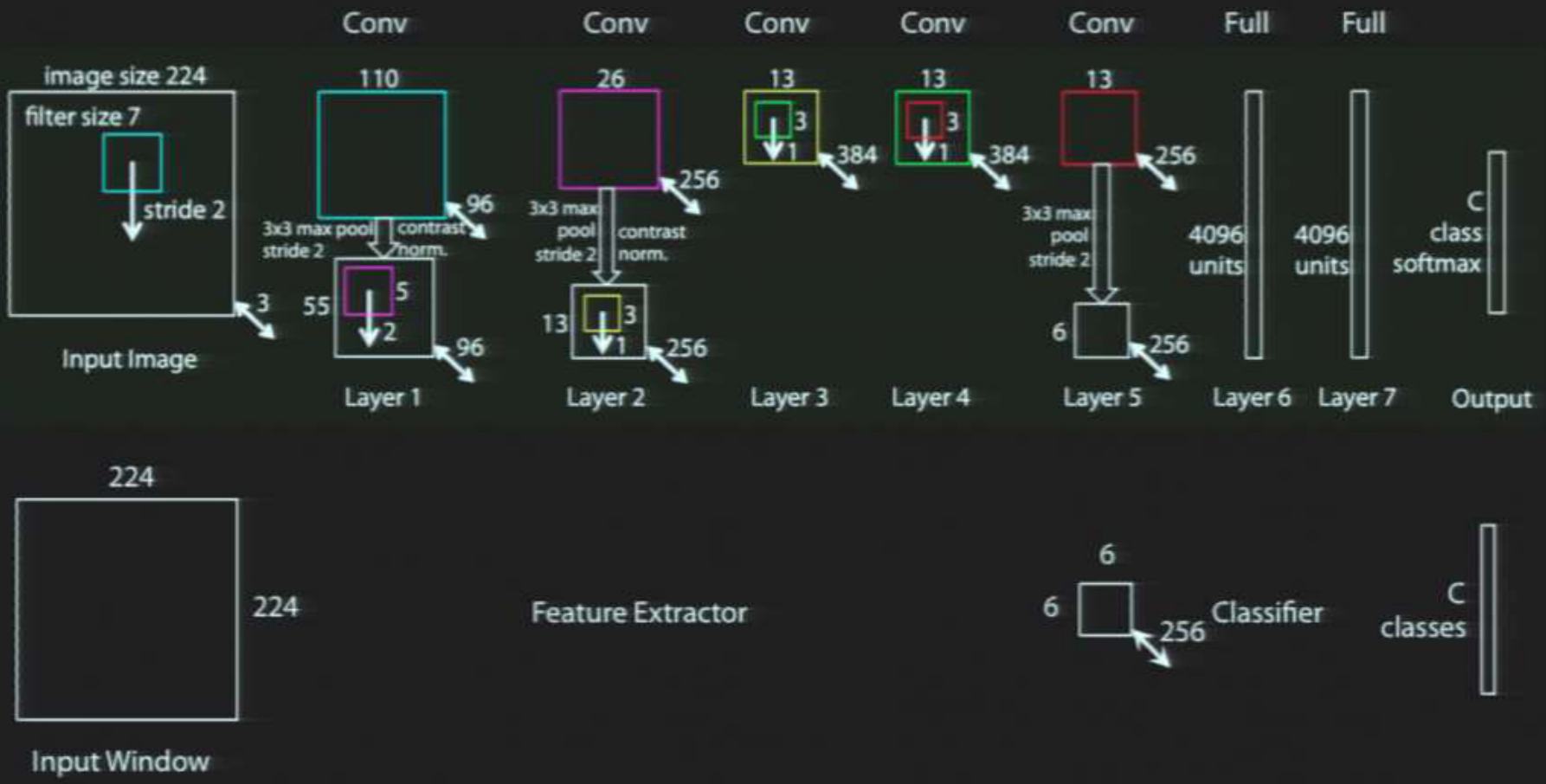
Most probable class



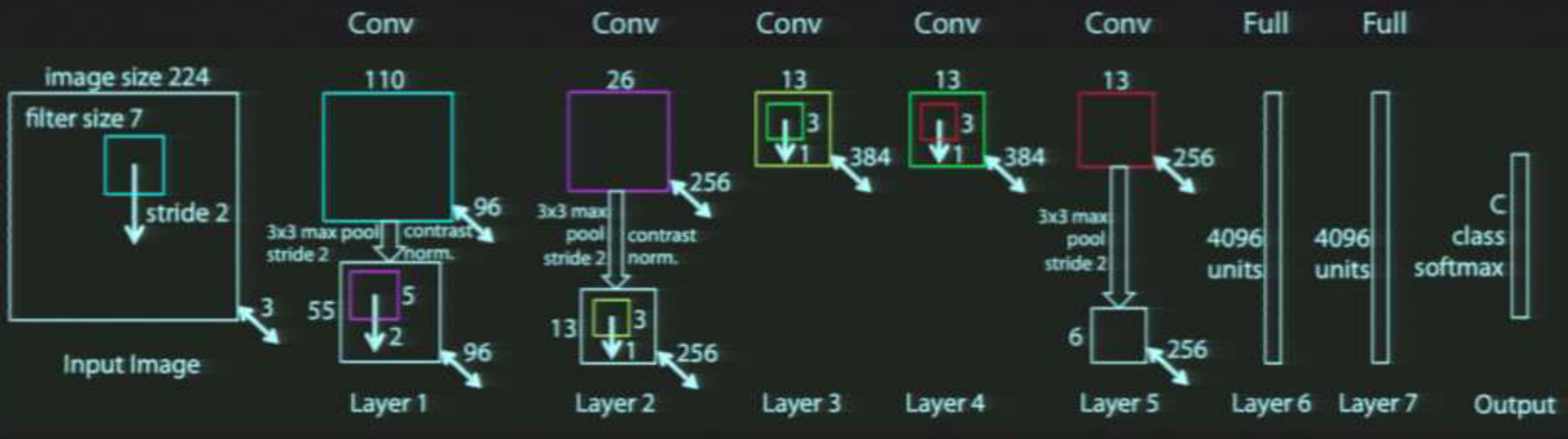
Sliding Window with ConvNet



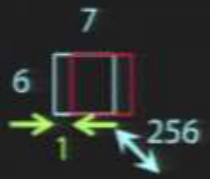
Sliding Window with ConvNet



Sliding Window with ConvNet



Feature Extractor



No need to compute two separate windows
Just one big input window, computed in a single pass

