05/05/15

Convolutional Neural Networks

Computer Vision

CS 543 / ECE 549

University of Illinois

Jia-Bin Huang

Reminder: final project

- Posters on **Friday, May 8** at **7pm** in **SC 2405**
  - two rounds, 1.25 hr each
- Papers due **May 11** by email
- Cannot accept late papers/posters due to grading deadlines
- Send Derek an email if you can’t present your poster

Today’s class

- Overview
- Convolutional Neural Network (CNN)
- Understanding and Visualizing CNN
- Training CNN
- Probabilistic Interpretation

Image Categorization: Training phase

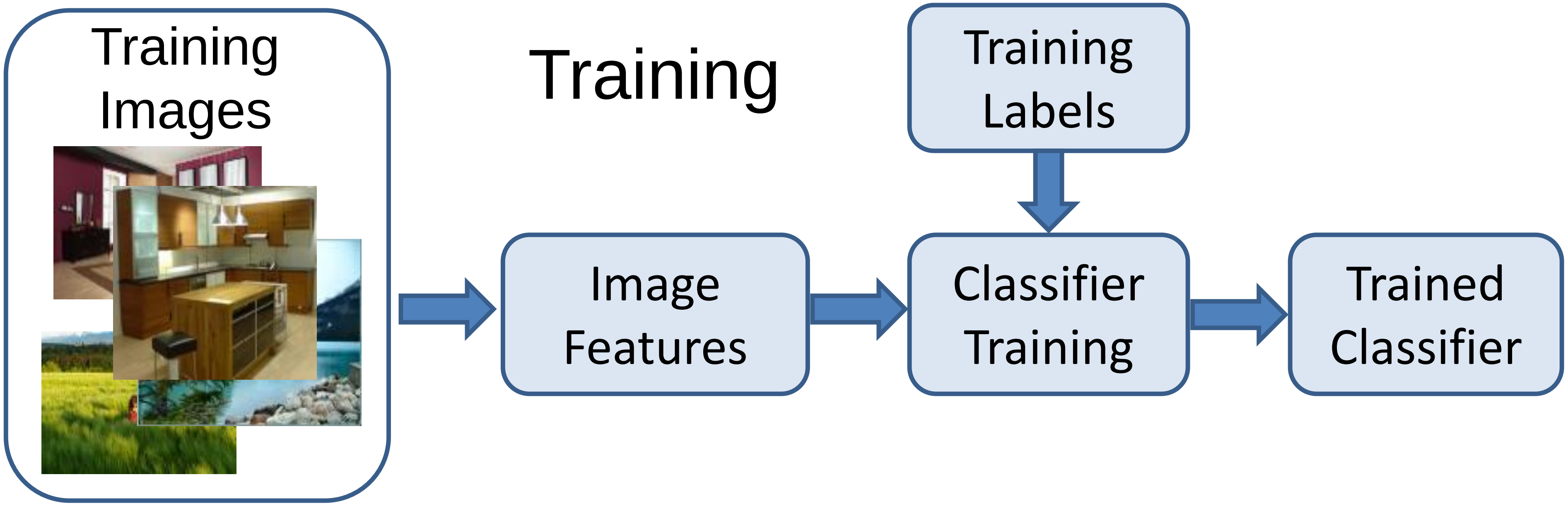
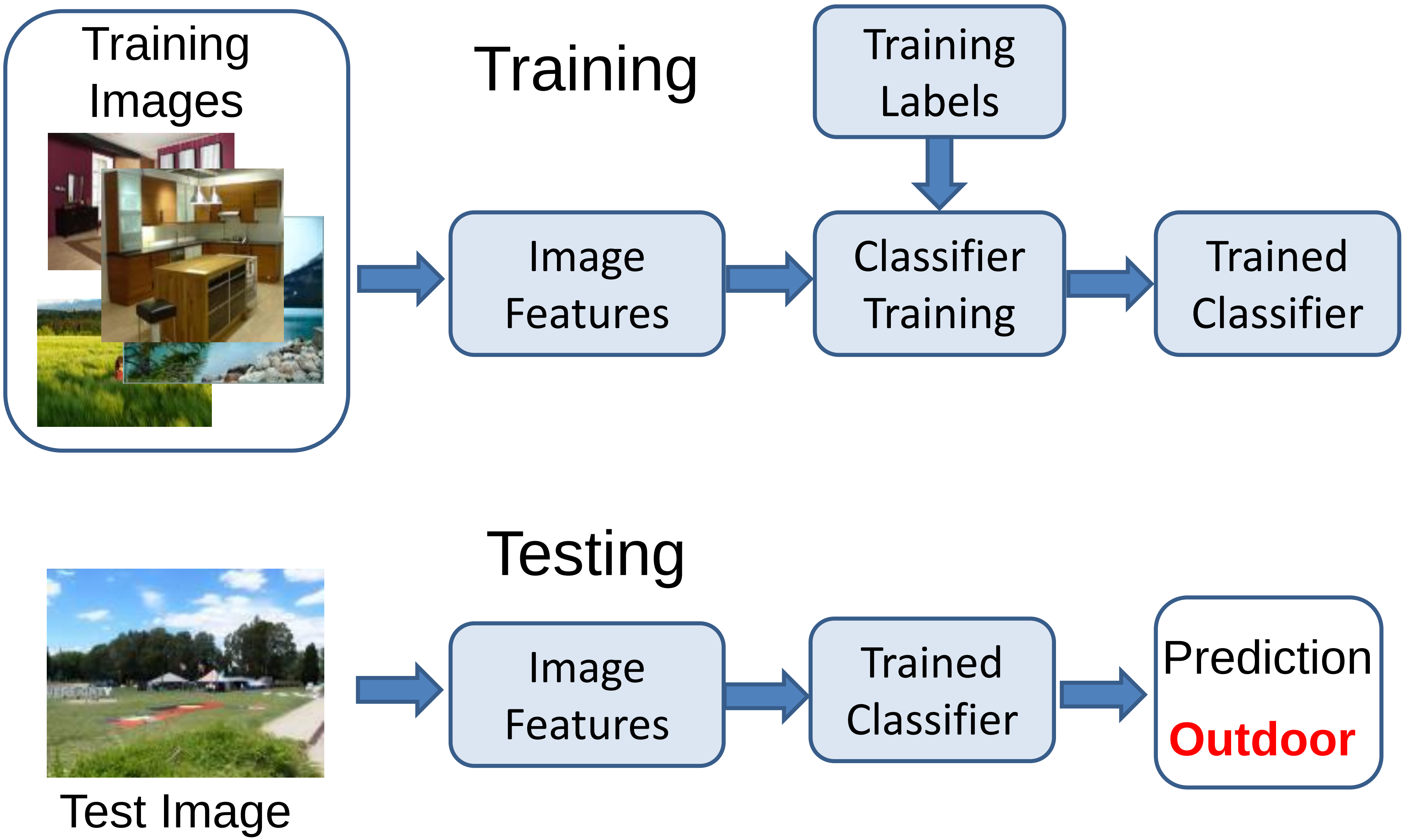
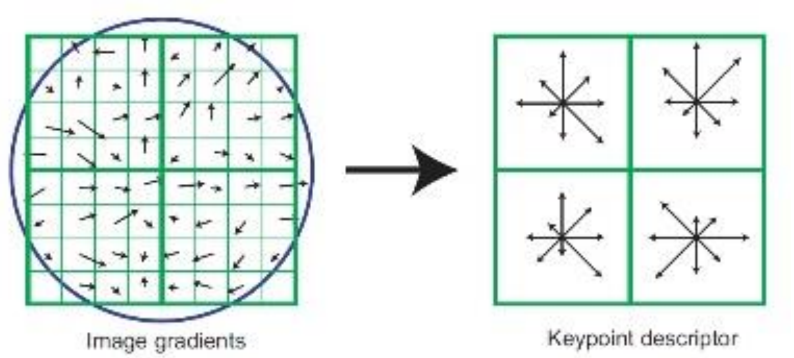


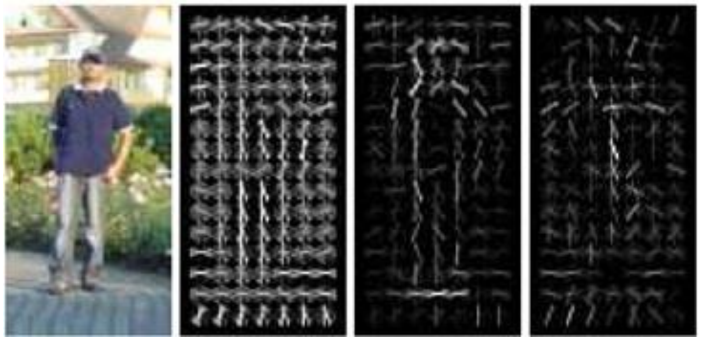
Image Categorization: Testing phase



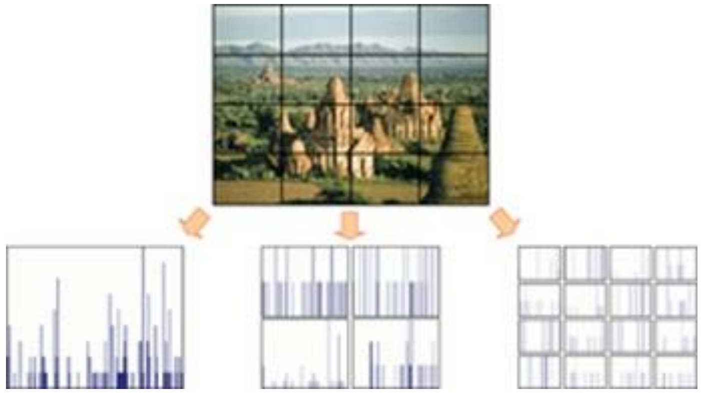
Features are the Keys



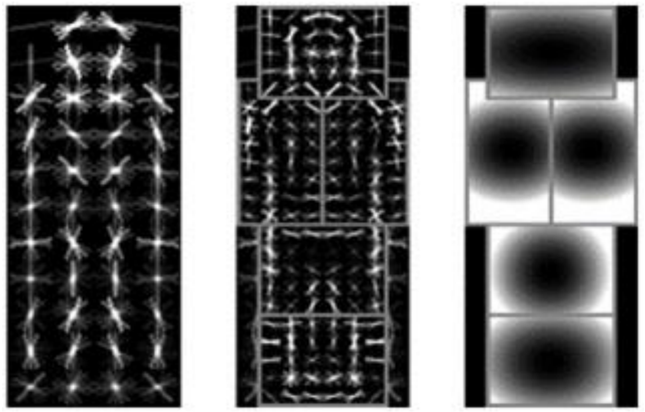
SIFT [Loewe IJCV 04]



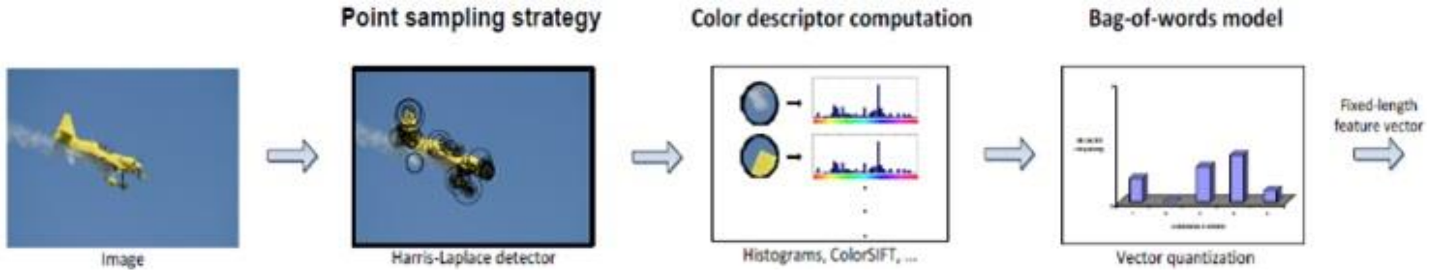
HOG [Dalal and Triggs CVPR 05]



SPM [Lazebnik et al. CVPR 06]



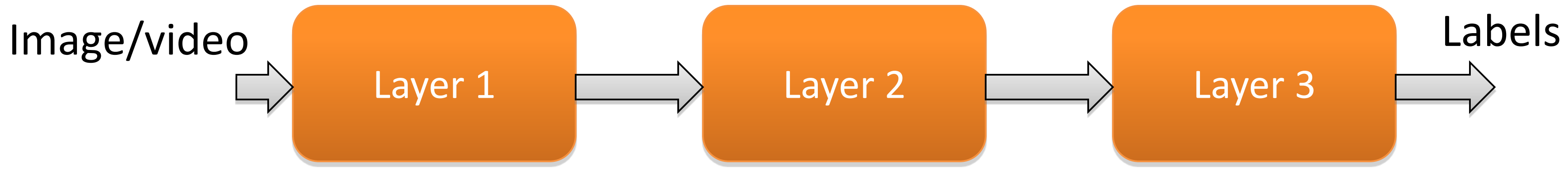
DPM [Felzenszwalb et al. PAMI 10]



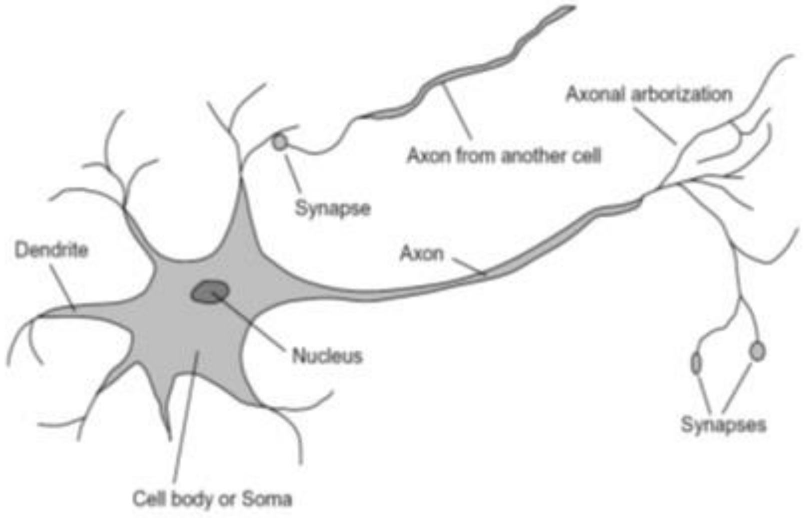
Color Descriptor [Van De Sande et al. PAMI 10]

Learning a Hierarchy of Feature Extractors

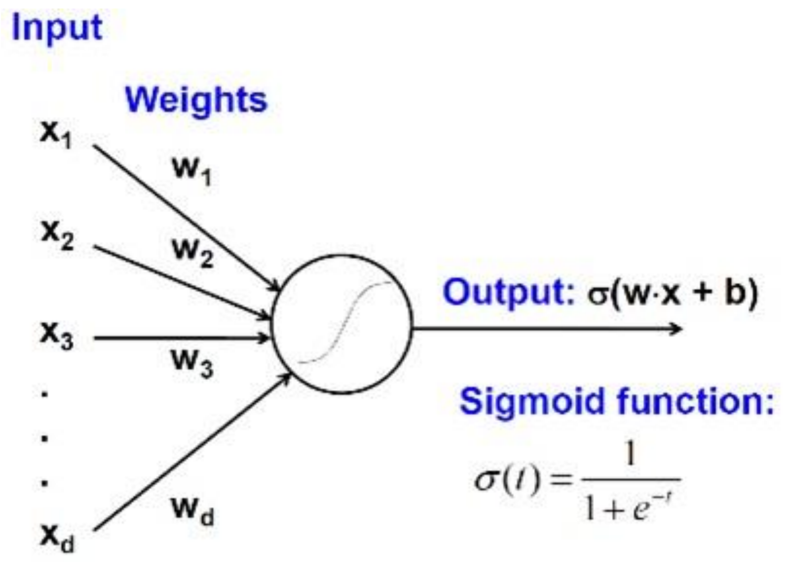
- Each layer of hierarchy extracts features from output of previous layer
- All the way from pixels → classifier
- Layers have the (nearly) same structure