ConvNets for Detection

Feature Extractor

Feature Maps

256

256

256

256

Classifier

Class Maps

C=1000

C=1000

C=1000

C=1000

ConvNets for Detection

Feature Extractor

Feature Maps

256

256

256

Classifier

Class Maps

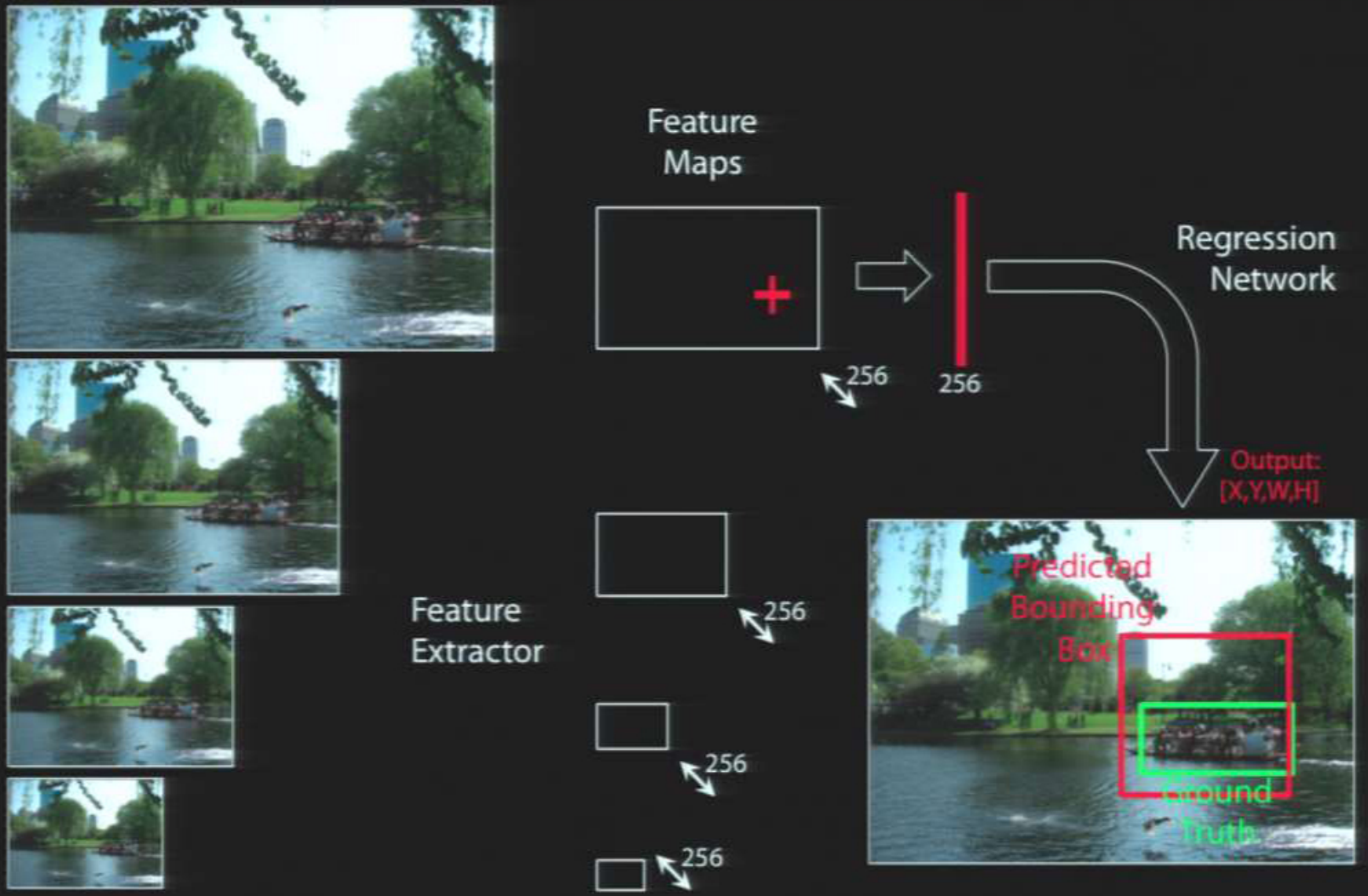
Boat

Boat

Boat

Boat

ConvNets for Detection



ConvNets for Detection

Feature Extractor

Feature Maps

256

256

256

Classifier

Class Maps

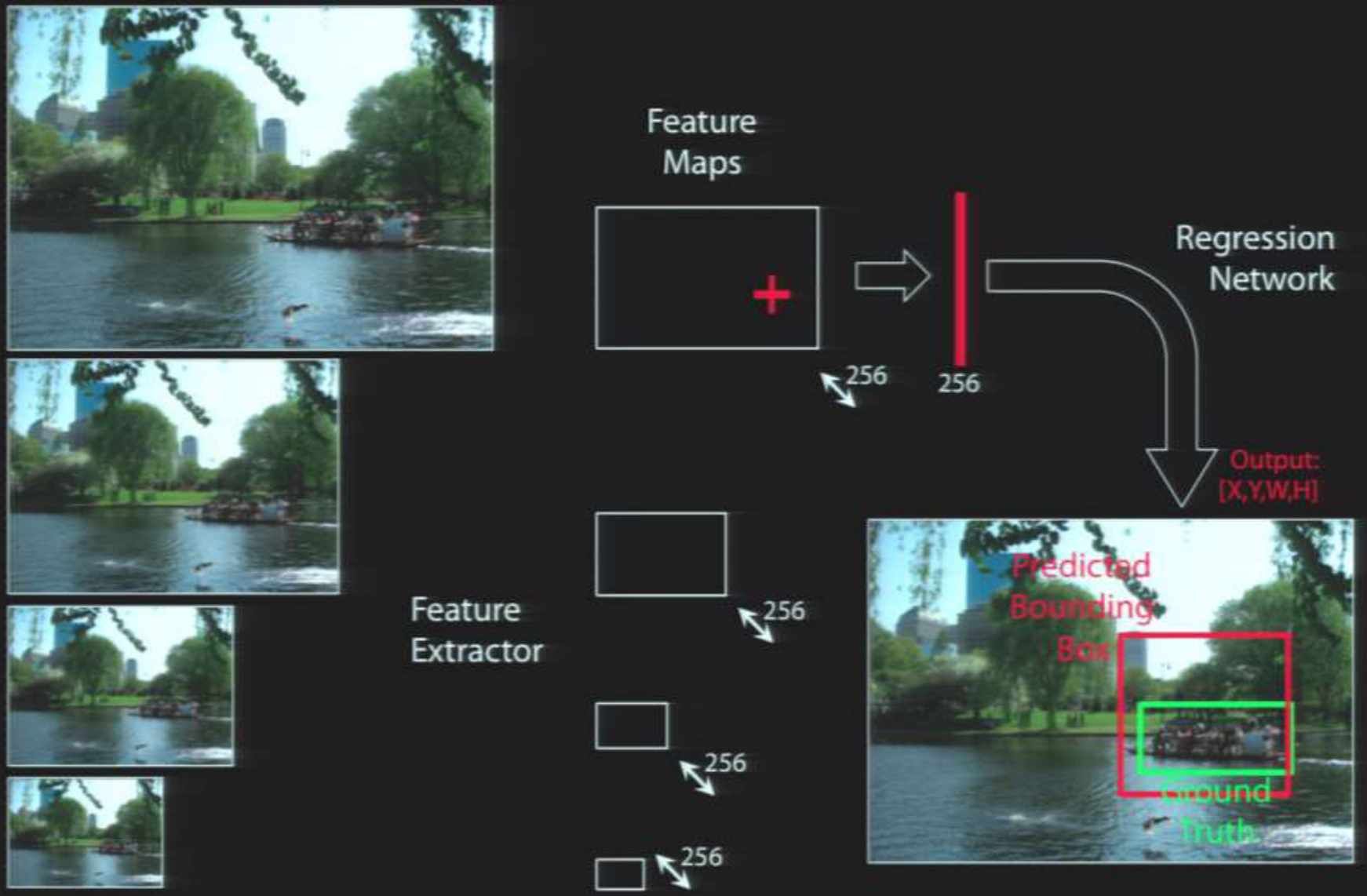
Boat

Boat

Boat

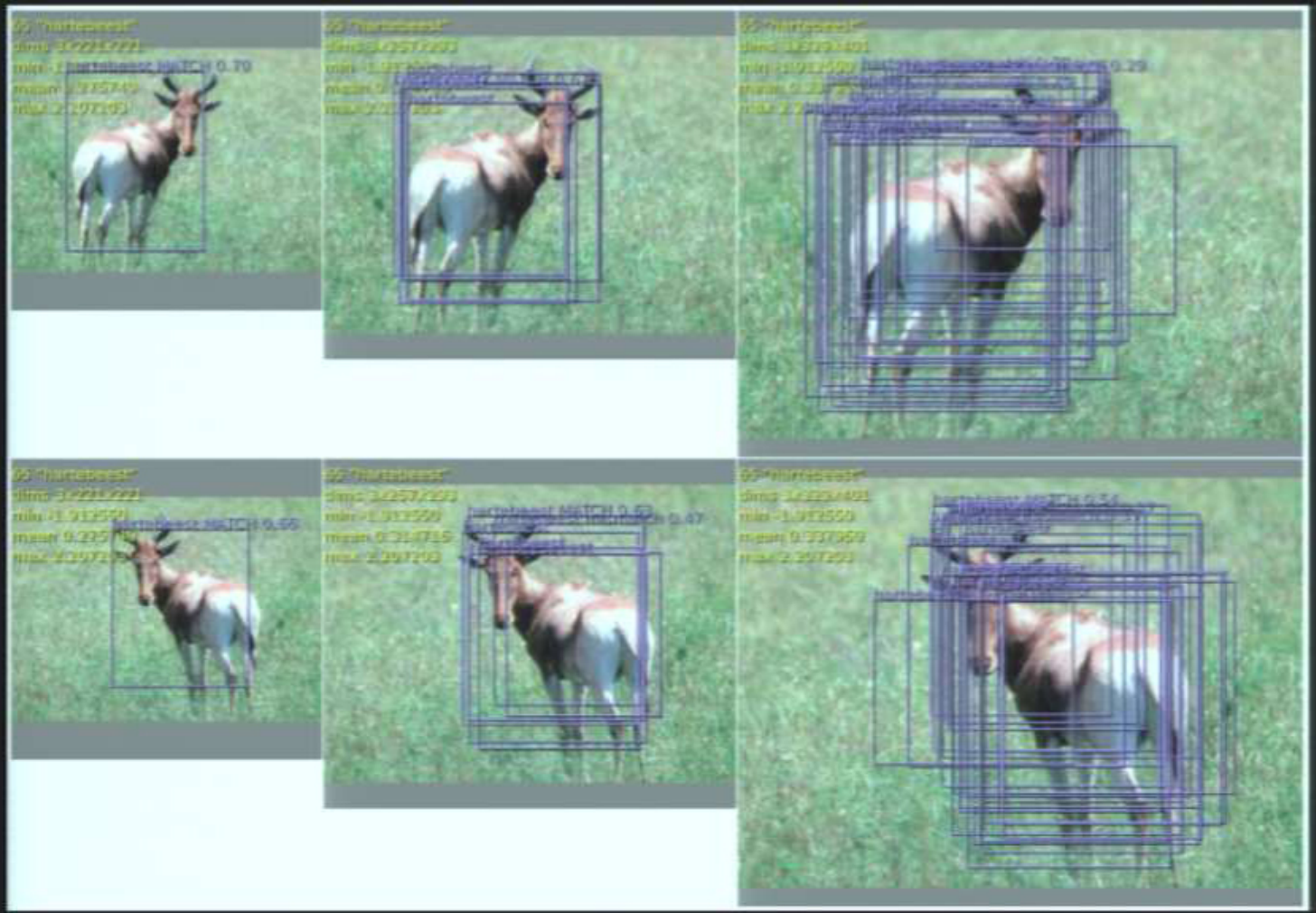
Boat

ConvNets for Detection



Bounding Box prediction example

[Sermanet et al. CVPR'14, under review]



ConvNets for Detection

Feature Extractor

Feature Maps

256

256

256

Classifier

Class Maps

Boat

Boat

Boat

Boat

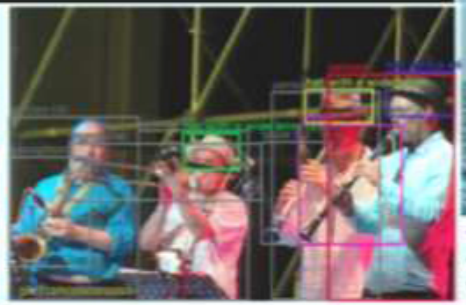
Bounding Box prediction example

[Sermanet et al. CVPR'14, under review]

Detection Results

[Sermanet et al. CVPR'14, under review]

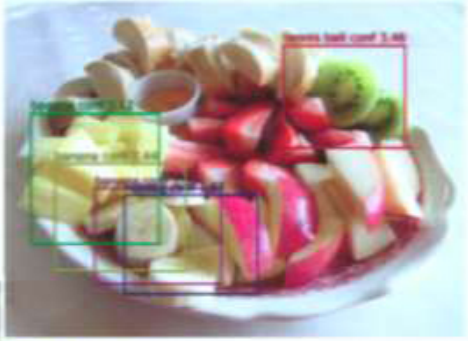
Top predictions:
trombone (confidence 26.8)
oboe (confidence 17.5)
oboe (confidence 11.5)



Groundtruth:
person
hat with a wide brim
hat with a wide brim (2)
hat with a wide brim (3)
oboe
oboe (2)
saxophone
trombone
person (2)
person (3)
person (4)

Top predictions:
watercraft (confidence 72.2)
watercraft (confidence 2.1)

Groundtruth:
watercraft
watercraft (2)



Top predictions:
tennis ball (confidence 3.5)
banana (confidence 2.4)
banana (confidence 2.1)
hotdog (confidence 2.0)
banana (confidence 1.9)

Groundtruth:
strawberry
strawberry (2)
strawberry (3)
strawberry (4)
strawberry (5)
strawberry (6)
strawberry (7)
strawberry (8)
strawberry (9)
strawberry (10)
apple
apple (2)
apple (3)

Top predictions:
microwave (confidence 5.6)
refrigerator (confidence 2.5)

Groundtruth:
bowl
microwave

Detection Results

[Sermanet et al. CVPR'14, under review]

Top predictions:

tv or monitor (confidence 11.5)
person (confidence 4.5)
miniskirt (confidence 3.1)

ILSVRC2012_val_00000119.JPEG

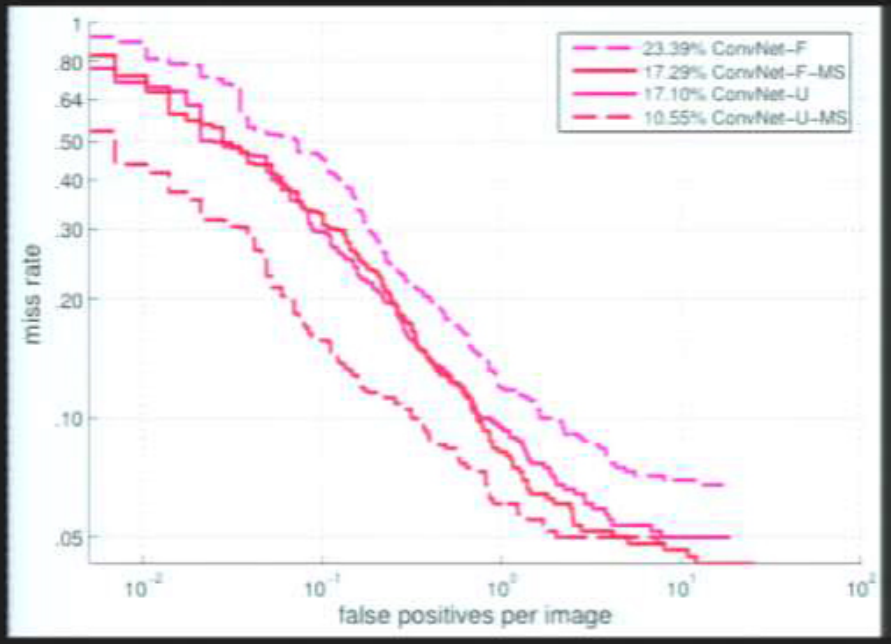
Groundtruth:

tv or monitor
tv or monitor (2)
tv or monitor (3)
person
remote control
remote control (2)

Pedestrian Detection

Sermanet et al. "Pedestrian detection with unsupervised multi-stage.." CVPR 2013

- Model helped by unsupervised pre-training



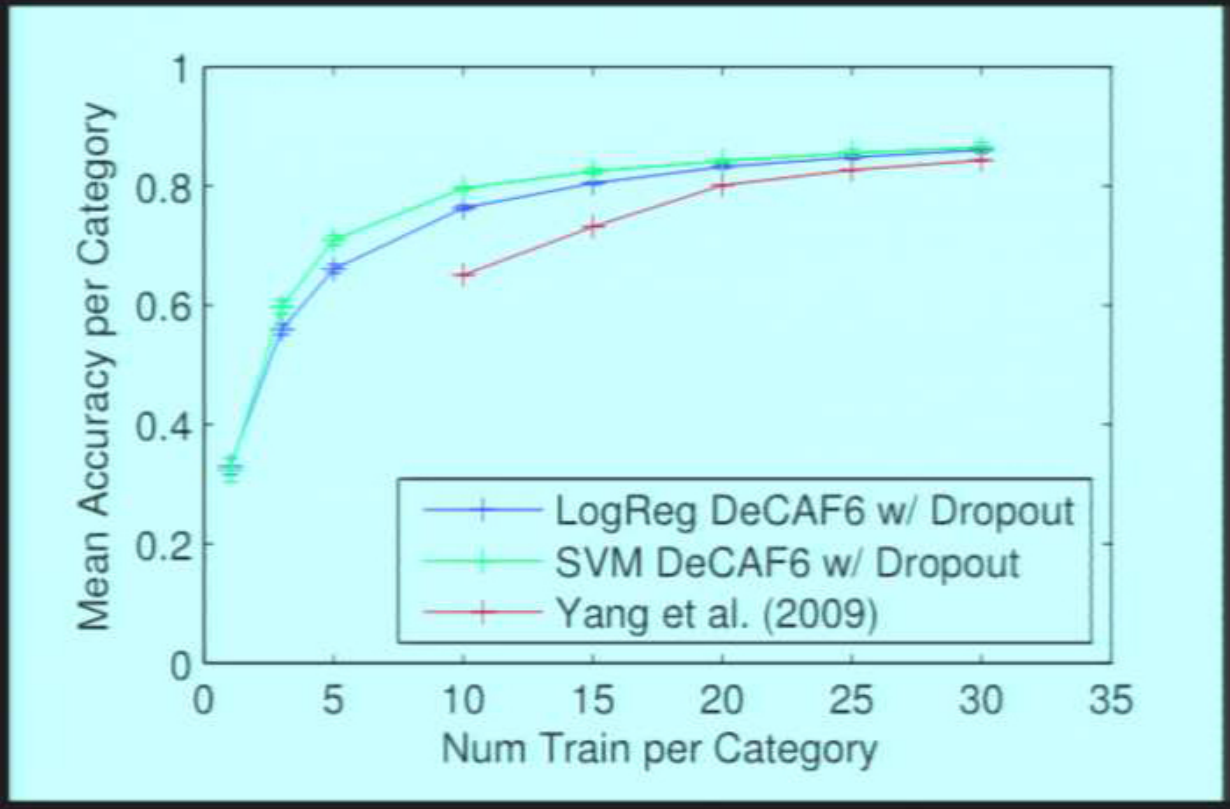
Feature Generalization

Using Features on Other Datasets

- Train model on ImageNet 2012 training set
- Re-train classifier on new dataset
  - Just the softmax layer
- Classify test set of new dataset

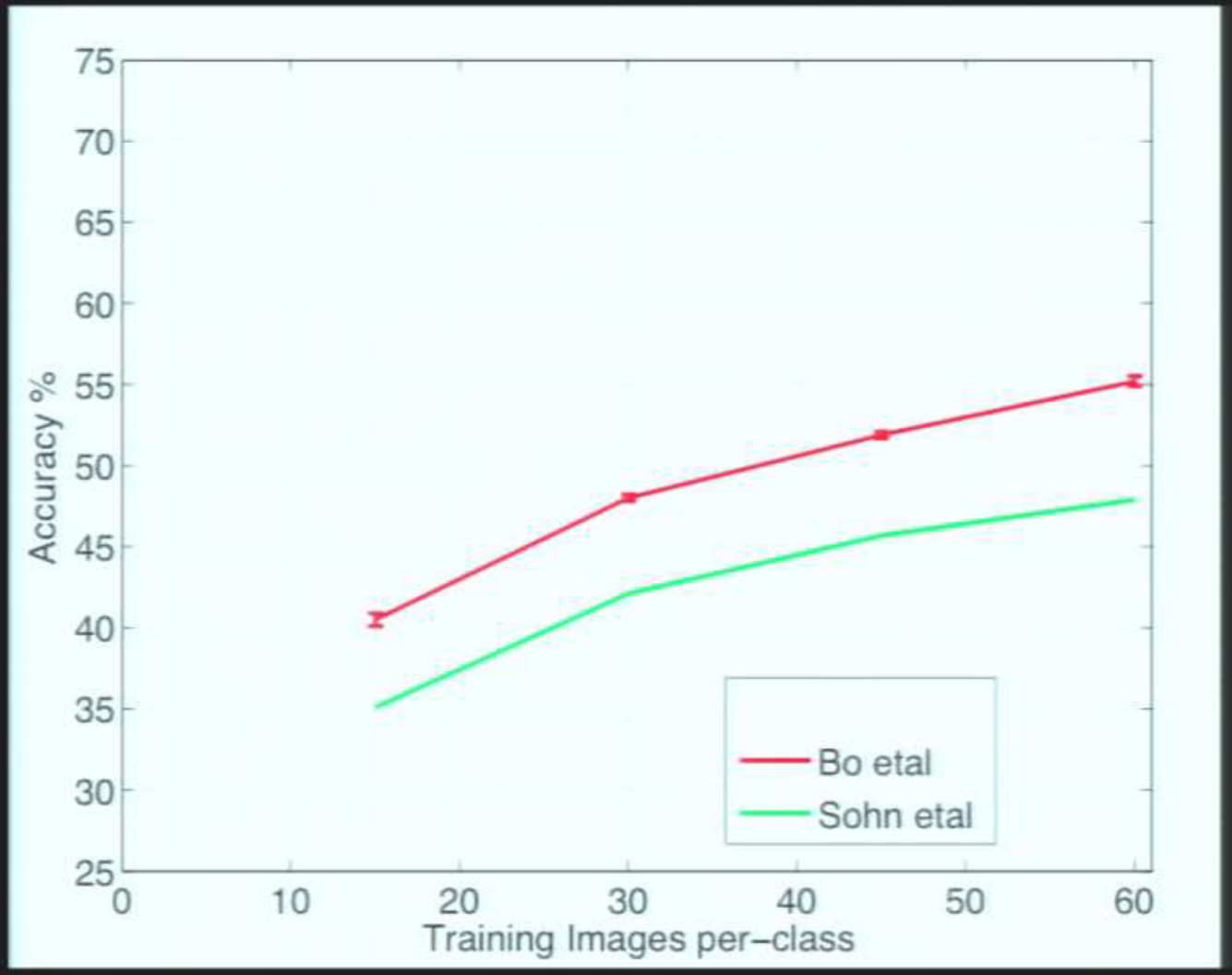
Caltech-101

Donahue et al., *DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition*, arXiv 1310.1531, 2013



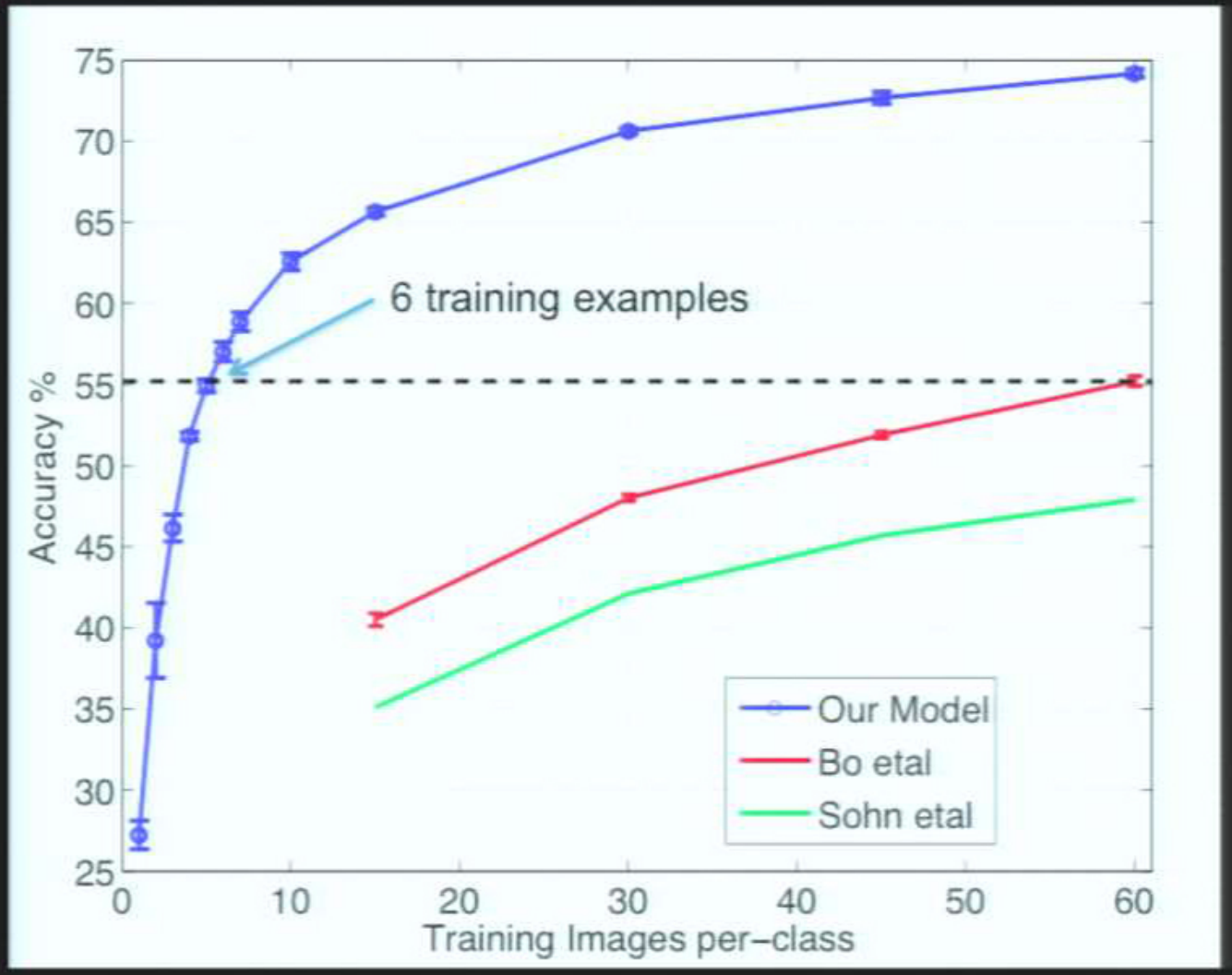
Caltech 256

Zeiler & Fergus, *Visualizing and Understanding Convolutional Networks*, arXiv 1311.2901, 2013



Caltech 256

Zeiler & Fergus, *Visualizing and Understanding Convolutional Networks*, arXiv 1311.2901, 2013



Caltech 256

Zeiler & Fergus, *Visualizing and Understanding Convolutional Networks*, arXiv 1311.2901, 2013