Biological neuron and Perceptrons



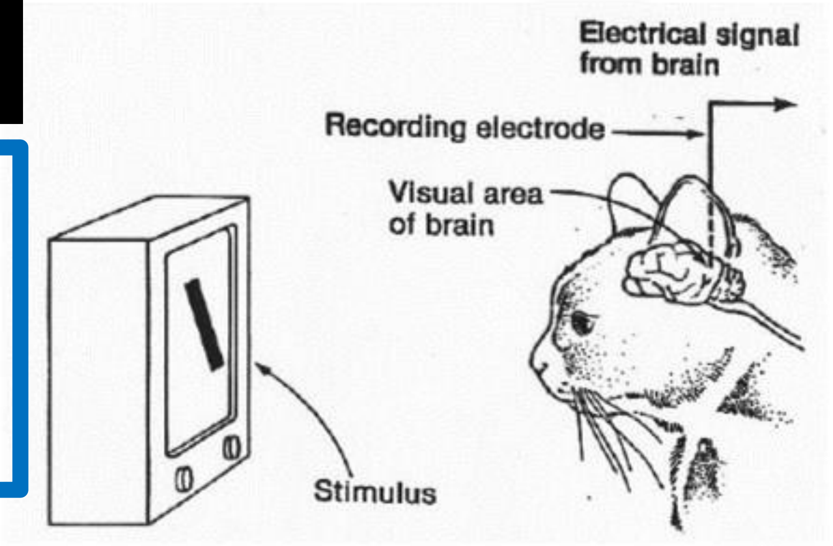
A biological neuron



An artificial neuron (Perceptron)
- a linear classifier

Simple, Complex and Hypercomplex cells

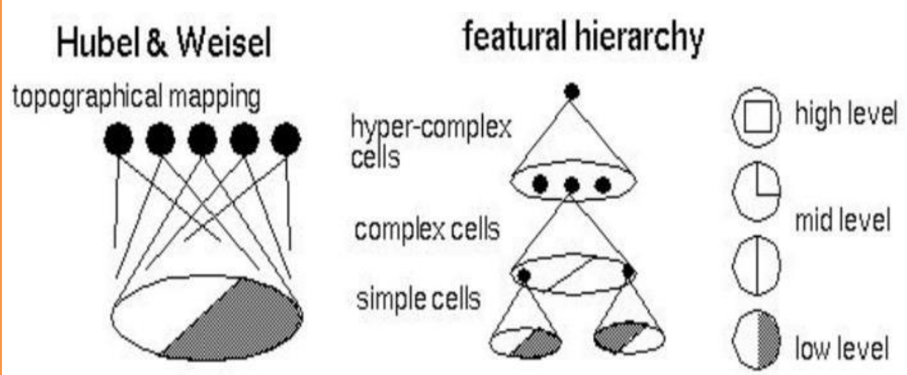
David H. Hubel and Torsten Wiesel



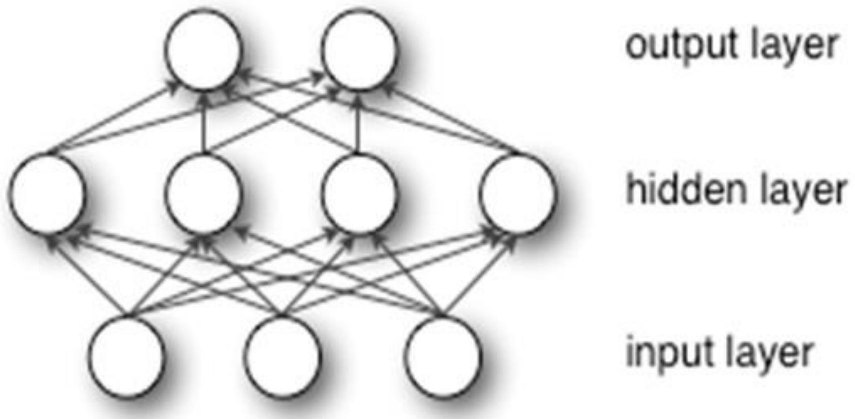
Suggested a **hierarchy** of **feature detectors** in the visual cortex, with higher level features responding to patterns of activation in lower level cells, and propagating activation upwards to still higher level cells.

David Hubel's Eye, Brain, and Vision

Hubel/Wiesel Architecture and Multi-layer Neural Network



Hubel and Weisel's architecture



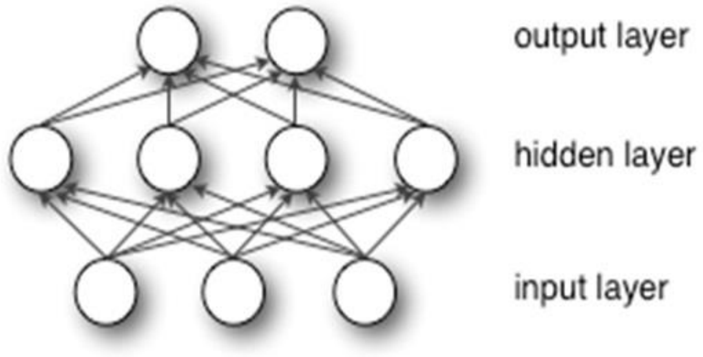
Multi-layer Neural Network
- A *non-linear* classifier

Multi-layer Neural Network

- A non-linear classifier
- **Training:** find network weights **w** to minimize the error between true training labels y_i and estimated labels $f_{\boldsymbol{w}}(\boldsymbol{x_i})$

$$E(\mathbf{w}) = \sum_{i=1}^{N} \left(y_i - f_{\mathbf{w}}(\mathbf{x}_i)\right)^2$$

- Minimization can be done by gradient descent provided f is differentiable
- This training method is called **back-propagation**



Convolutional Neural Networks (CNN, ConvNet, DCN)

- CNN = a multi-layer neural network with
  - **Local** connectivity
  - **Share** weight parameters across spatial positions

- One activation map (a depth slice), computed with one set of weights

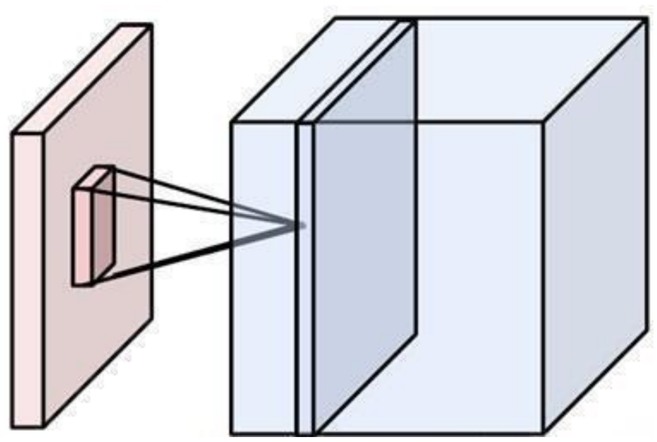


Image credit: A. Karpathy

Neocognitron [Fukushima, Biological Cybernetics 1980]

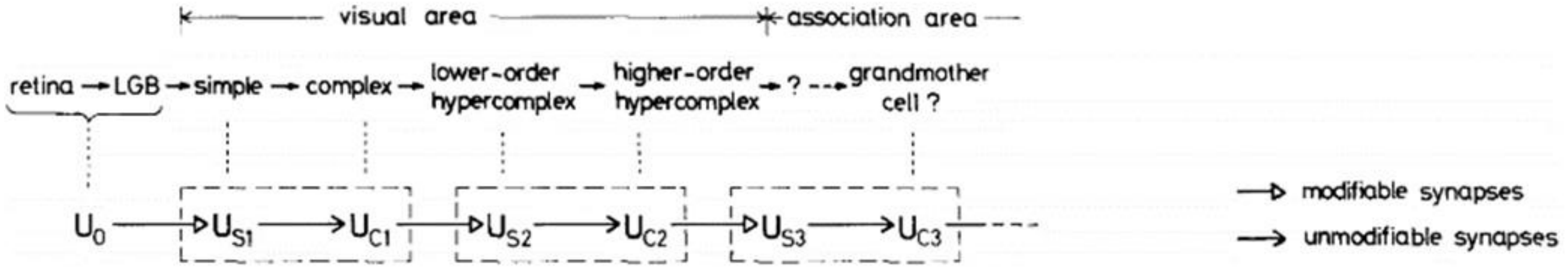
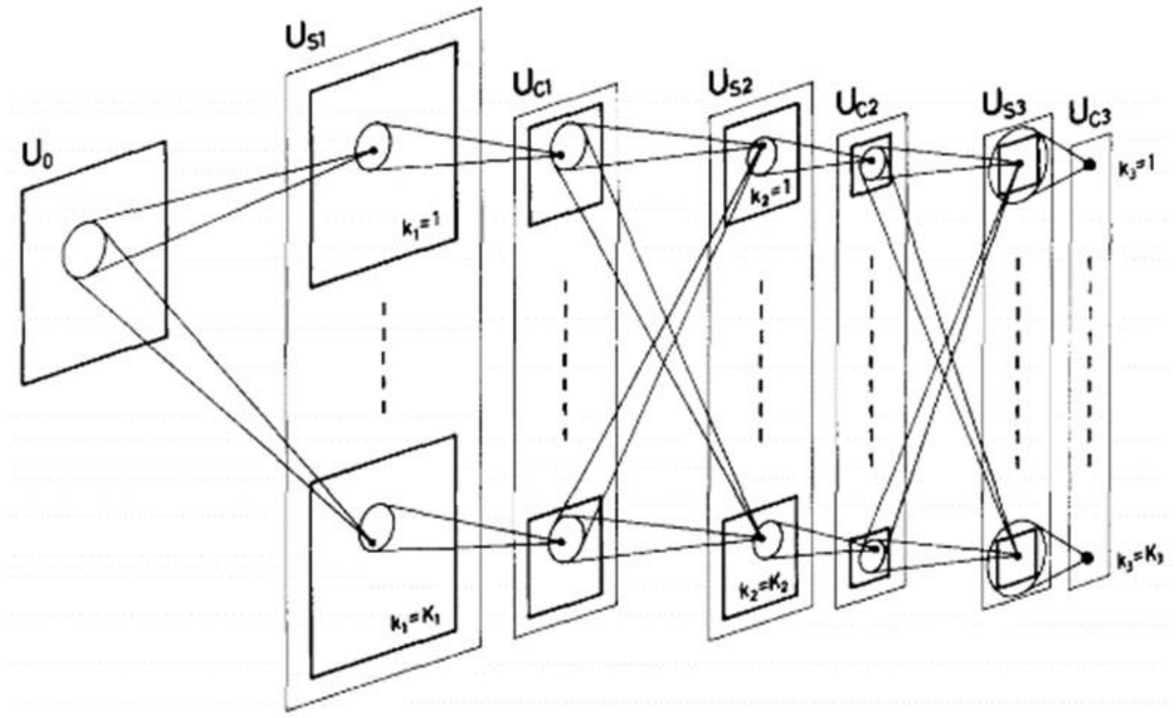


Fig. 1. Correspondence between the hierarchy model by Hubel and Wiesel, and the neural network of the neocognitron

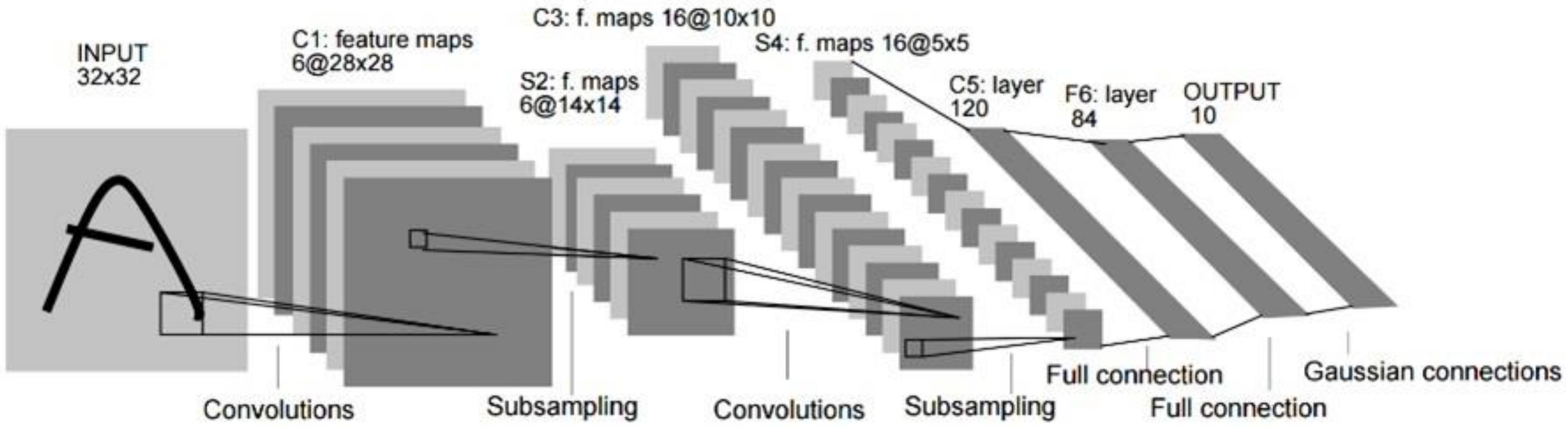


Deformation-Resistant Recognition

S-cells: (simple)
- extract local features

C-cells: (complex)
- allow for positional errors

LeNet [LeCun et al. 1998]

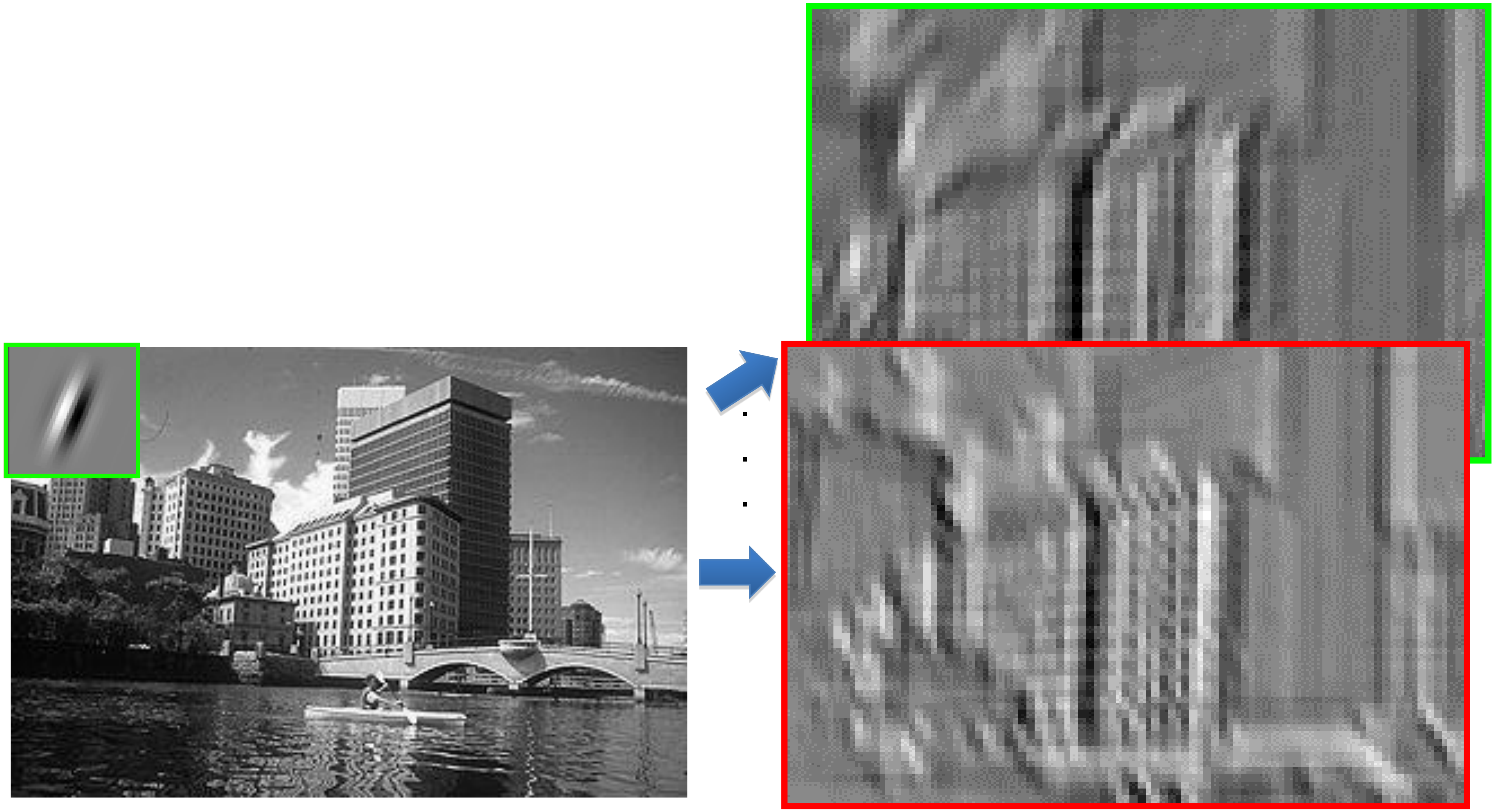


Gradient-based learning applied to document recognition [LeCun, Bottou, Bengio, Haffner 1998]

LeNet-1 from 1993

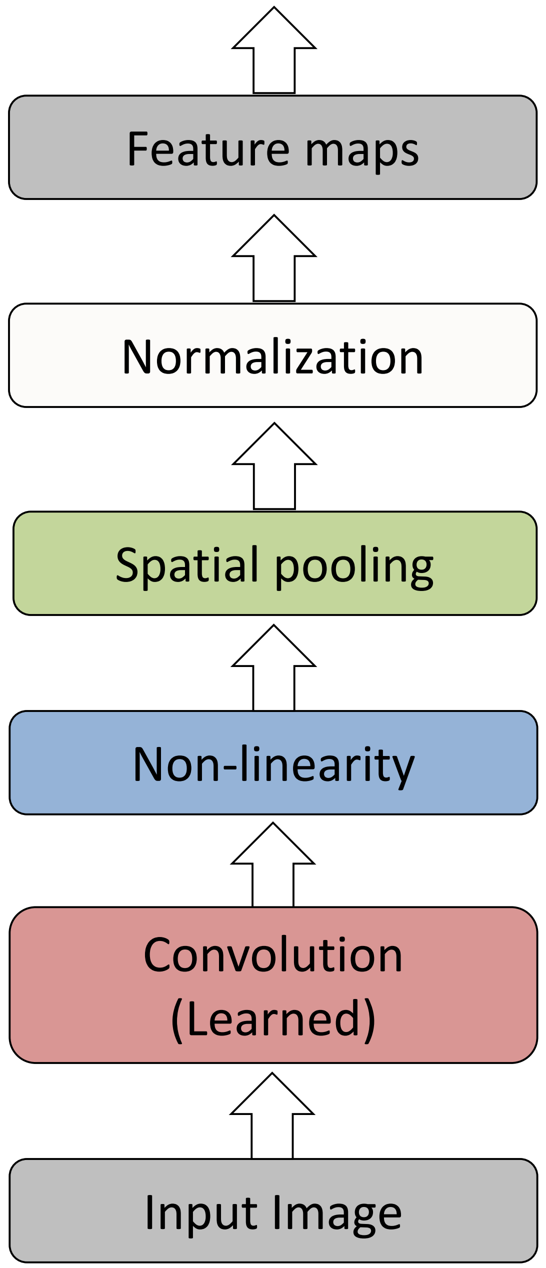
What is a Convolution?

- Weighted moving sum



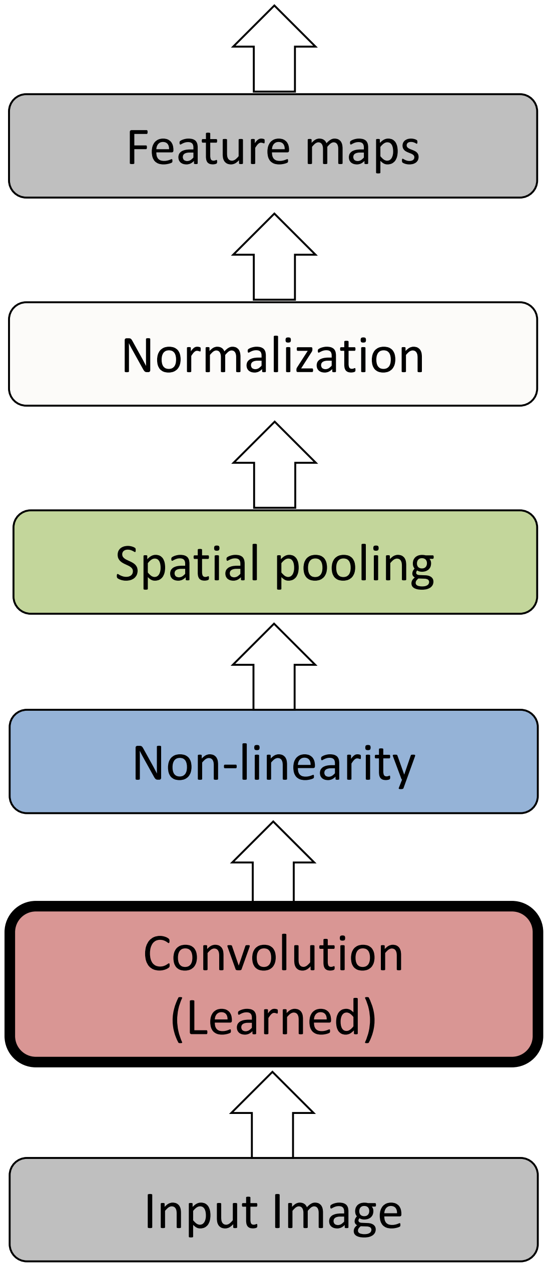
slide credit: S. Lazebnik

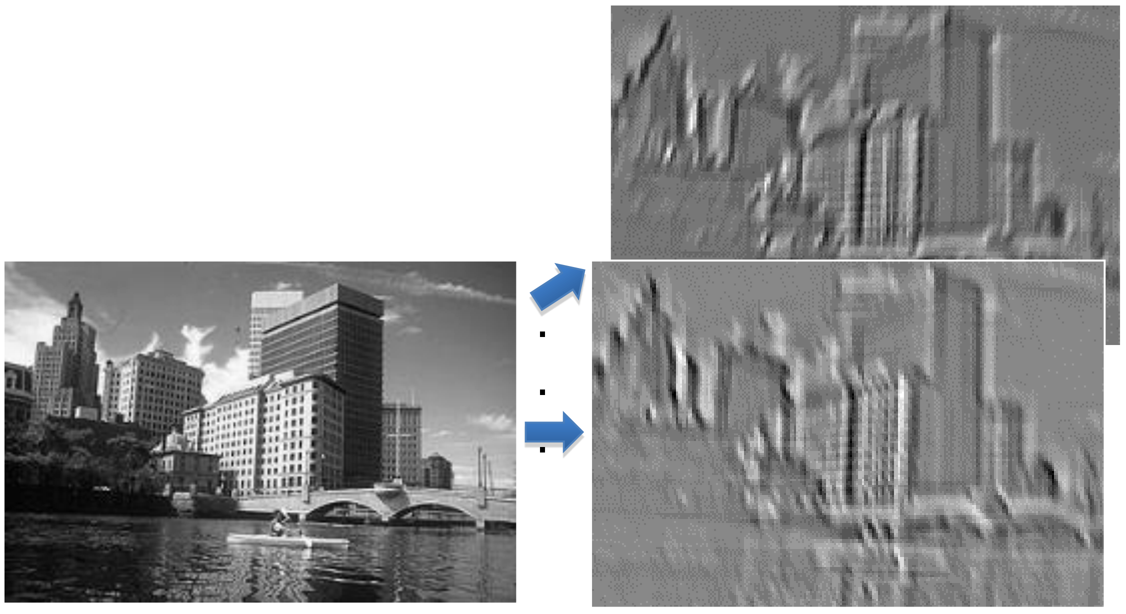
Convolutional Neural Networks



slide credit: S. Lazebnik

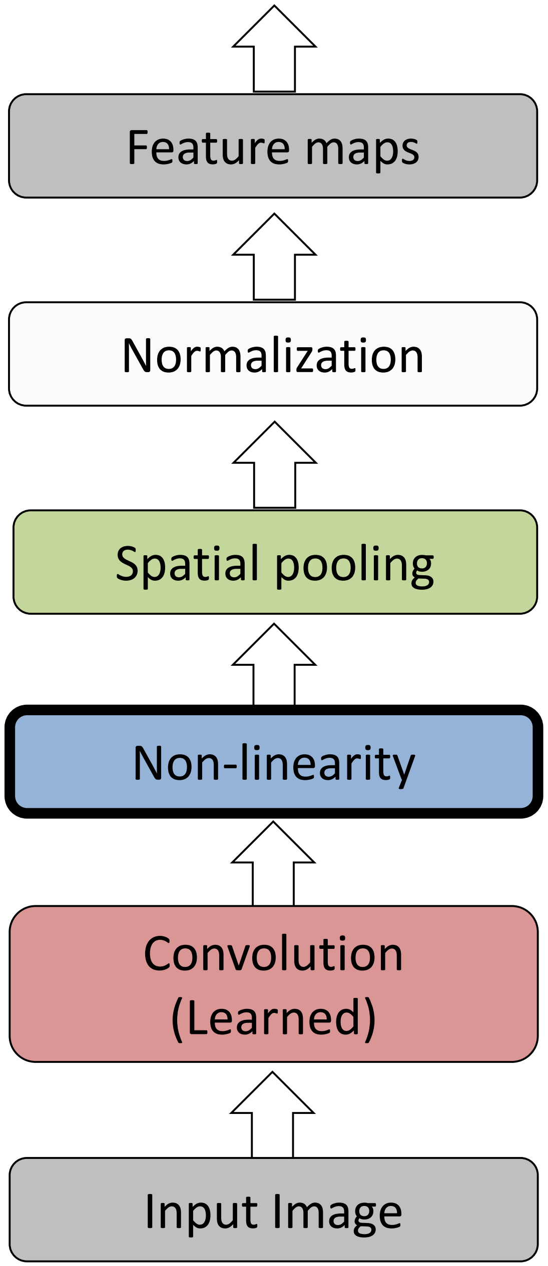
Convolutional Neural Networks



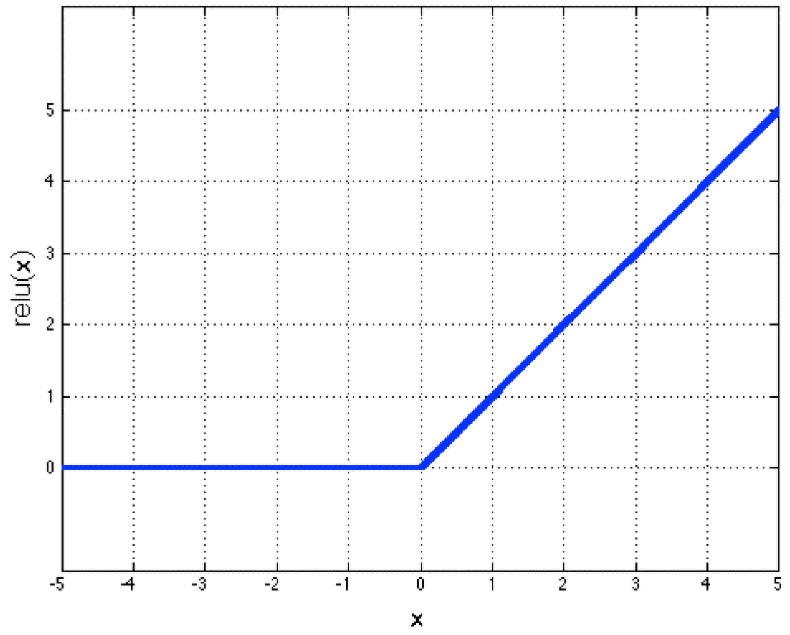


slide credit: S. Lazebnik

Convolutional Neural Networks

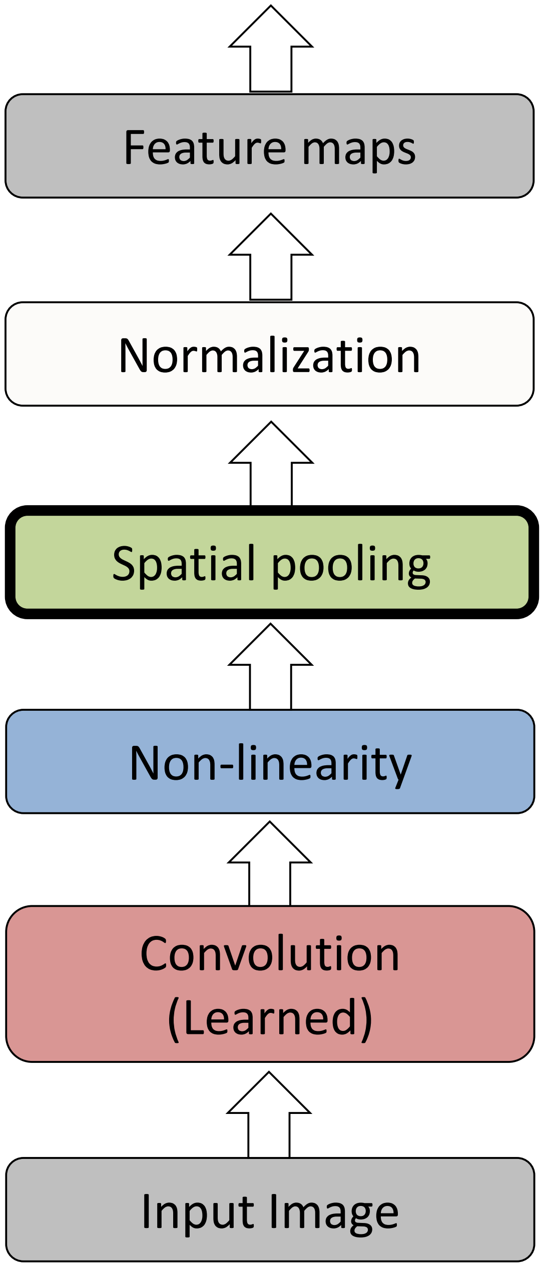


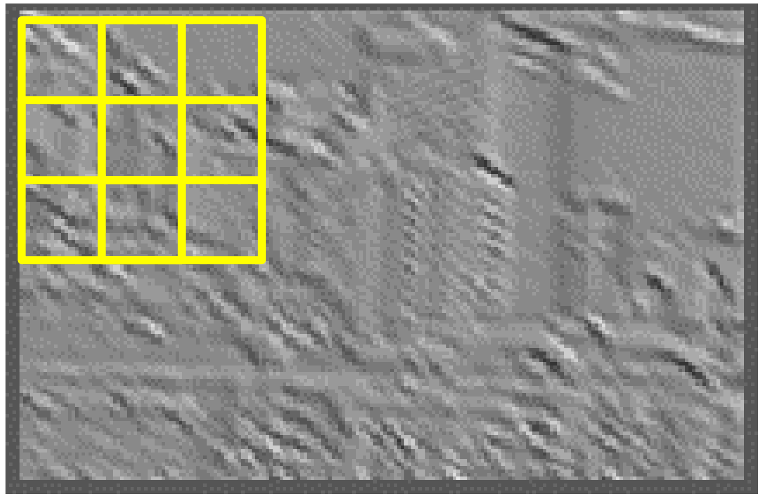
Rectified Linear Unit (ReLU)