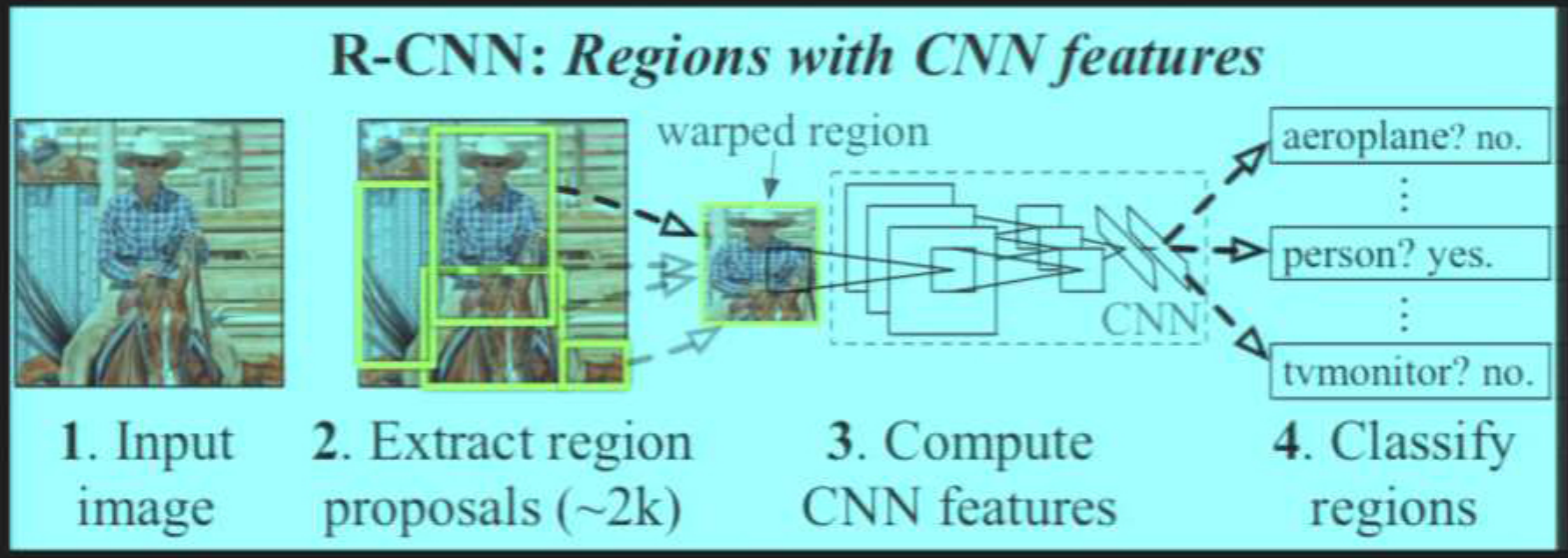
# Train	Acc % 15/class	Acc % 30/class	Acc % 45/class	Acc % 60/class
Sohn *et al.* [16]	35.1	42.1	45.7	47.9
Bo *et al.* [3]	40.5 ± 0.4	48.0 ± 0.2	51.9 ± 0.2	55.2 ± 0.3
Non-pretr.	9.0 ± 1.4	22.5 ± 0.7	31.2 ± 0.5	38.8 ± 1.4
ImageNet-pretr.	$\mathbf{65.7 \pm 0.2}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{72.7 \pm 0.4}$	$\mathbf{74.2 \pm 0.3}$

[3] L. Bo, X. Ren, and D. Fox. Multipath sparse coding using hierarchical matching pursuit. In CVPR, 2013.

[16] K. Sohn, D. Jung, H. Lee, and A. Hero III. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In ICCV, 2011.

PASCAL VOC Detection

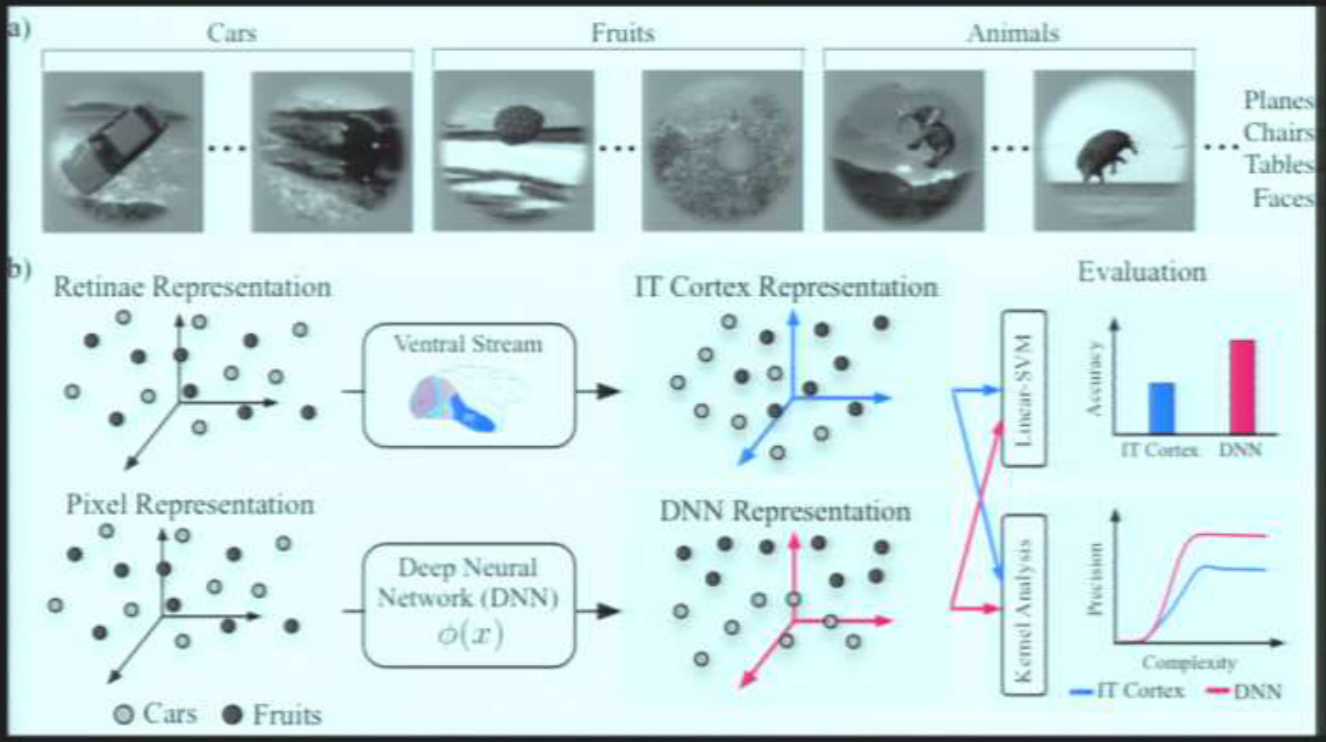
Girshick et al., Rich feature hierarchies for accurate object detection and semantic segmentation, arXiv 1311.2524, 2013



- 43.1% mean AP vs previous 35.1%

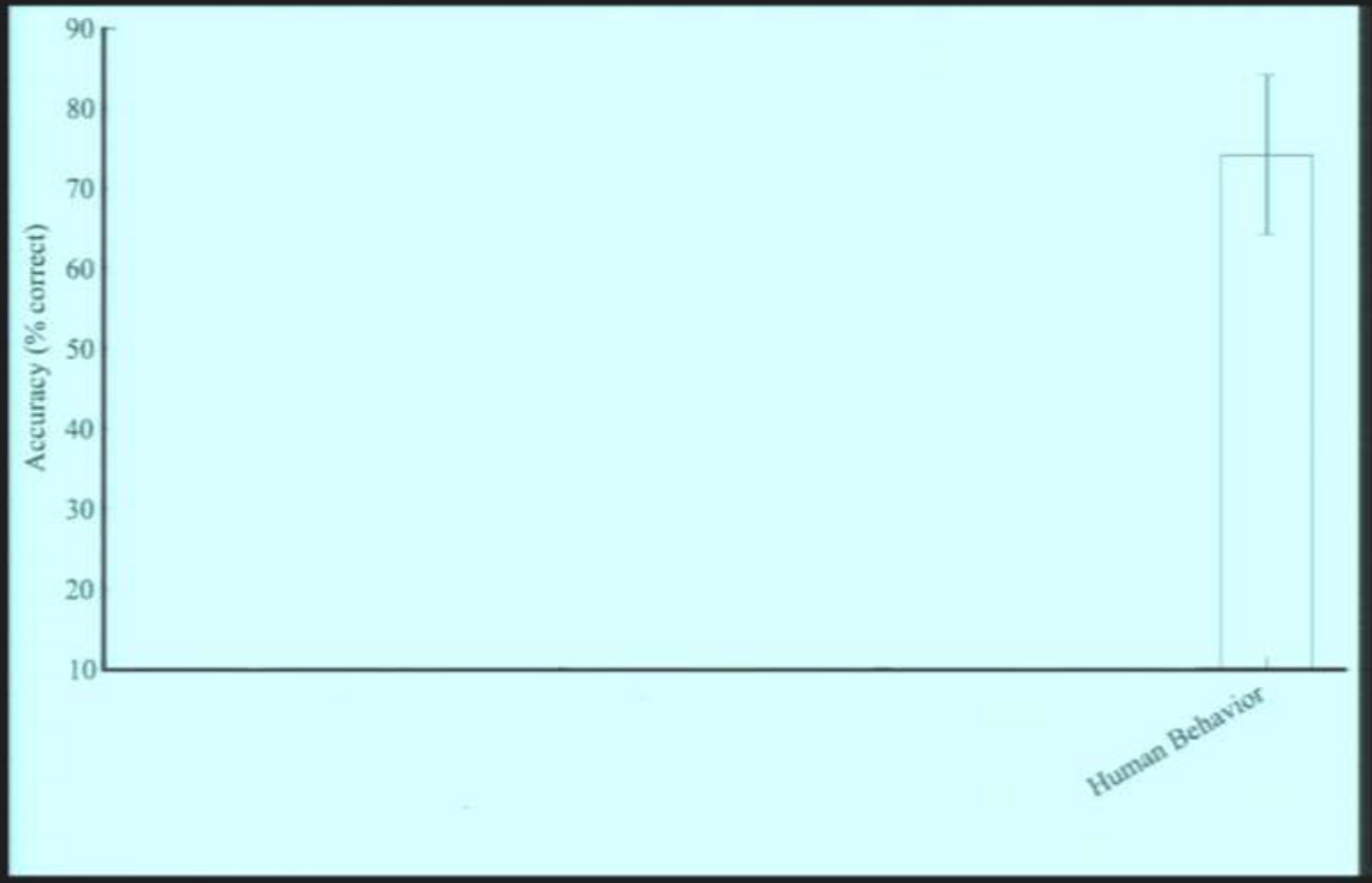
Deep Nets vs Monkey vs Human

C.F. Cadieu, H. Hong, D. Yamins, N. Pinto, E.A. Solomon, N.J. Majaj, and J.J. DiCarlo. *Deep Neural Networks Rival the Object Recognition Performance of the Primate Visual System*. (PLOS One Biology, in submission, 2013).



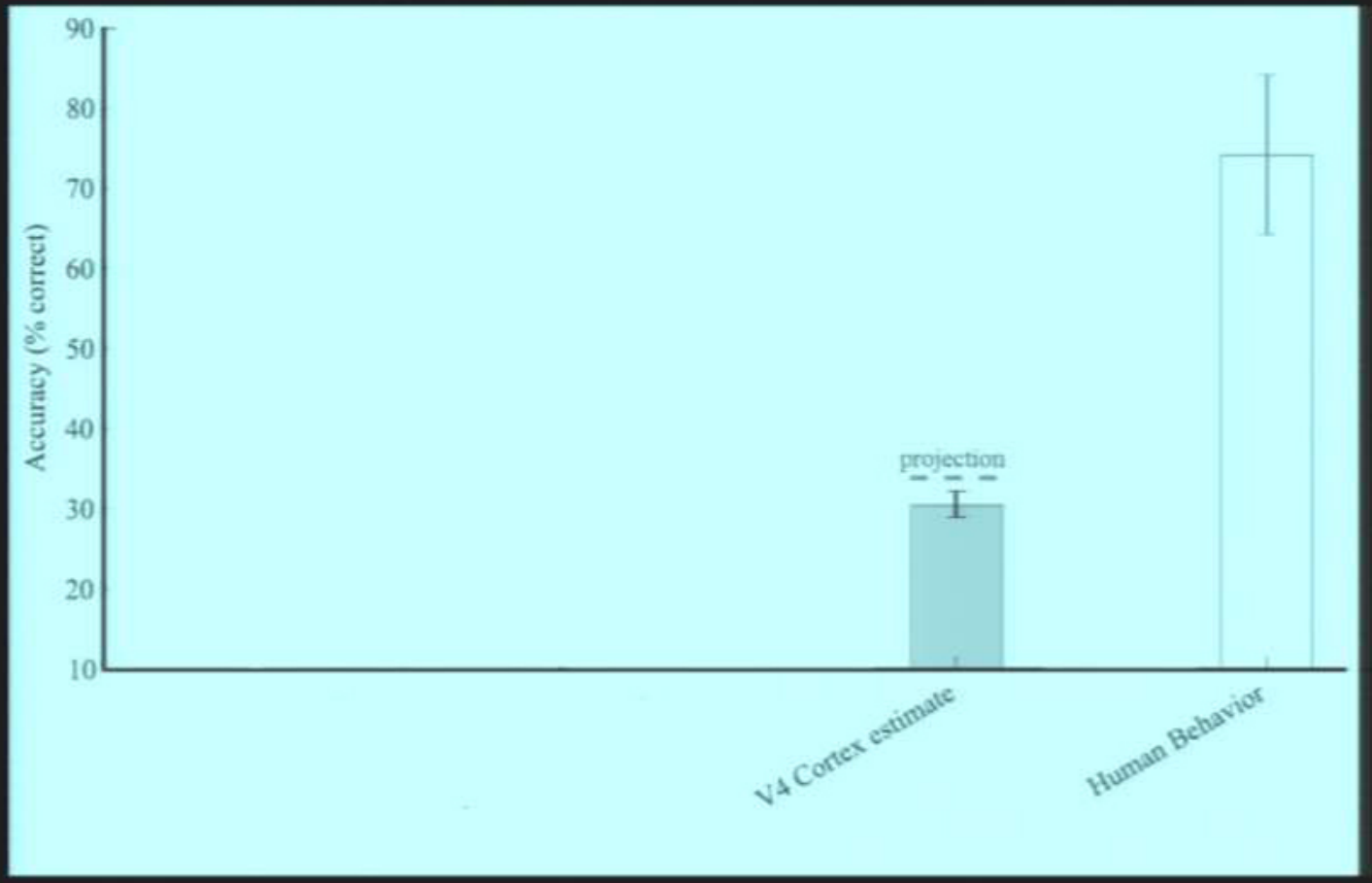
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