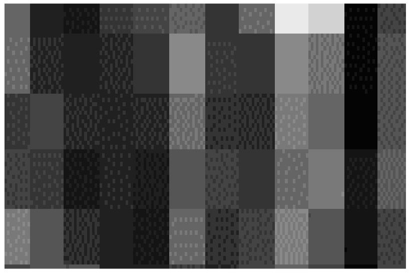
slide credit: S. Lazebnik

Convolutional Neural Networks





Max pooling

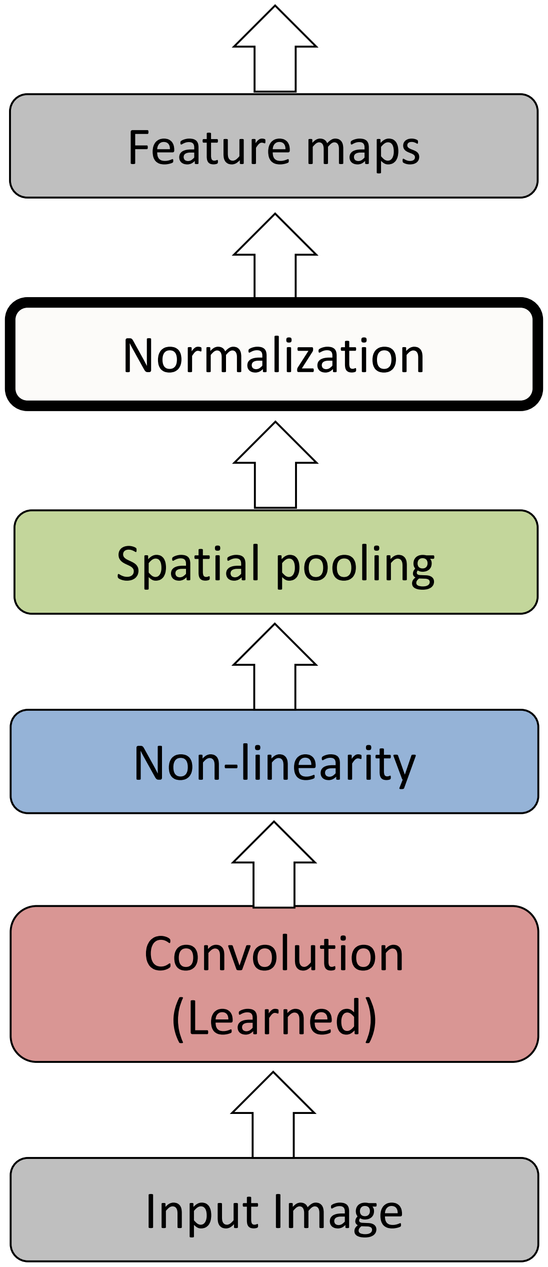


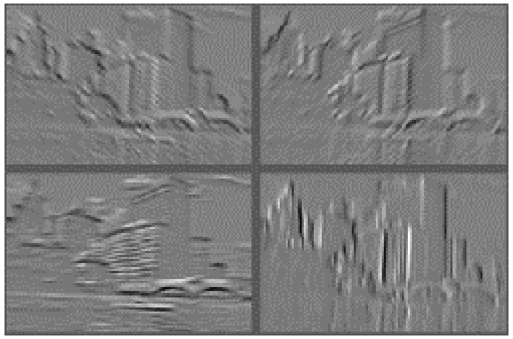
Max-pooling: a non-linear down-sampling

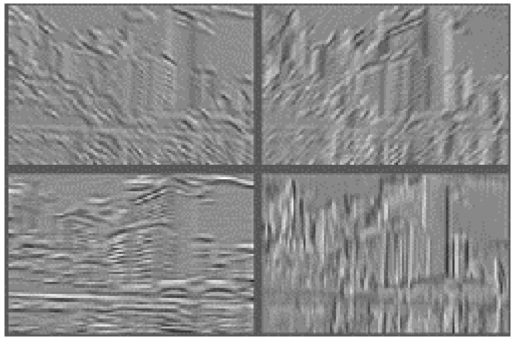
Provide *translation invariance*

slide credit: S. Lazebnik

Convolutional Neural Networks

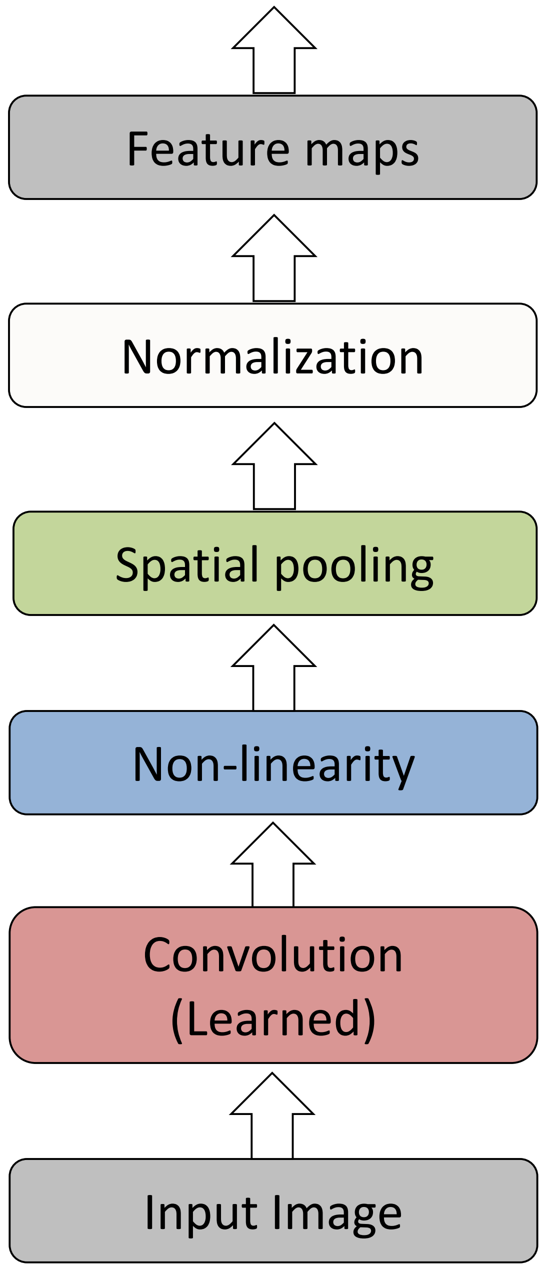



Feature Maps


Feature Maps
After Contrast
Normalization

slide credit: S. Lazebnik

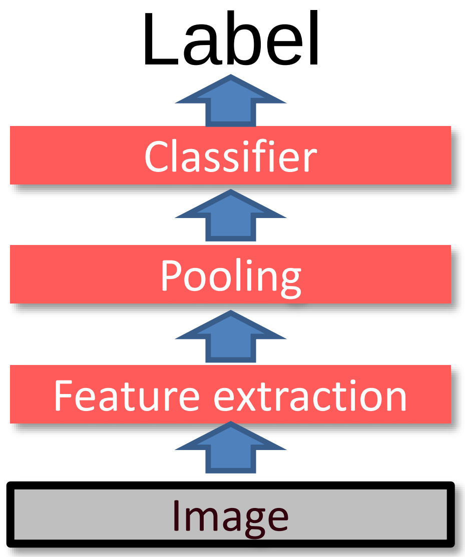
Convolutional Neural Networks

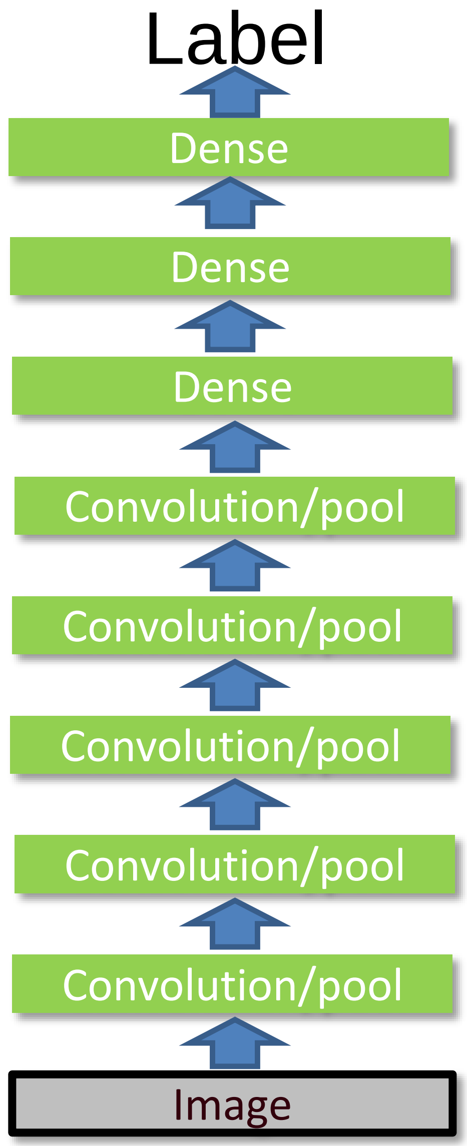


slide credit: S. Lazebnik

Engineered vs. learned features

Convolutional filters are trained in a supervised manner by back-propagating classification error



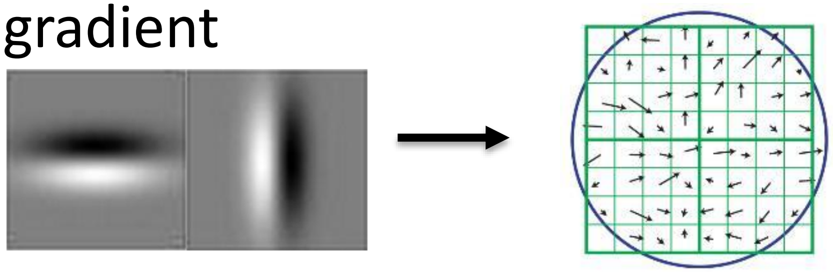


SIFT Descriptor

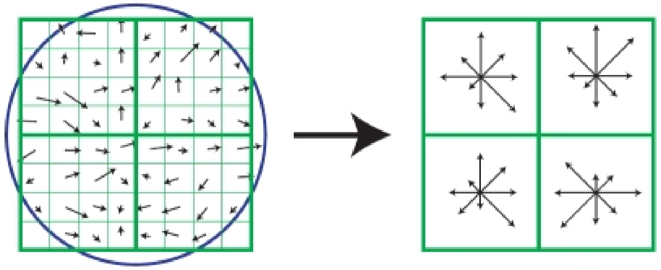
Lowe [IJCV 2004]

Image Pixels

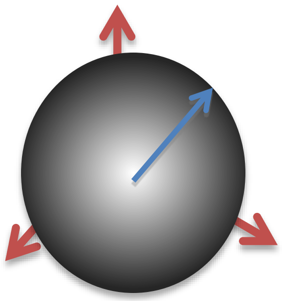
Apply gradient filters



Spatial pool (Sum)



Normalize to unit length



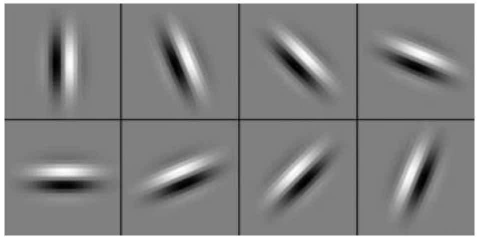
Feature Vector

SIFT Descriptor

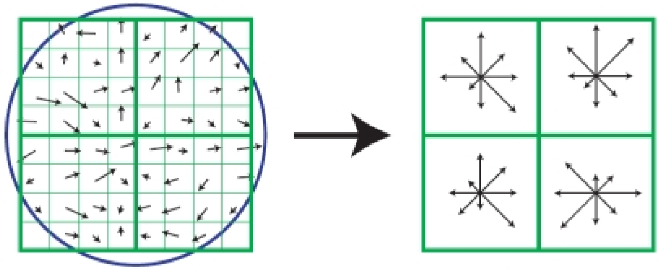
Lowe [IJCV 2004]

Image Pixels

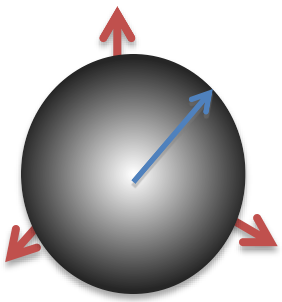
Apply oriented filters



Spatial pool (Sum)



Normalize to unit length



Feature Vector

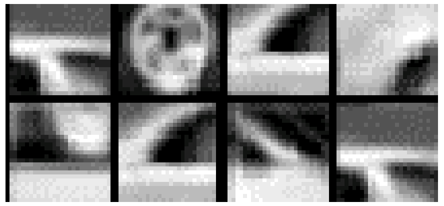
slide credit: R. Fergus

Spatial Pyramid Matching

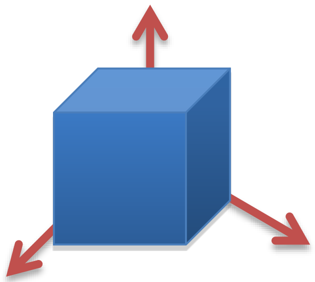
Lazebnik, Schmid, Ponce [CVPR 2006]

SIFT Features

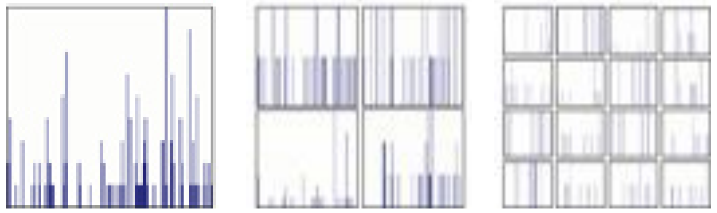
Filter with Visual Words



Max



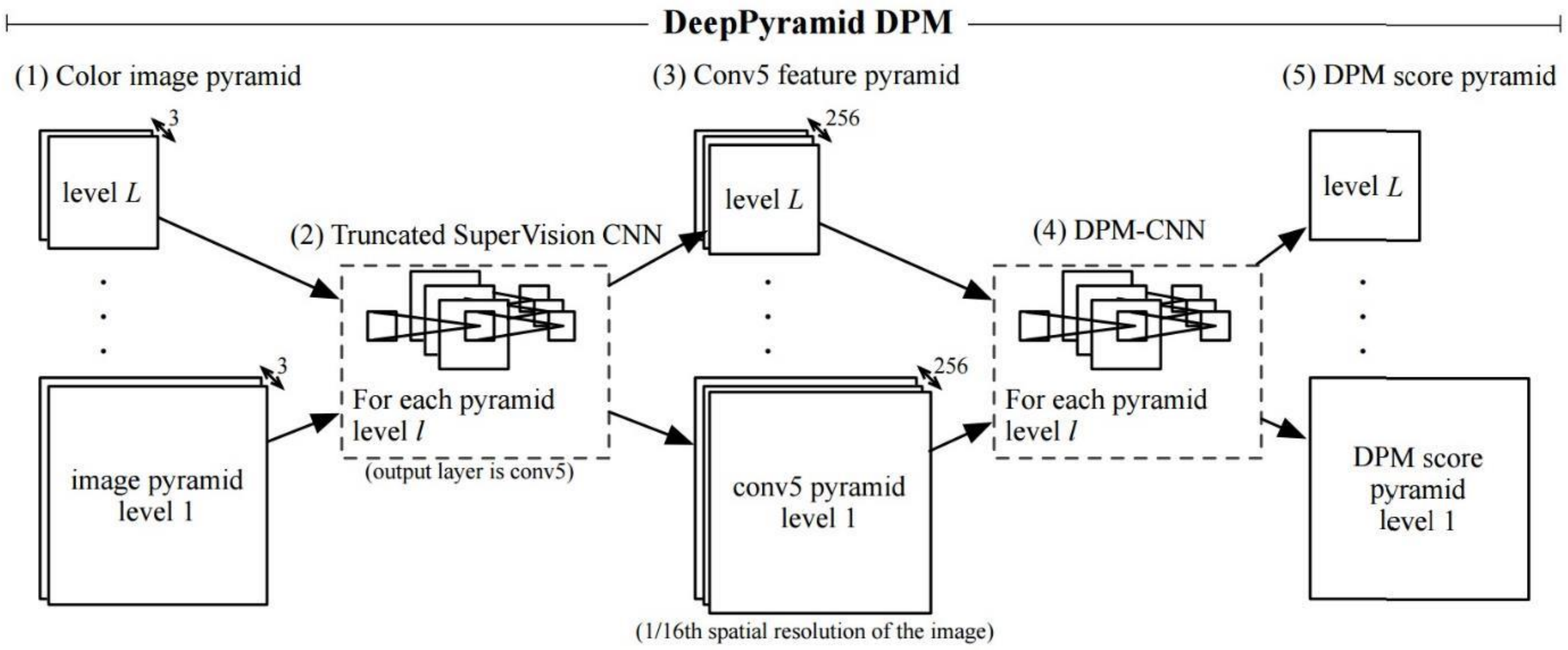
Multi-scale spatial pool (Sum)



Classifier

slide credit: R. Fergus

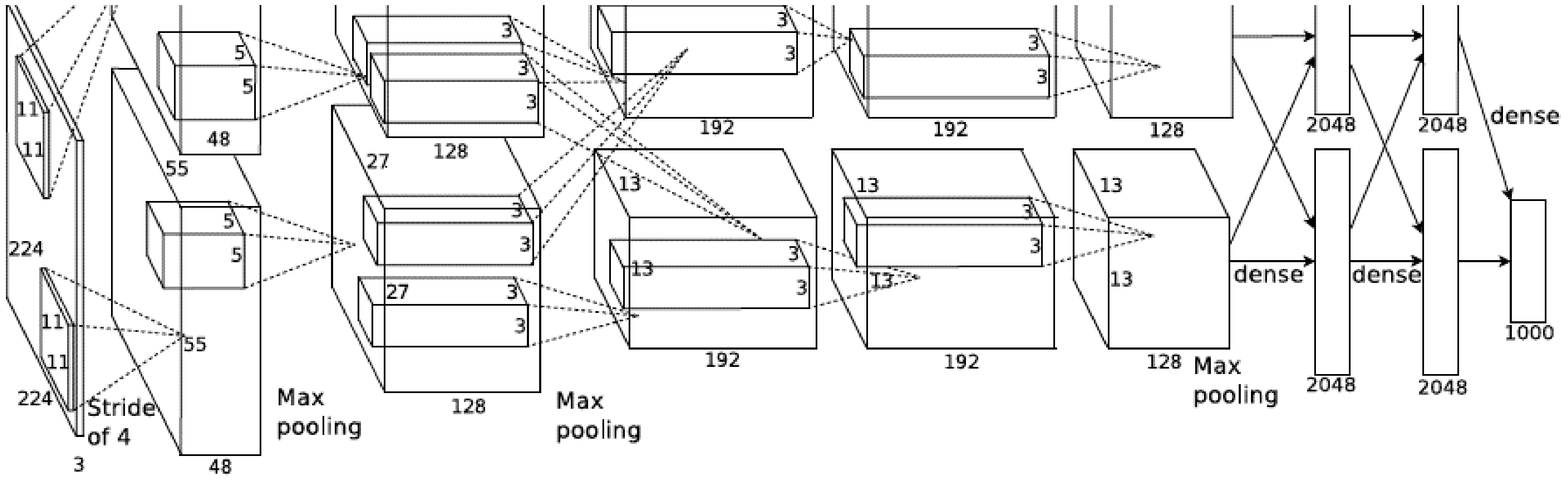
Deformable Part Model


DeepPyramid DPM
(1) Color image pyramid
(3) Conv5 feature pyramid
(5) DPM score pyramid
3
level L
256
level L
level L
(2) Truncated SuperVision CNN
(4) DPM-CNN
3
256
For each pyramid level l
For each pyramid level l
image pyramid level 1
(output layer is conv5)
conv5 pyramid level 1
DPM score pyramid level 1
(1/16th spatial resolution of the image)

Deformable Part Models are Convolutional Neural Networks [Girshick et al. CVPR 15]

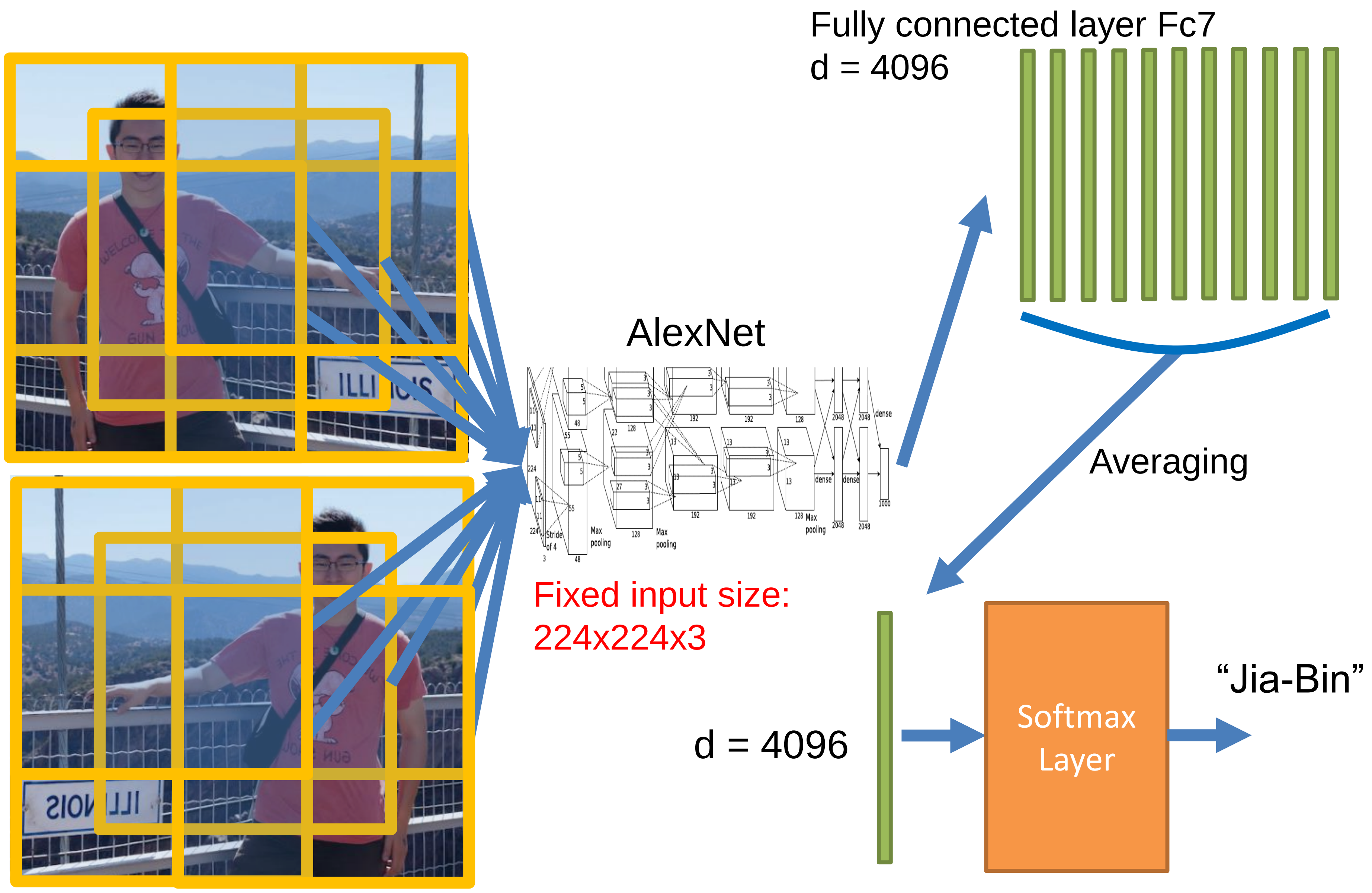
AlexNet

- Similar framework to LeCun'98 but:
  - Bigger model (7 hidden layers, 650,000 units, 60,000,000 params)
  - More data (10^6 vs. 10^3 images)
  - GPU implementation (50x speedup over CPU)
    - Trained on two GPUs for a week



A. Krizhevsky, I. Sutskever, and G. Hinton,
ImageNet Classification with Deep Convolutional Neural Networks, NIPS 2012

Using CNN for Image Classification



ImageNet Challenge 2012-2014

Best non-convnet in 2012: 26.2%

Team	Year	Place	Error (top-5)	External data
SuperVision – Toronto (7 layers)	2012	-	16.4%	no
SuperVision	2012	1st	15.3%	ImageNet 22k
Clarifai – NYU (7 layers)	2013	-	11.7%	no
Clarifai	2013	1st	11.2%	ImageNet 22k
VGG – Oxford (16 layers)	2014	2nd	7.32%	no
GoogLeNet (19 layers)	2014	1st	6.67%	no
Human expert*			5.1%	

Team	Method	Error (top-5)
DeepImage - Baidu	Data augmentation + multi GPU	5.33%
PReLU-nets - MSRA	Parametric ReLU + smart initialization	4.94%
BN-Inception ensemble - Google	Reducing internal covariate shift	4.82%

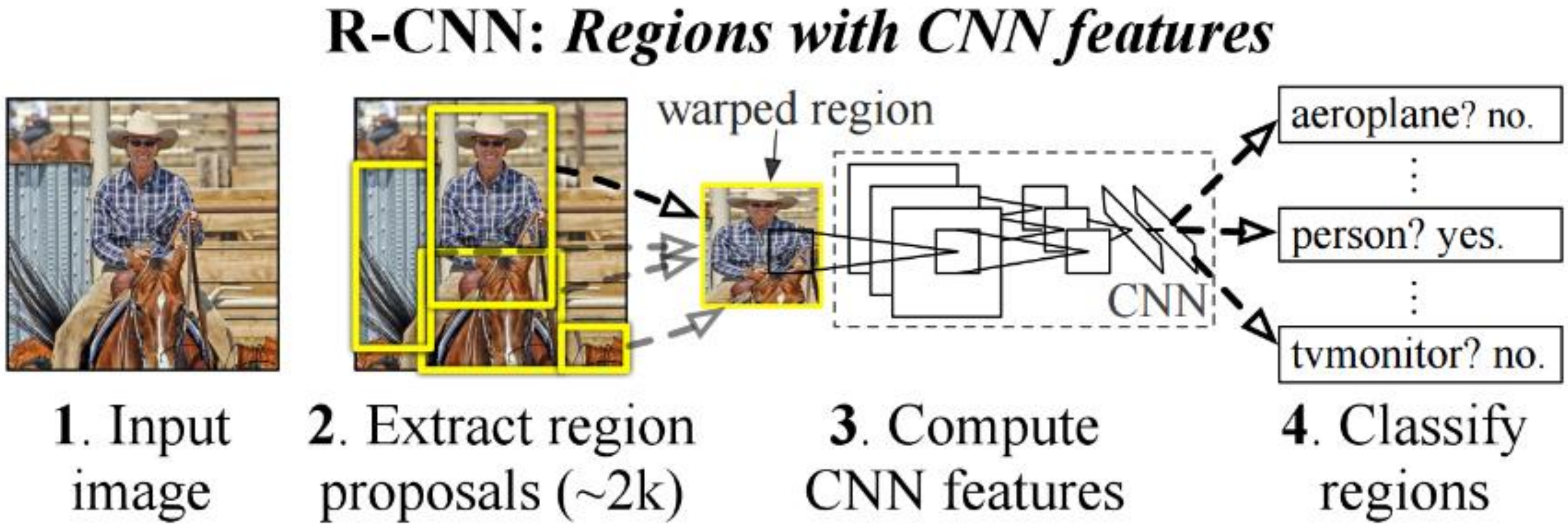
Beyond classification

- Detection
- Segmentation
- Regression
- Pose estimation
- Matching patches
- Synthesis

and many more…

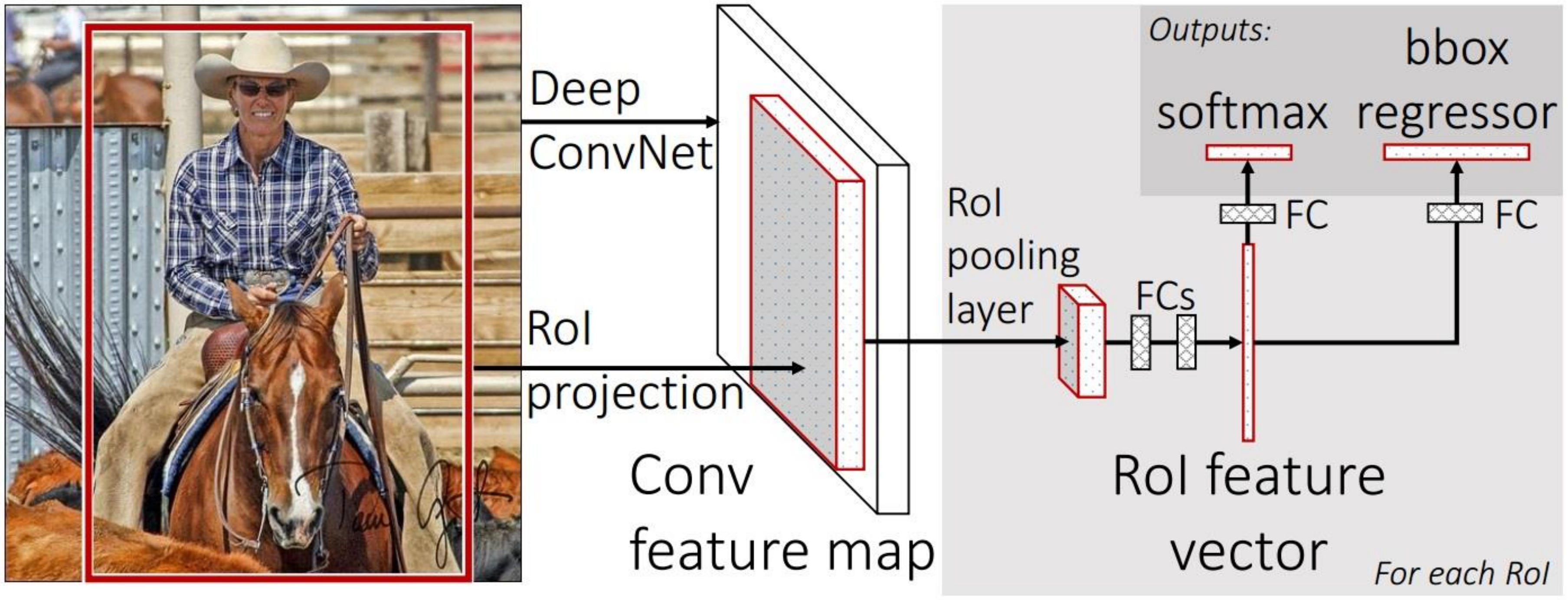
R-CNN: Regions with CNN features

- Trained on ImageNet classification
- Finetune CNN on PASCAL



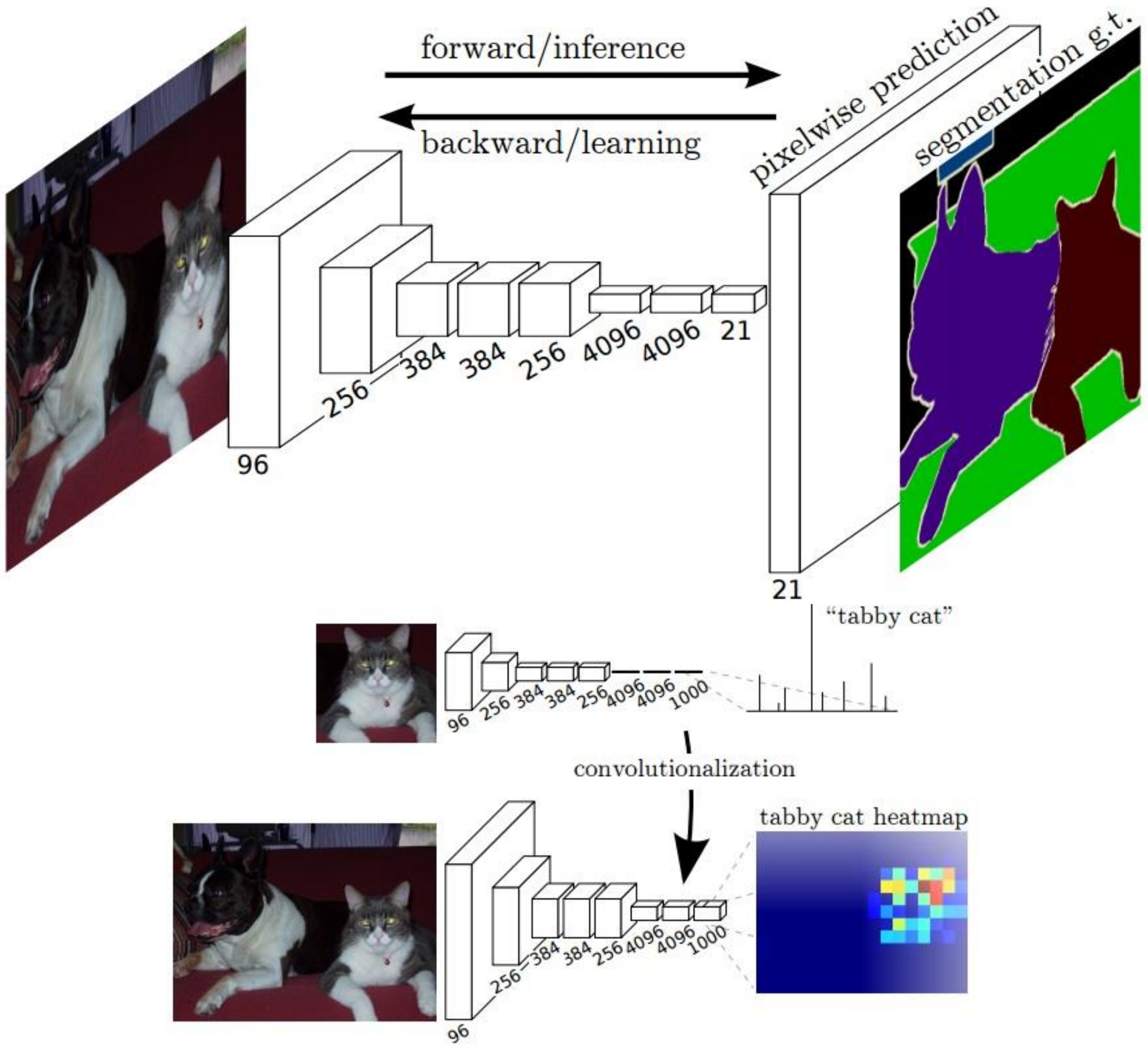
RCNN [Girshick et al. CVPR 2014]