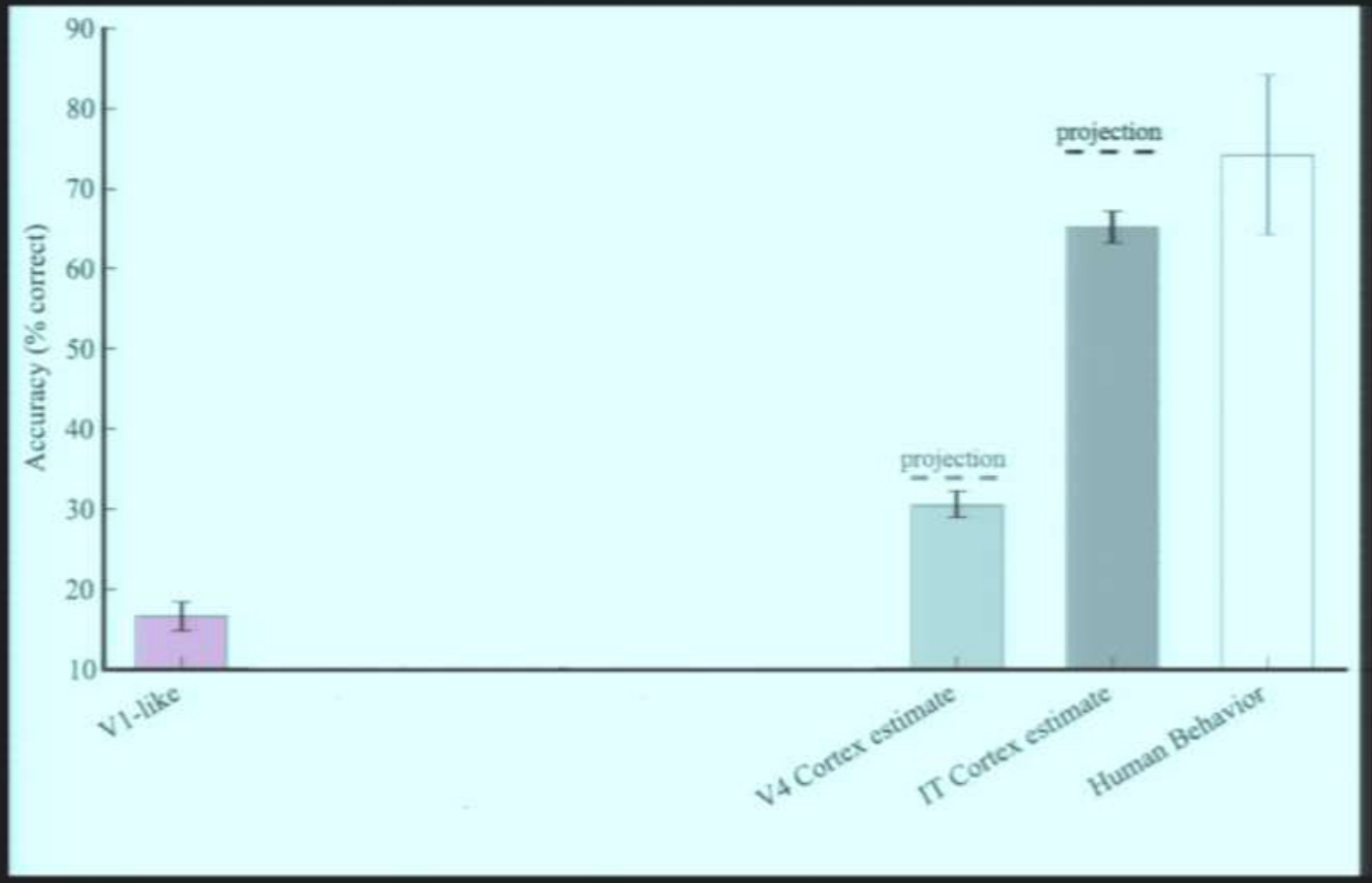
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



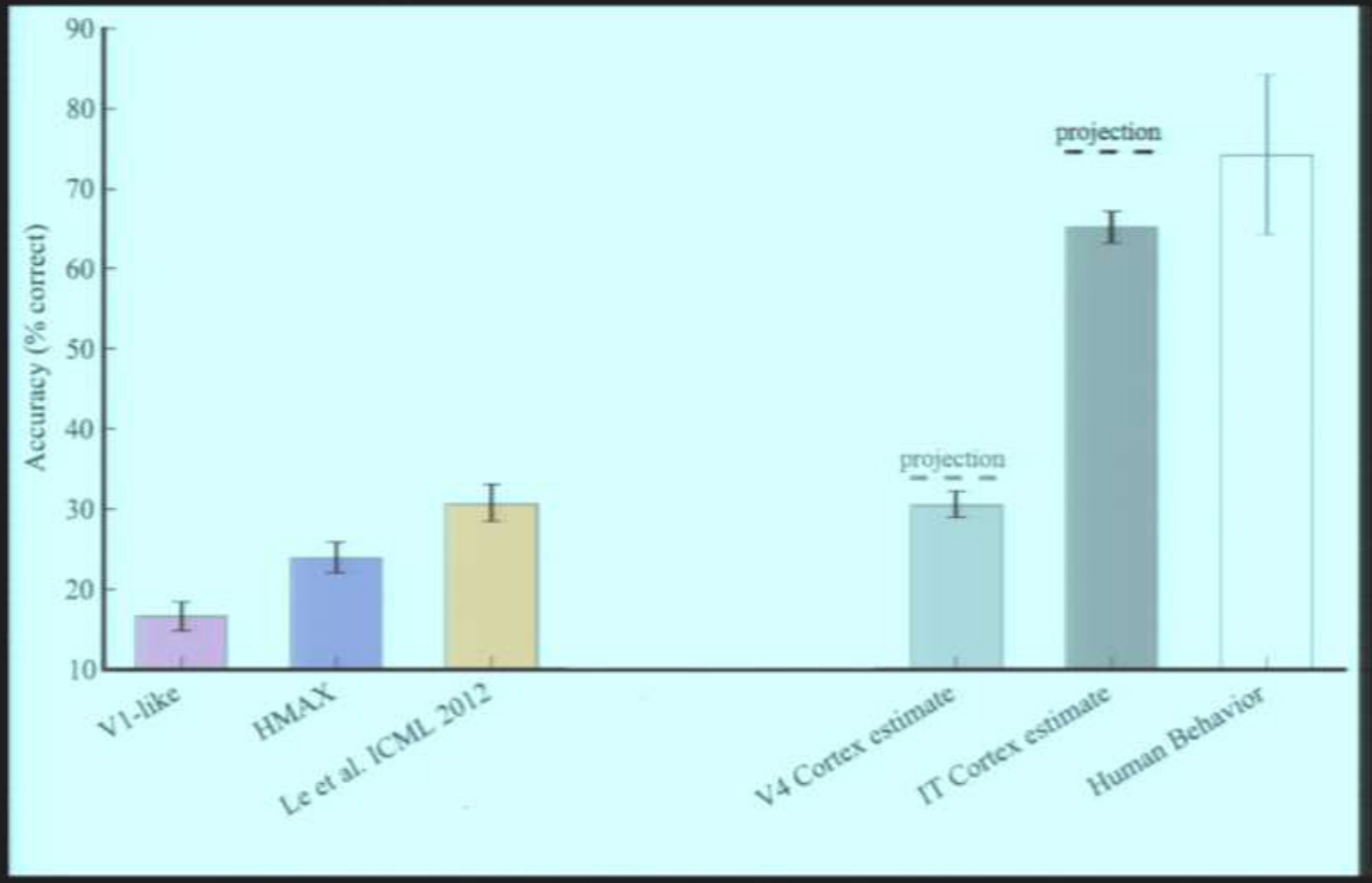
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



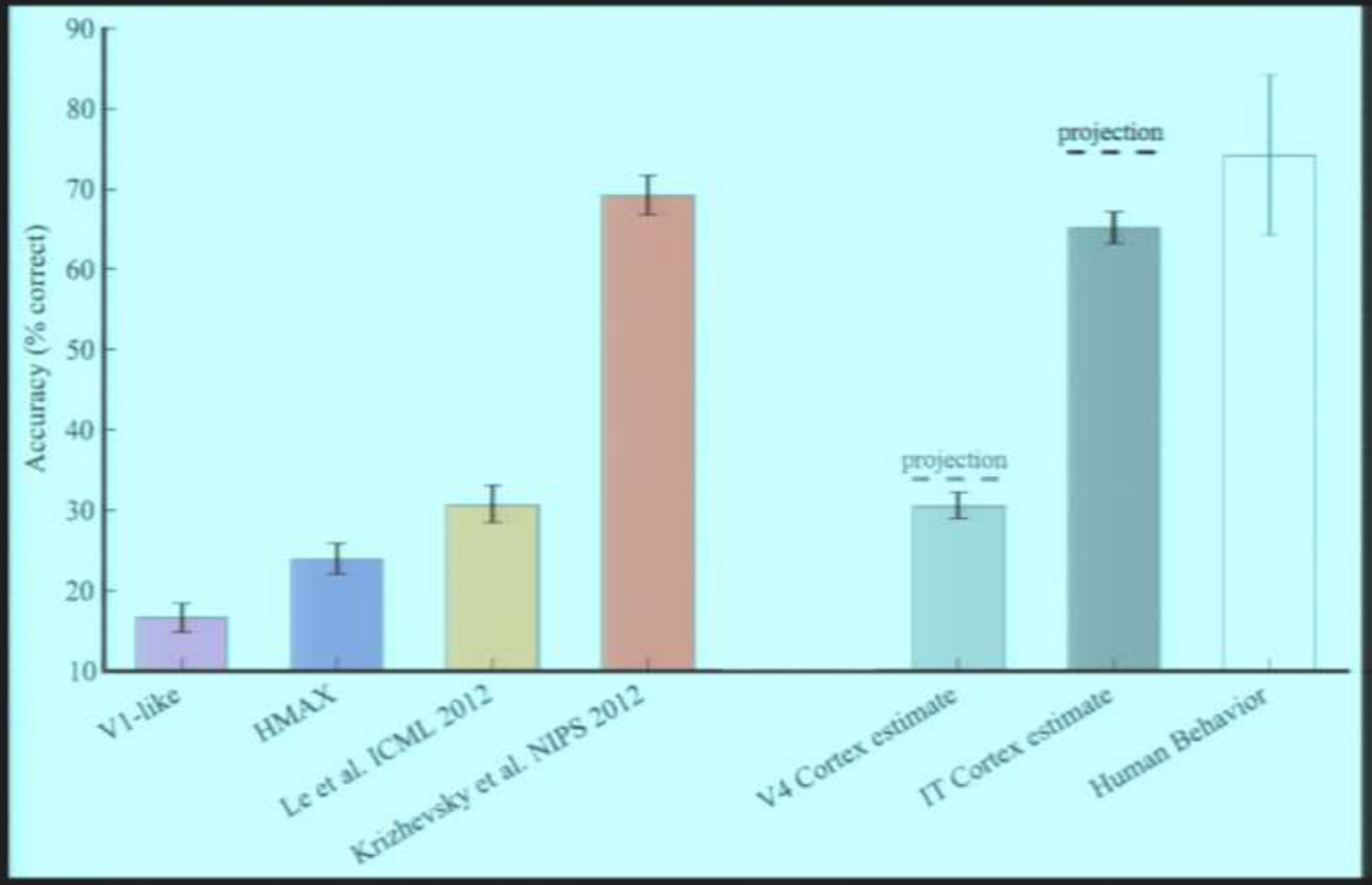
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



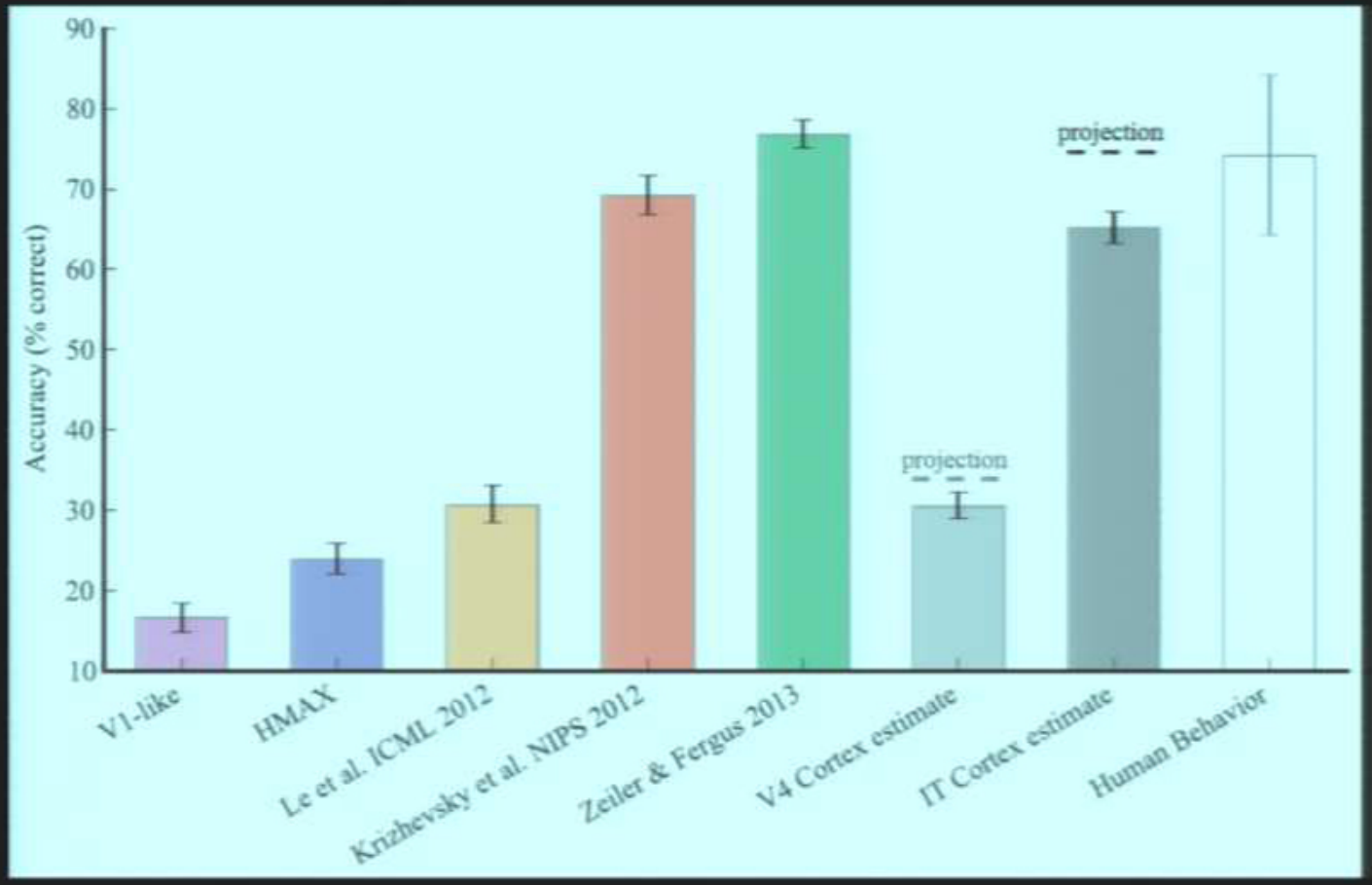
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



Deep Nets vs Monkey vs Humans [Cadieu et al.]

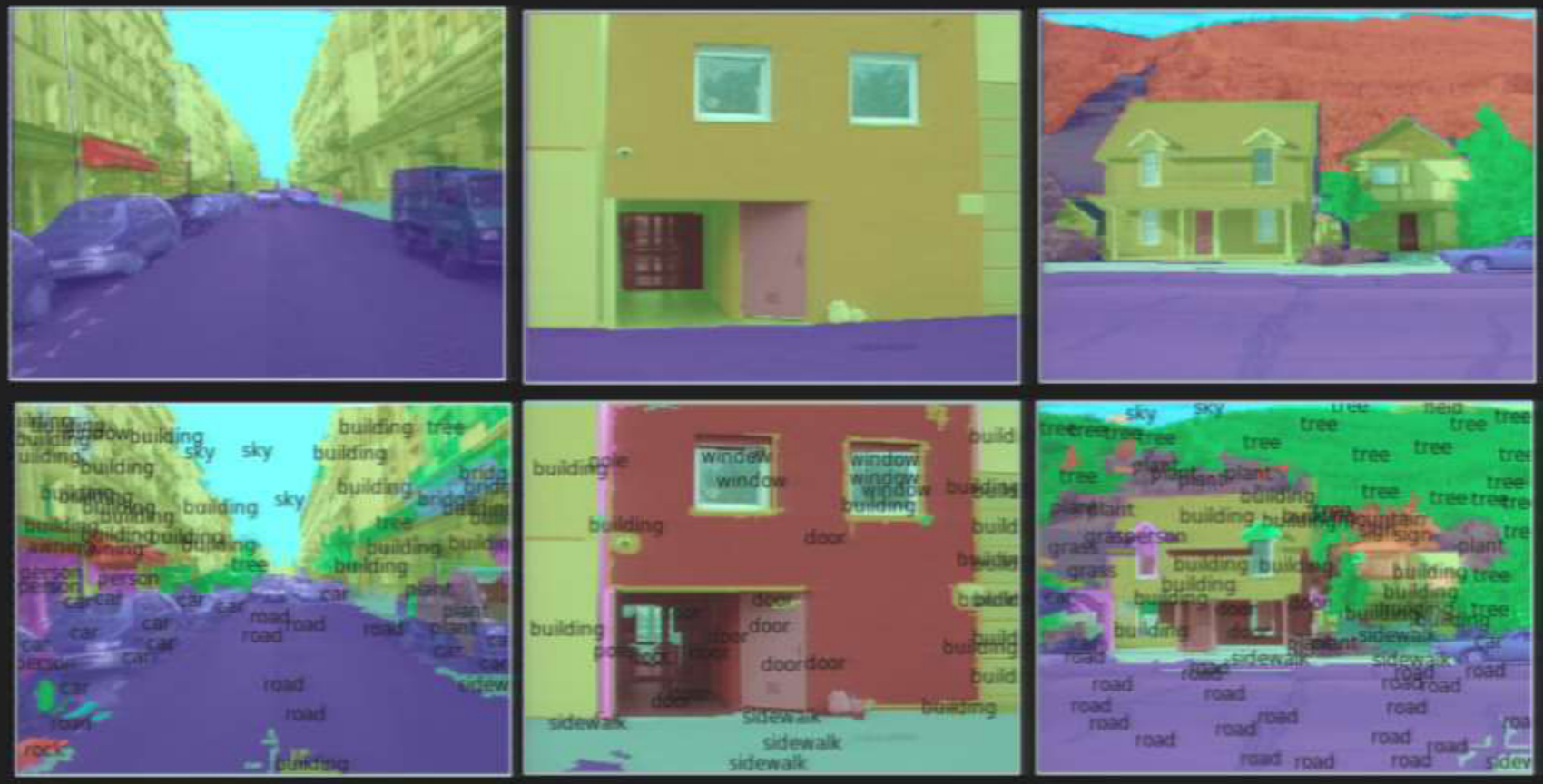
- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