Fast R-CNN



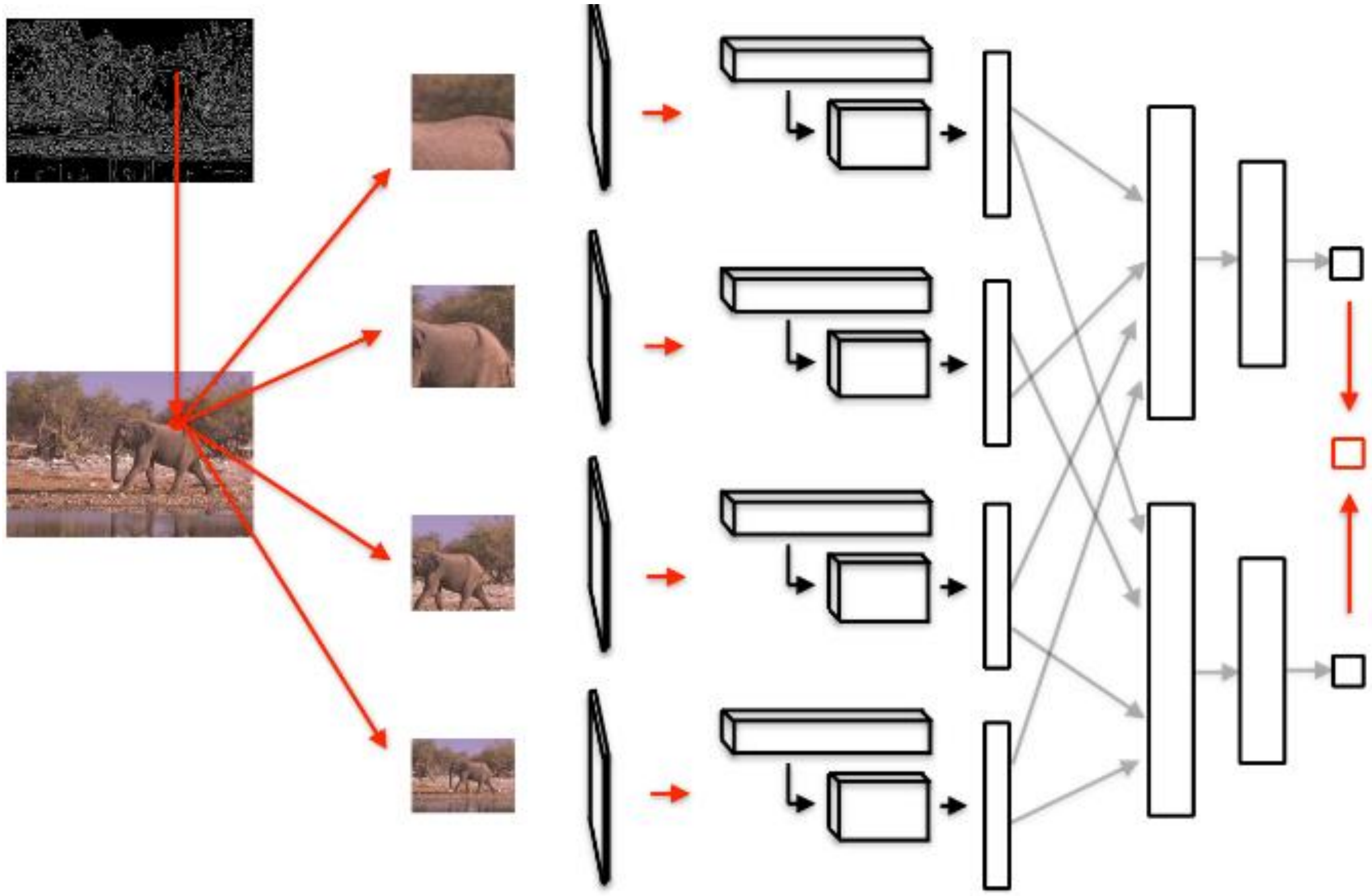
Fast RCNN [Girshick, R 2015]
https://github.com/rbgirshick/fast-rcnn

Labeling Pixels: Semantic Labels



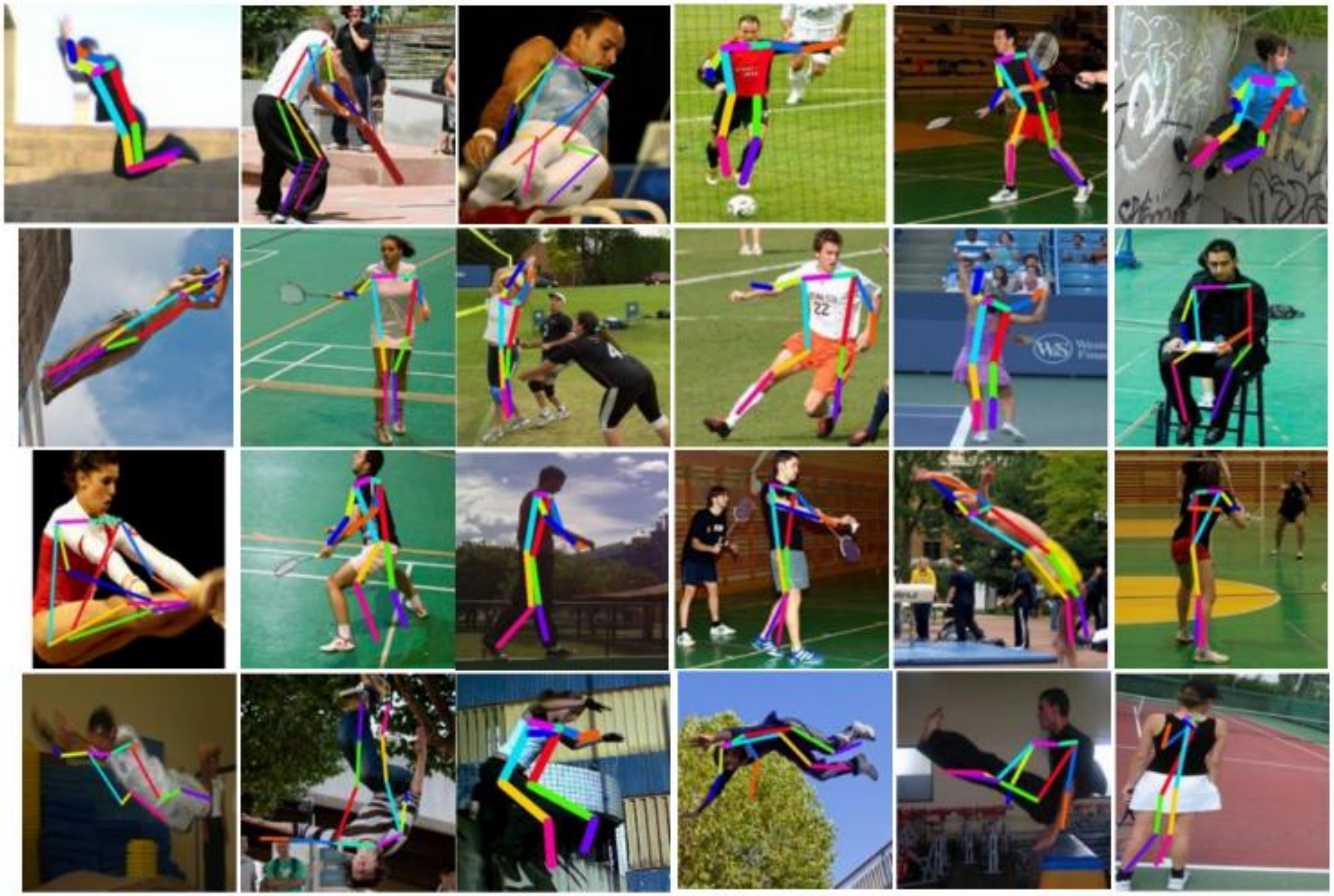
Fully Convolutional Networks for Semantic Segmentation [Long et al. CVPR 2015]

Labeling Pixels: Edge Detection



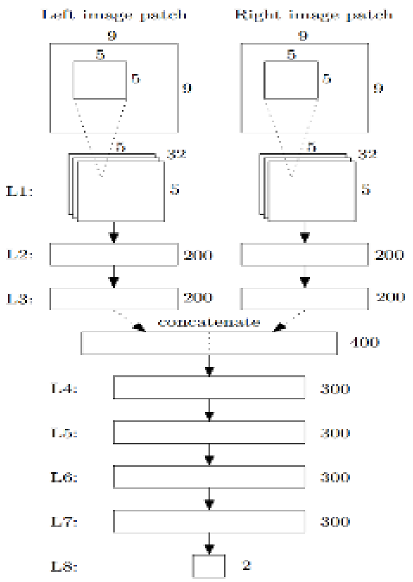
DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection
[Bertasius et al. CVPR 2015]

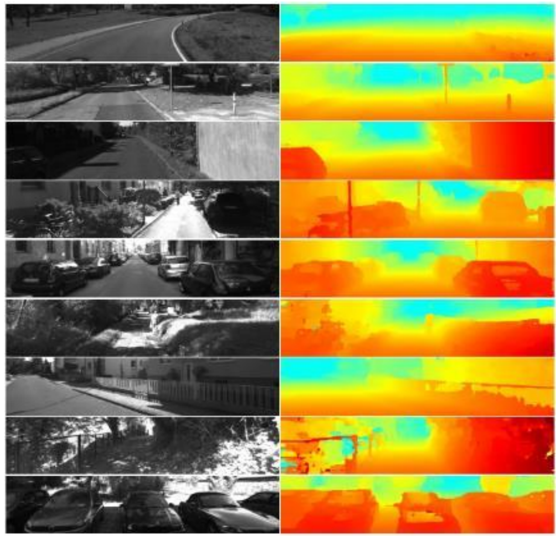
CNN for Regression



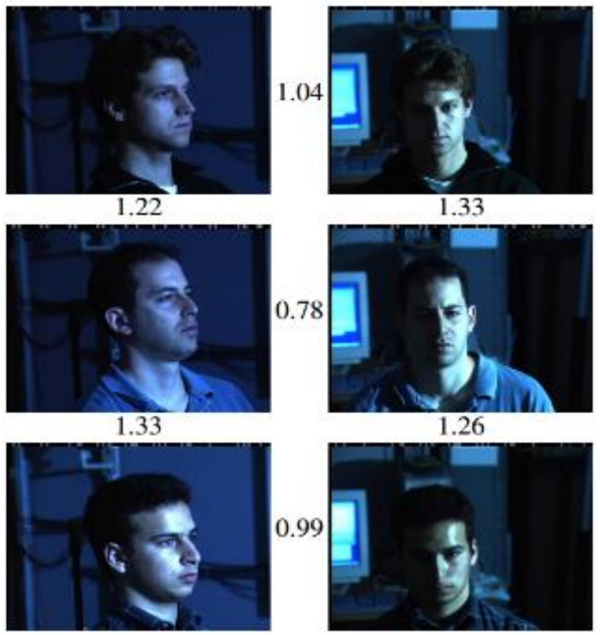
DeepPose [Toshev and Szegedy CVPR 2014]

CNN as a Similarity Measure for Matching

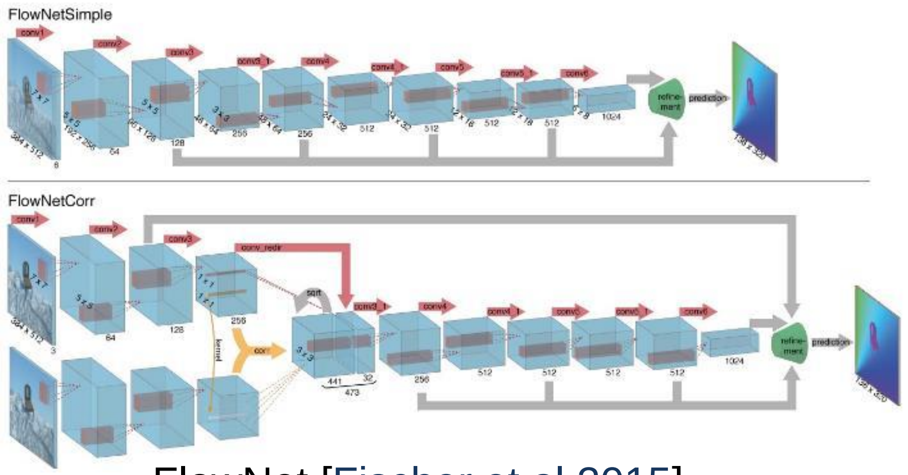




Stereo matching [Zbontar and LeCun CVPR 2015]
Compare patch [Zagoruyko and Komodakis 2015]