Other Vision Applications

Scene Parsing

- Farabet et al. "Learning hierarchical features for scene labeling" PAMI 2013

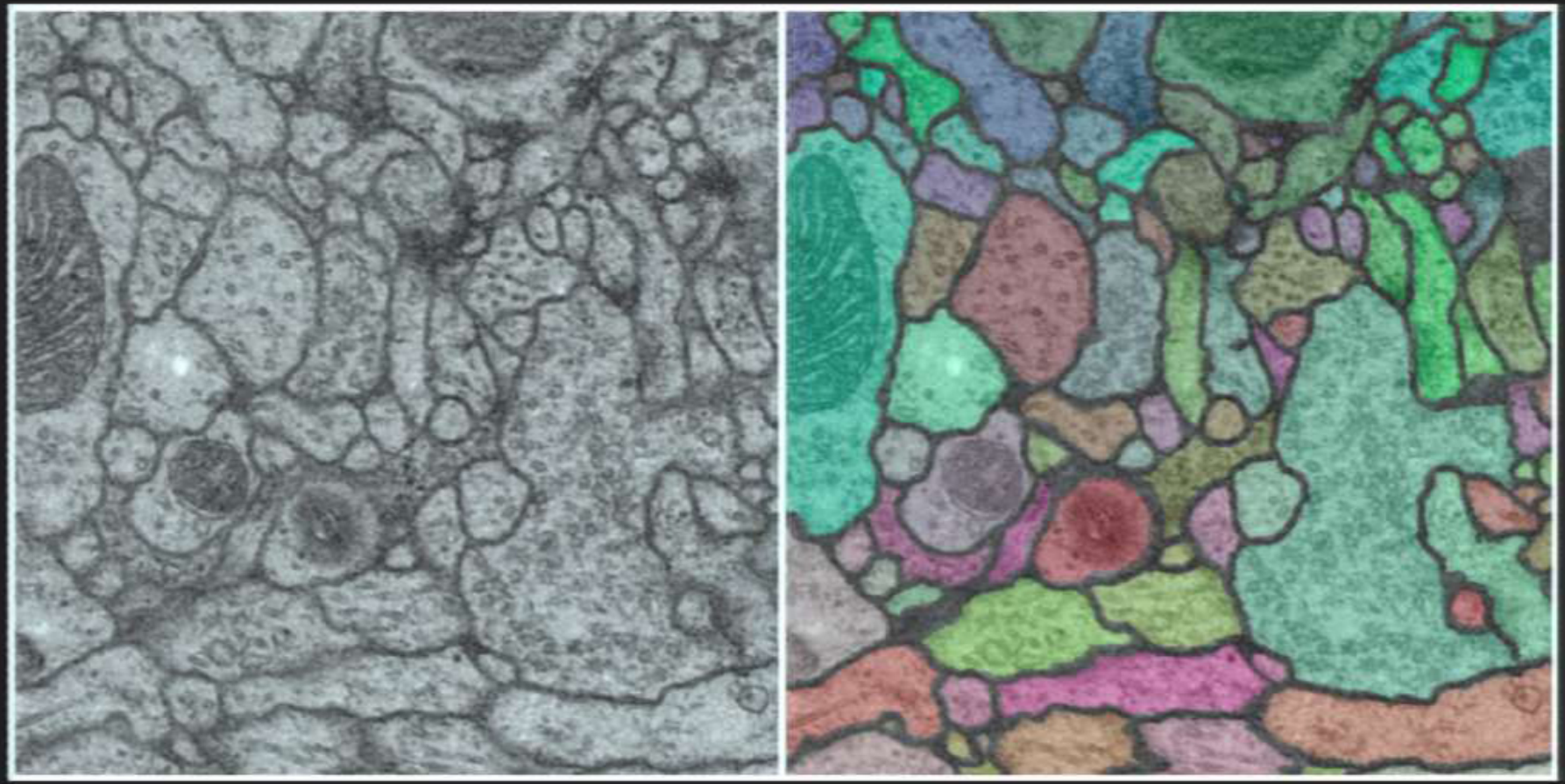


Action Recognition from Video

Taylor et al. "Convolutional learning of spatio-temporal features" ECCV 2010

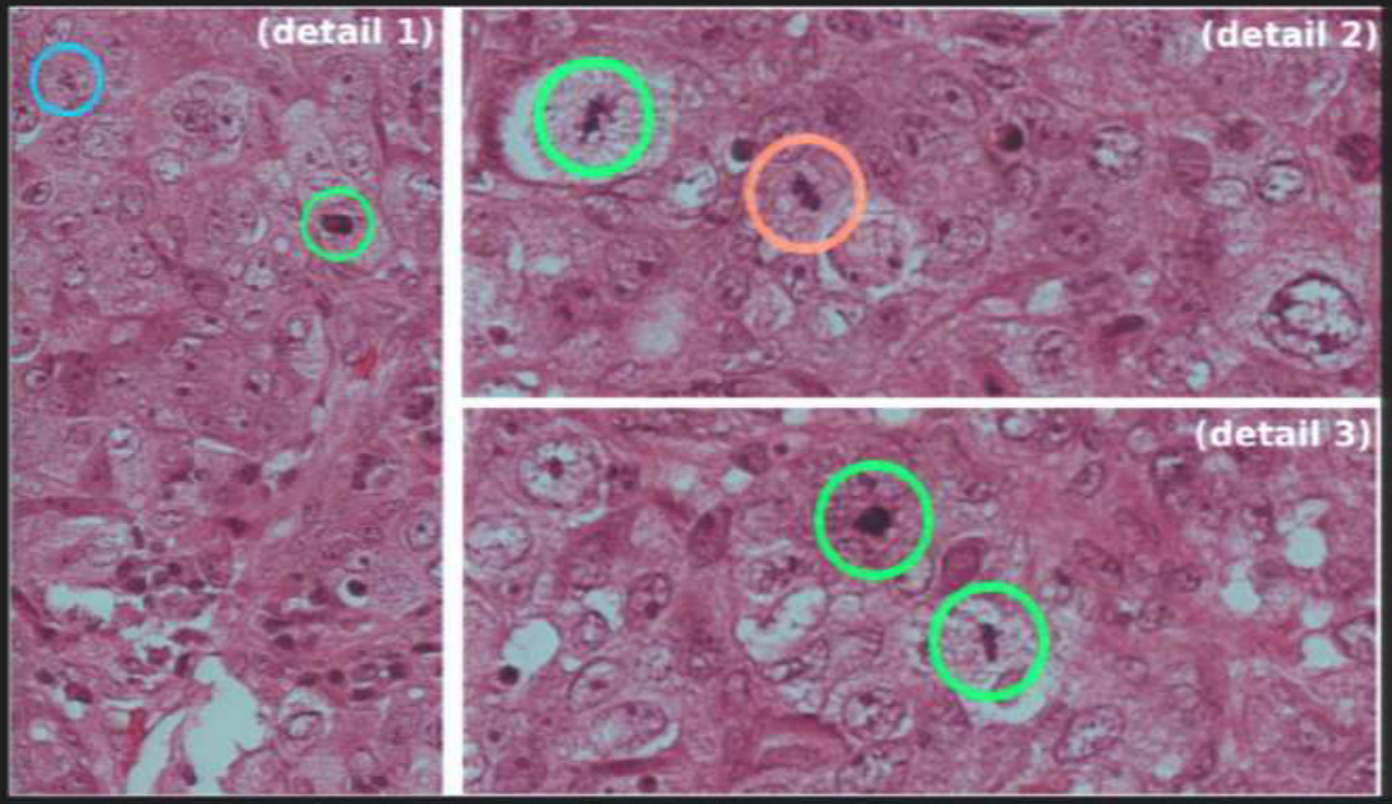
Segmentation

- Ciresan et al. "DNN segment neuronal membranes..." NIPS 2012
- Turaga et al. "Maximin learning of image segmentation" NIPS 2009



Biological Detection

- D. Ciresan, A. Giusti, L.M. Gambardella, J. Schmidhuber - Mitosis Detection in Breast Cancer Histology Images using Deep Neural Networks (MICCAI 2013)



Denoising

- Burger et al. "Can plain NNs compete with BM3D?" CVPR 2012

Original

Noised

Denoised

Removing Artifacts

[Eigen et al. "Restoring an Image Taken Through a Window Covered with Dirt or Rain" ICCV 2013]

Removing Artifacts

[Eigen et al. "Restoring an Image Taken Through a Window Covered with Dirt or Rain" ICCV 2013]

Removing Artifacts

[Eigen et al. "Restoring an Image Taken Through a Window Covered with Dirt or Rain" ICCV 2013]

Removing Artifacts

[Eigen et al. "Restoring an Image Taken Through a Window Covered with Dirt or Rain" ICCV 2013]

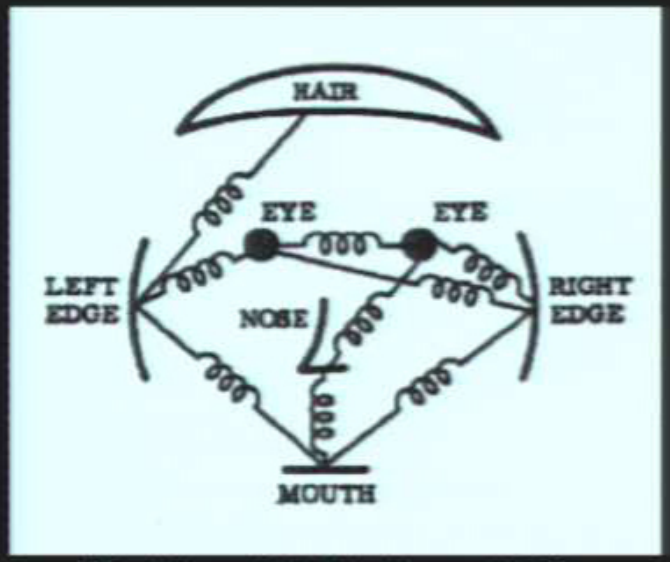
Future Directions

Video

- Relatively under-explored
- Learn about 3D structure from motion
- Need big labeled video dataset

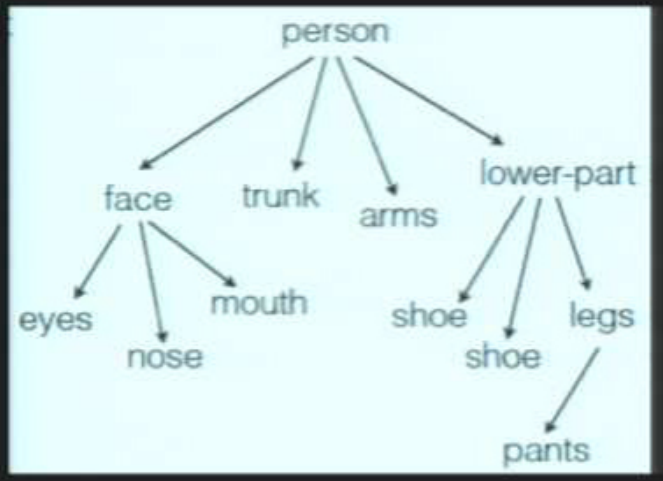
Deep Learning + Structured Prediction

- ConvNet feature extractor
- Combine with top-down reasoning



[Fischler and R. Elschlager 1973]

Stochastic Grammars



[R. Girshick, P. Felzenszwalb, D. McAllester, Object Detection with Grammar Models, NIPS 2011]

Summary

- Yann LeCun was right!
- Deep ConvNets work well for recognition
  - Quite a bit better than existing vision approaches
- Can be used for many other vision tasks
- But unsupervised learning still an open problem

DEMO

URL: http://horatio.cs.nyu.edu

Image Classifier Demo

NYU

Upload your images to have them classified by a machine! Upload multiple images using the button below or dropping them on this page. The predicted objects will be refreshed automatically or you can manually refresh yourself. Images are resized such that the smallest dimension becomes 256, then the center 256x256 crop is used. More about the demo can be found out here.

+ Upload Images | Remove All

Demo Notes

- The maximum file size for uploads in this demo is **10 MB**.
- Only image files (**JPEG, JPG, GIF, PNG**) are allowed in this demo.
- You can **drag & drop** files from your desktop on this webpage with Google Chrome, Mozilla Firefox and Apple Safari.
- All images from your current session and IP are shown above.
- This demo is powered by research out of New York University. Click here to find out more

Demo created by: Matthew Zeiler

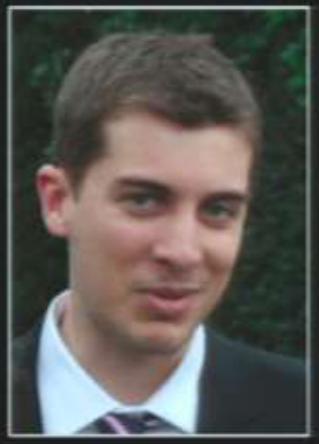
NEW YORK UNIVERSITY

Acknowledgements

- Colleagues in CILVR lab @ NYU:

Matt Zeiler

David Eigen



Pierre Sermanet

Yann LeCun

- Slides: Marc'Aurelio Ranzato
- Funding: NSF, DARPA, ONR
  Microsoft, Google, Facebook

ICLR 2014 Conference

- Deadline: 20th December 2013
- Banff, Canada, April 14-16th 2014
- Welcome anything to do with representation learning!