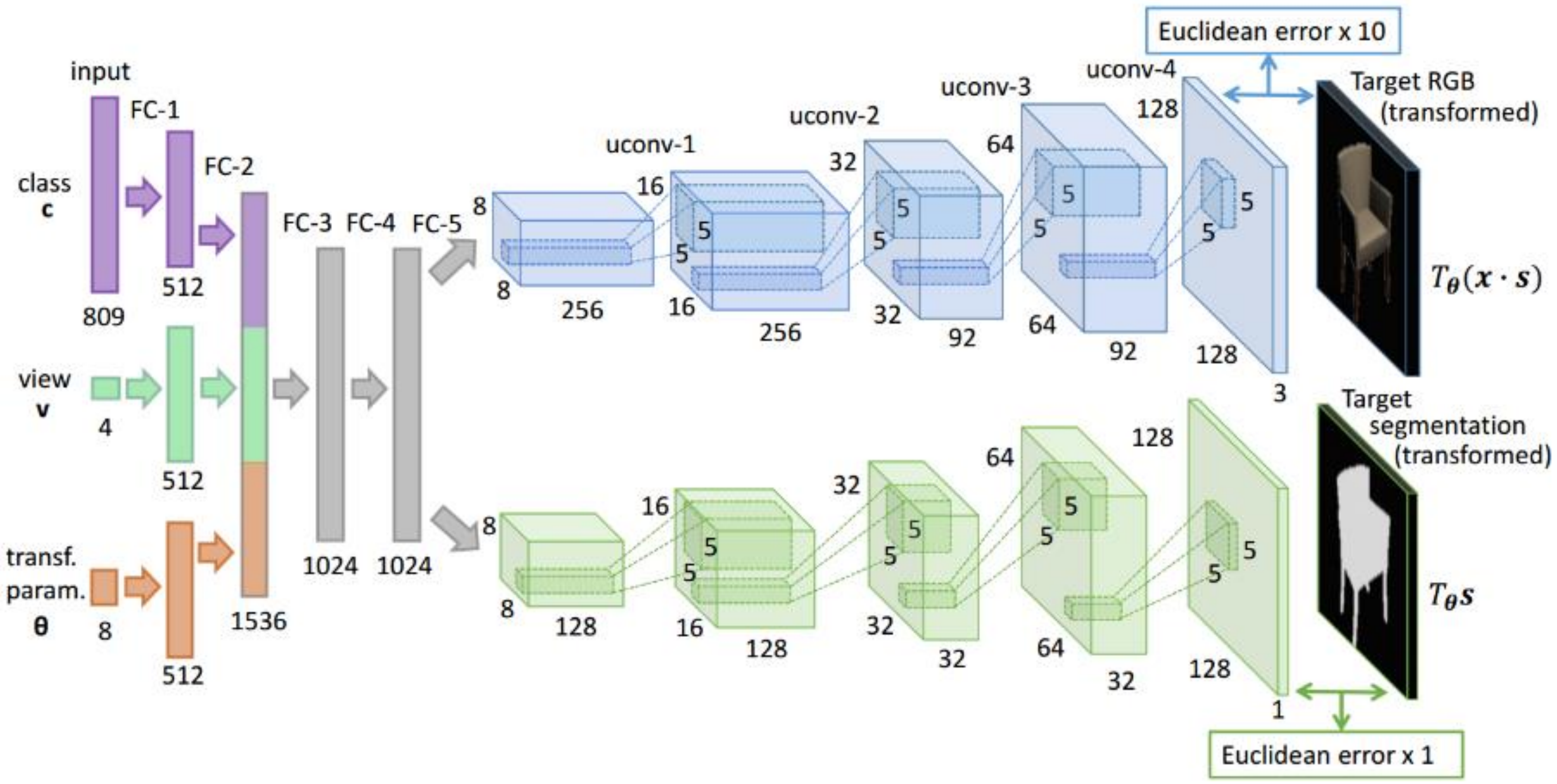
FaceNet [Schroff et al. 2015]



FlowNet [Fischer et al 2015]

Match ground and aerial images
[Lin et al. CVPR 2015]

CNN for Image Generation

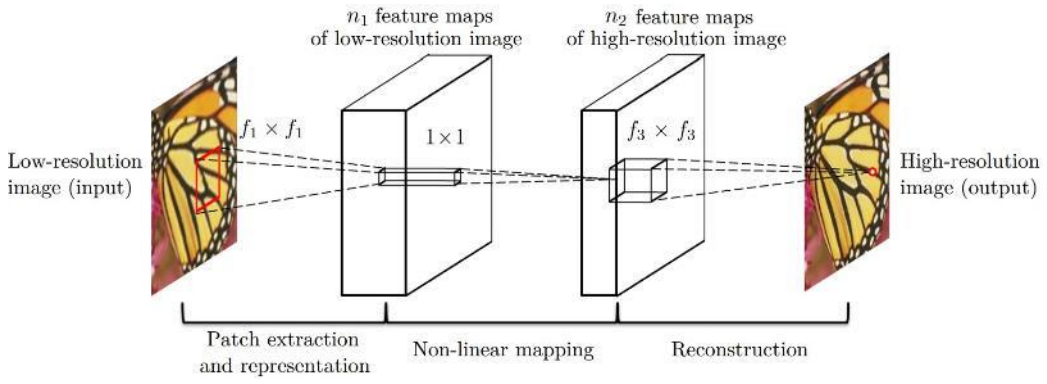


Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]

Chair Morphing

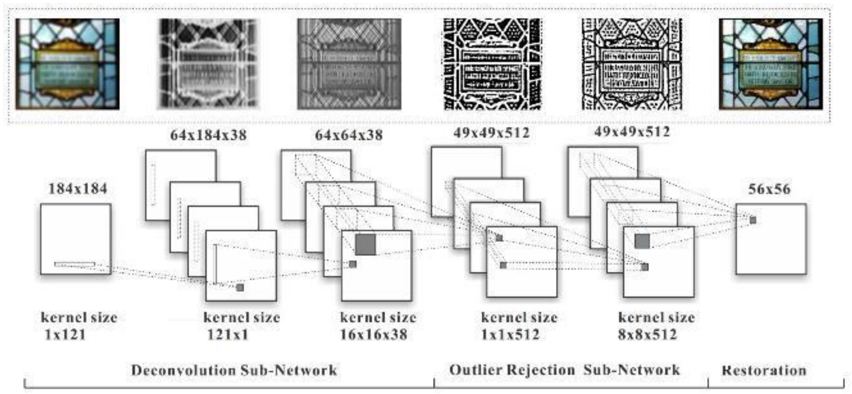
Learning to Generate Chairs with Convolutional Neural Networks [Dosovitskiy et al. CVPR 2015]

CNN for Image Restoration/Enhancement

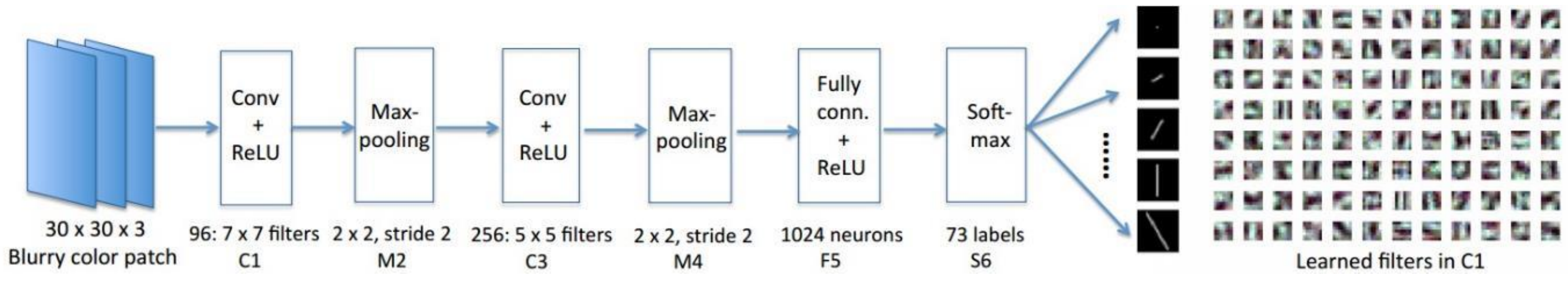


n_1 feature maps of low-resolution image
n_2 feature maps of high-resolution image
$f_1 \times f_1$
1×1
$f_3 \times f_3$
Low-resolution image (input)
High-resolution image (output)
Patch extraction and representation
Non-linear mapping
Reconstruction

Super-resolution
[Dong et al. ECCV 2014]



Non-blind deconvolution
[Xu et al. NIPS 2014]



Non-uniform blur estimation
[Sun et al. CVPR 2015]

Take a break…

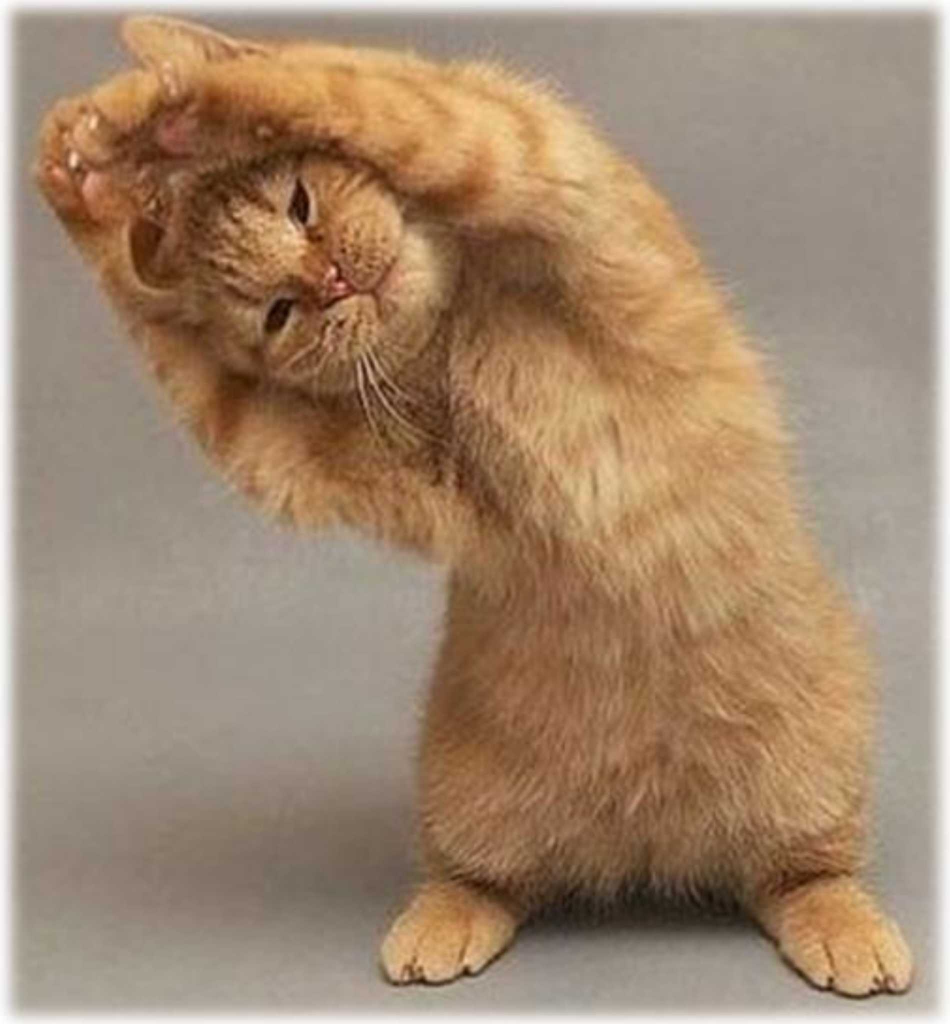
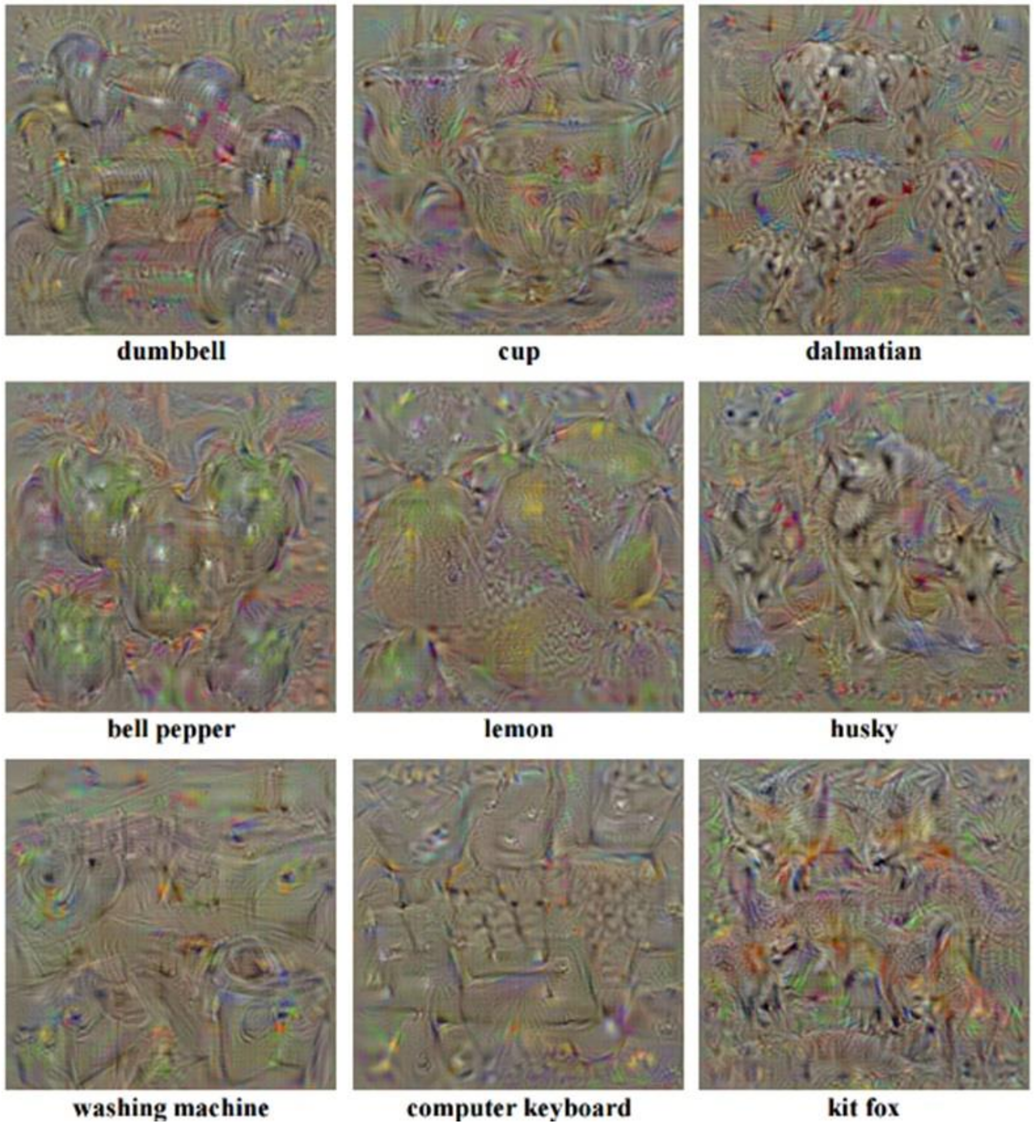


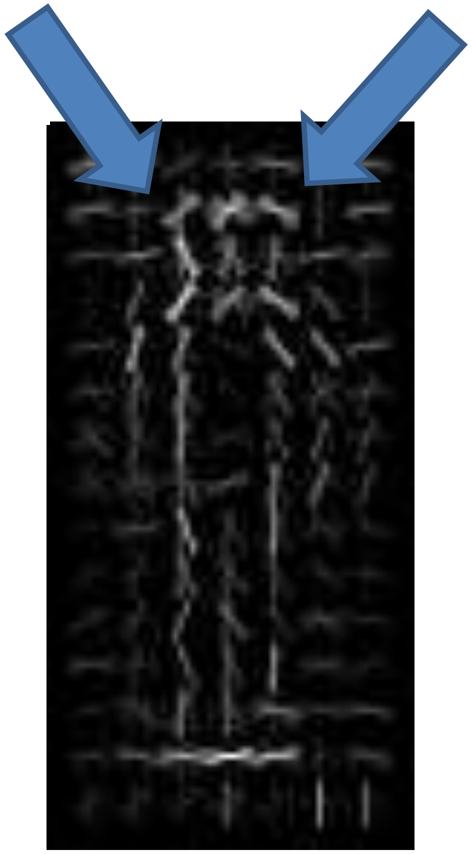
Image source: http://mehimandthecats.com/feline-care-guide/

Understanding and Visualizing CNN

- Find images that maximize some class scores
- Individual neuron activation
- Visualize input pattern using deconvnet
- Invert CNN features
- Breaking CNNs