Summary

- Yann LeCun was right!
- Deep ConvNets work well for recognition
  - Quite a bit better than existing vision approaches
- Can be used for many other vision tasks
- But unsupervised learning still an open problem

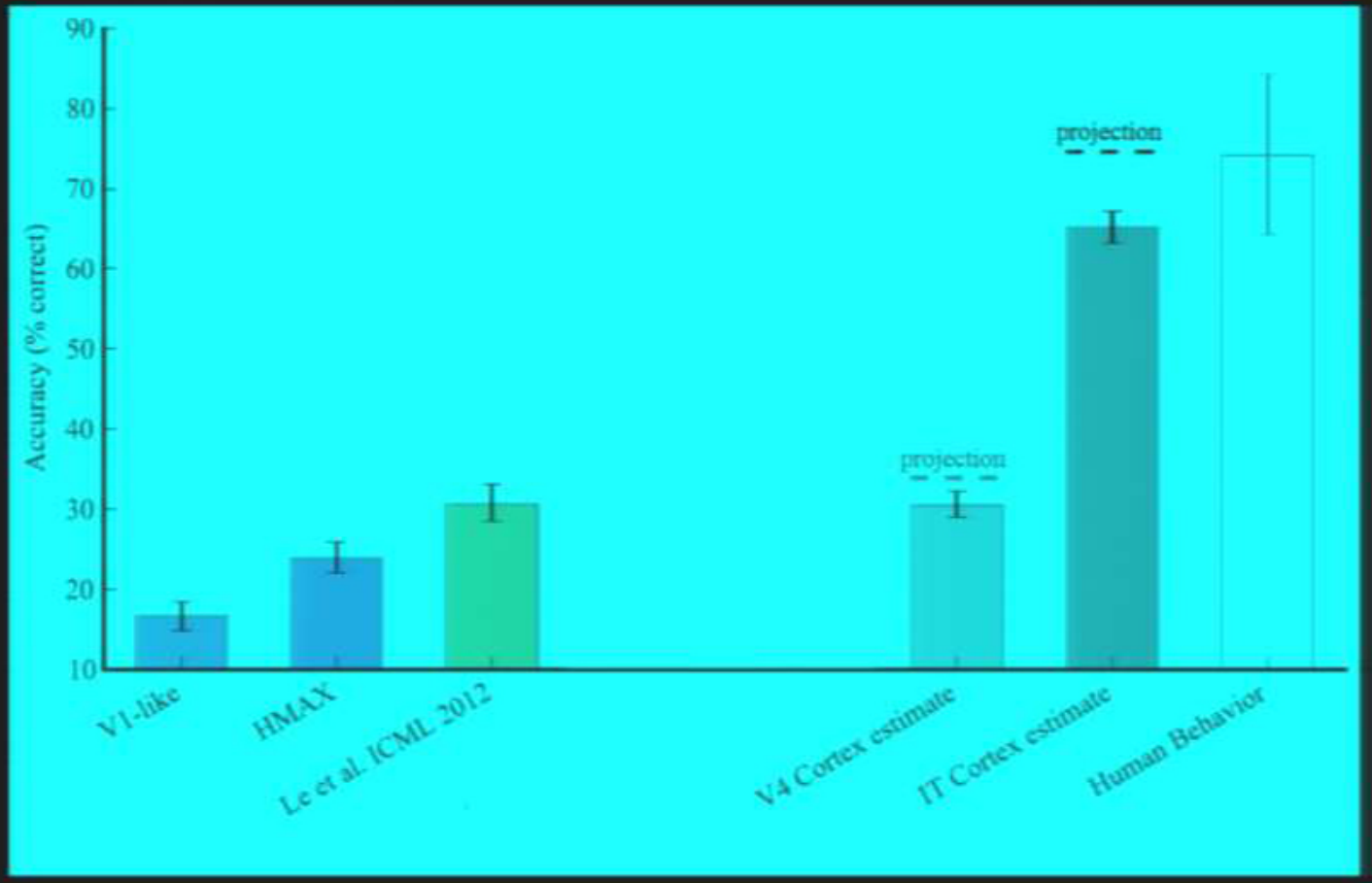
Removing Artifacts

[Eigen et al. "Restoring an Image Taken Through a Window Covered with Dirt or Rain" ICCV 2013]

Other Vision Applications

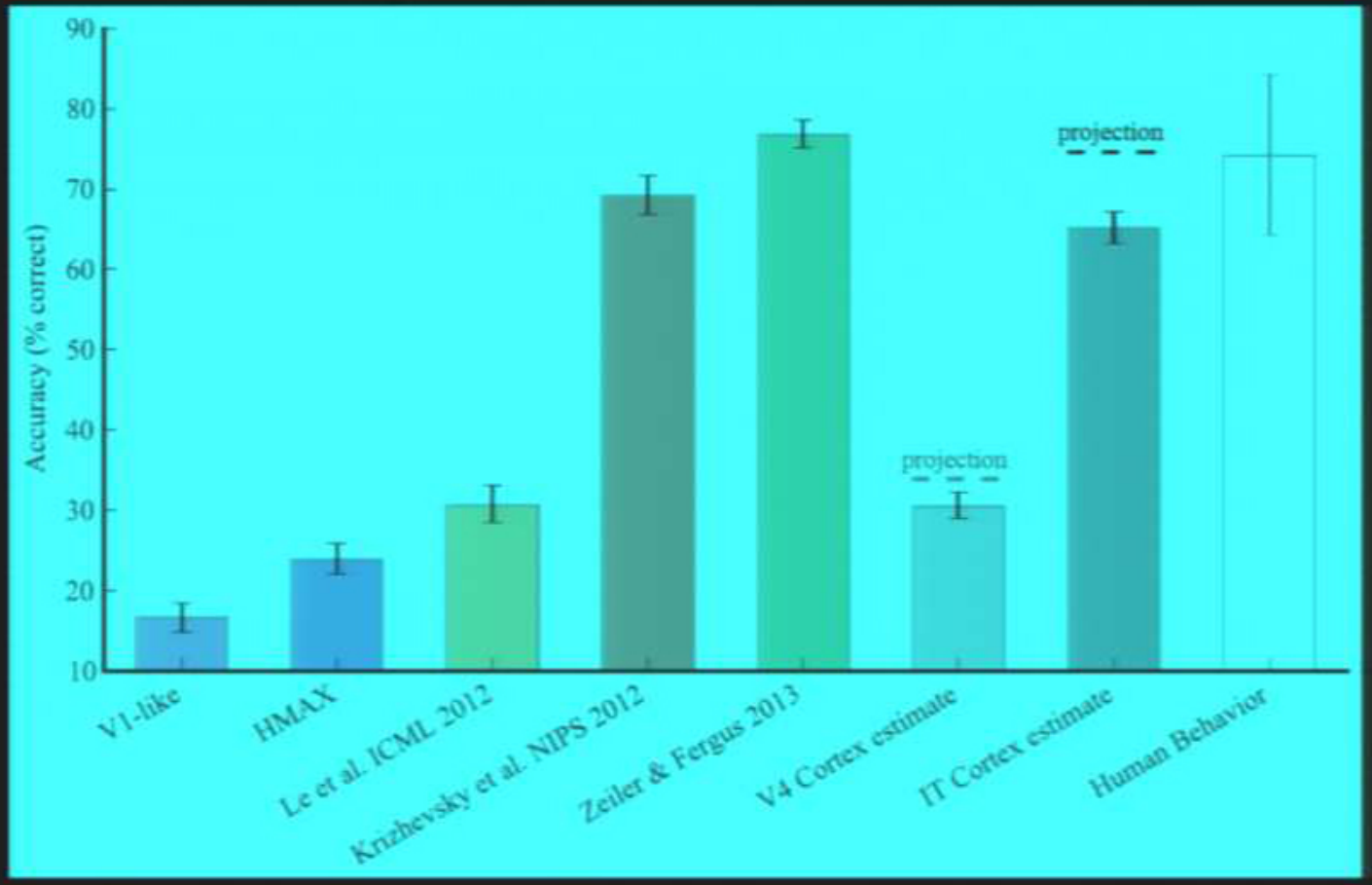
Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans



Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans

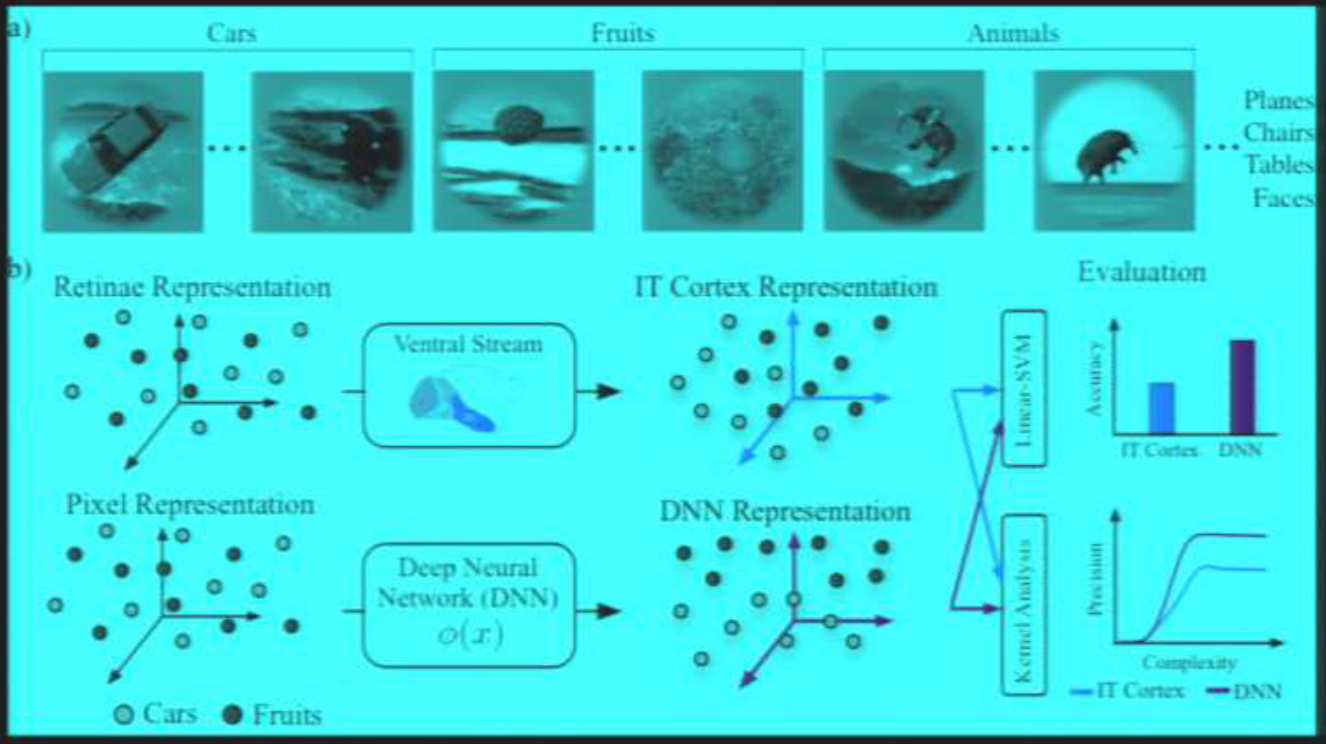


Deep Nets vs Monkey vs Humans [Cadieu et al.]

- Rapid presentation experiments (100ms)
- Feed-forward processing only in monkey/humans

Deep Nets vs Monkey vs Human

C.F. Cadieu, H. Hong, D. Yamins, N. Pinto, E.A. Solomon, N.J. Majaj, and J.J. DiCarlo. *Deep Neural Networks Rival the Object Recognition Performance of the Primate Visual System.* (PLOS One Biology, in submission, 2013).

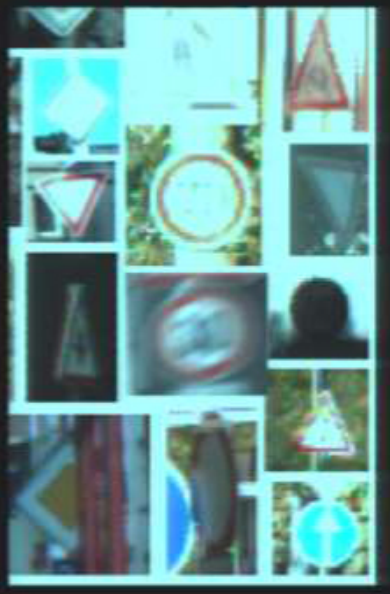


Deep Learning

Convnet Successes

- Handwritten text/digits
  - MNIST (0.17% error [Ciresan et al. 2011])
  - Arabic & Chinese [Ciresan et al. 2012]

- Simpler recognition benchmarks
  - CIFAR-10 (9.3% error [Wan et al. 2013])
  - Traffic sign recognition
    - 0.56% error vs 1.16% for humans [Ciresan et al. 2011]




Deep Learning for Computer Vision

NIPS 2013 Tutorial

Rob Fergus

Dept. of Computer Science
New York University

Unpooling Operation