Find images that maximize some class scores

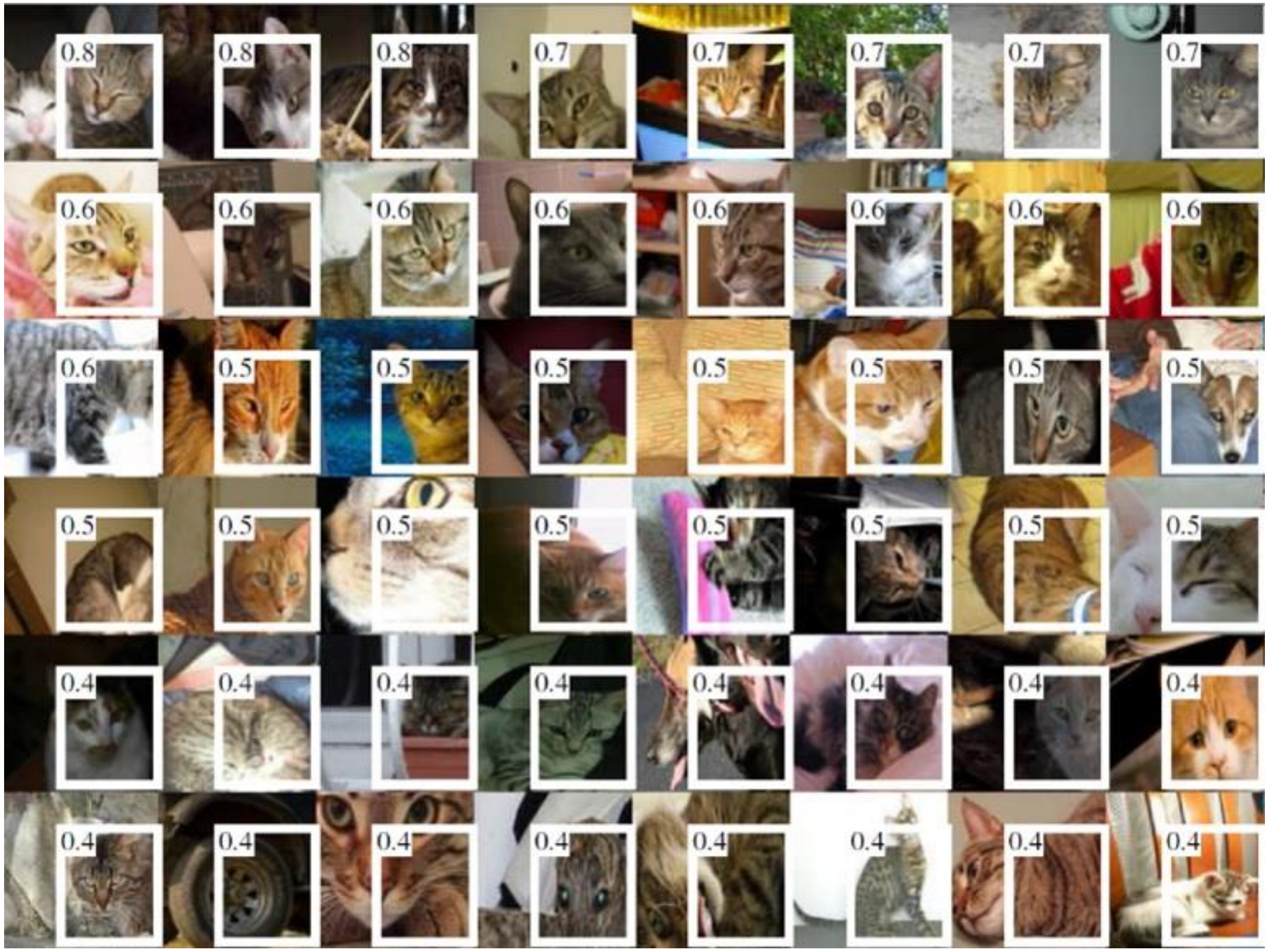




person: HOG template

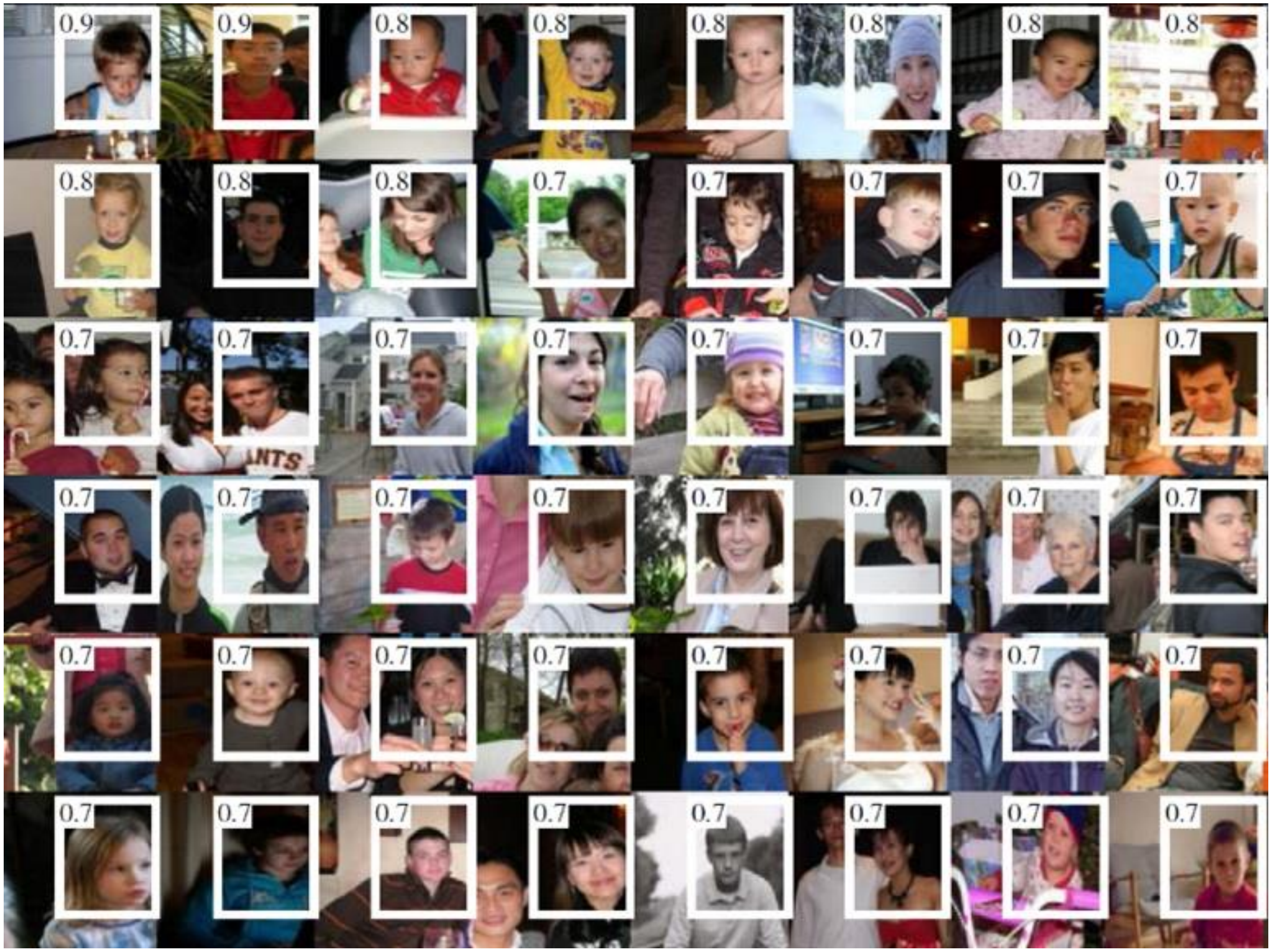
Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps
[Simonyan et al. ICLR Workshop 2014]

Individual Neuron Activation



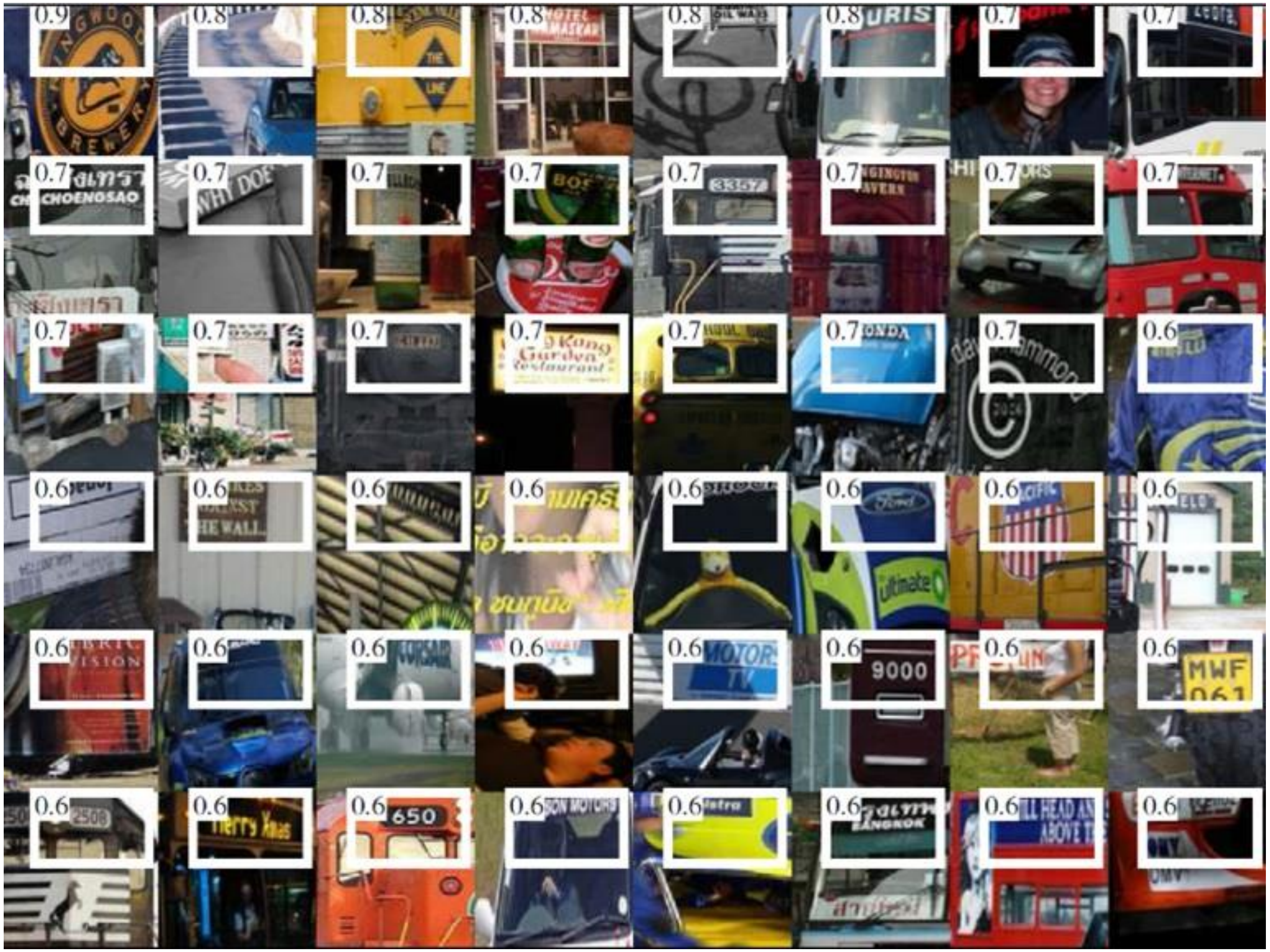
RCNN [Girshick et al. CVPR 2014]

Individual Neuron Activation



RCNN [Girshick et al. CVPR 2014]

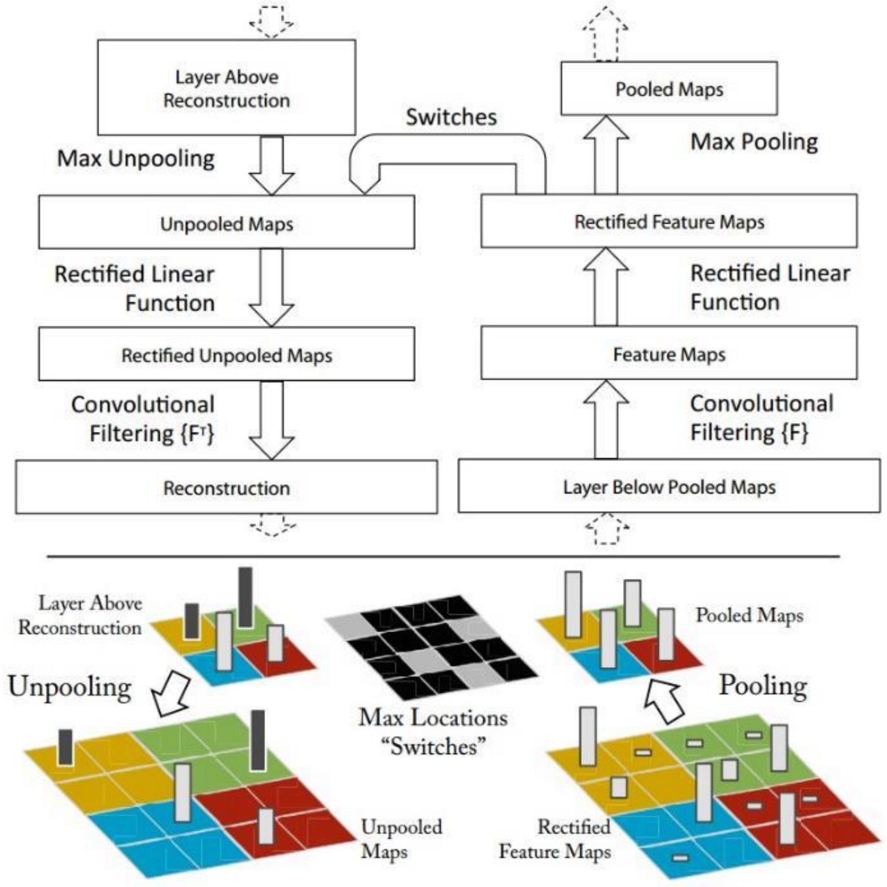
Individual Neuron Activation



RCNN [Girshick et al. CVPR 2014]

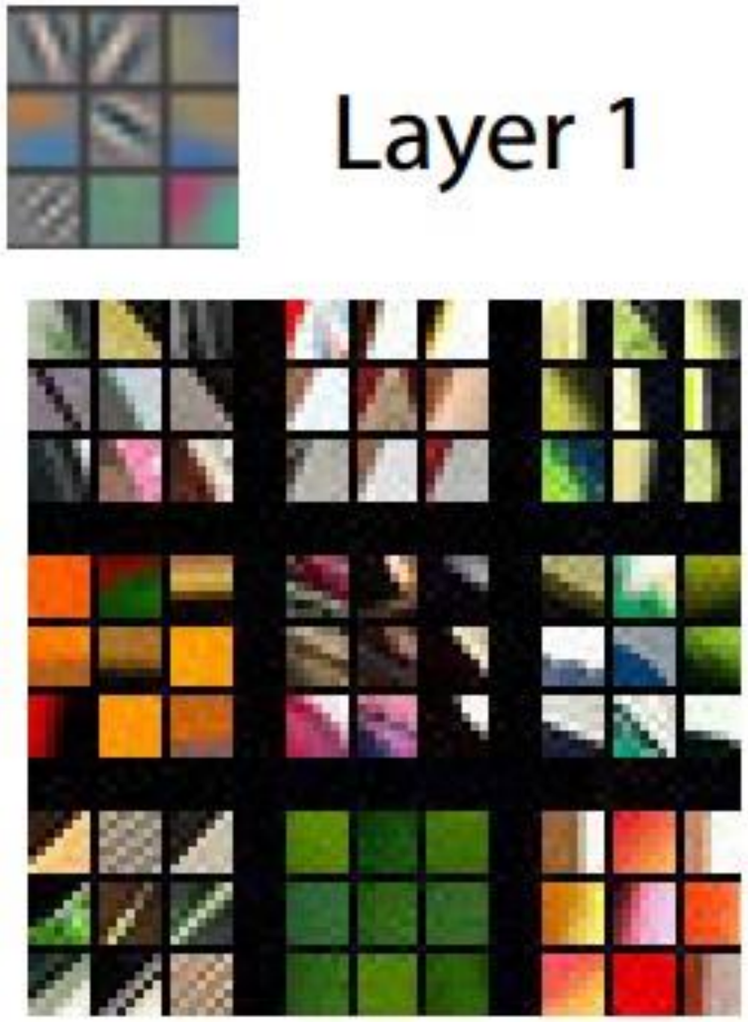
Map activation back to the input pixel space

- What input pattern originally caused a given activation in the feature maps?



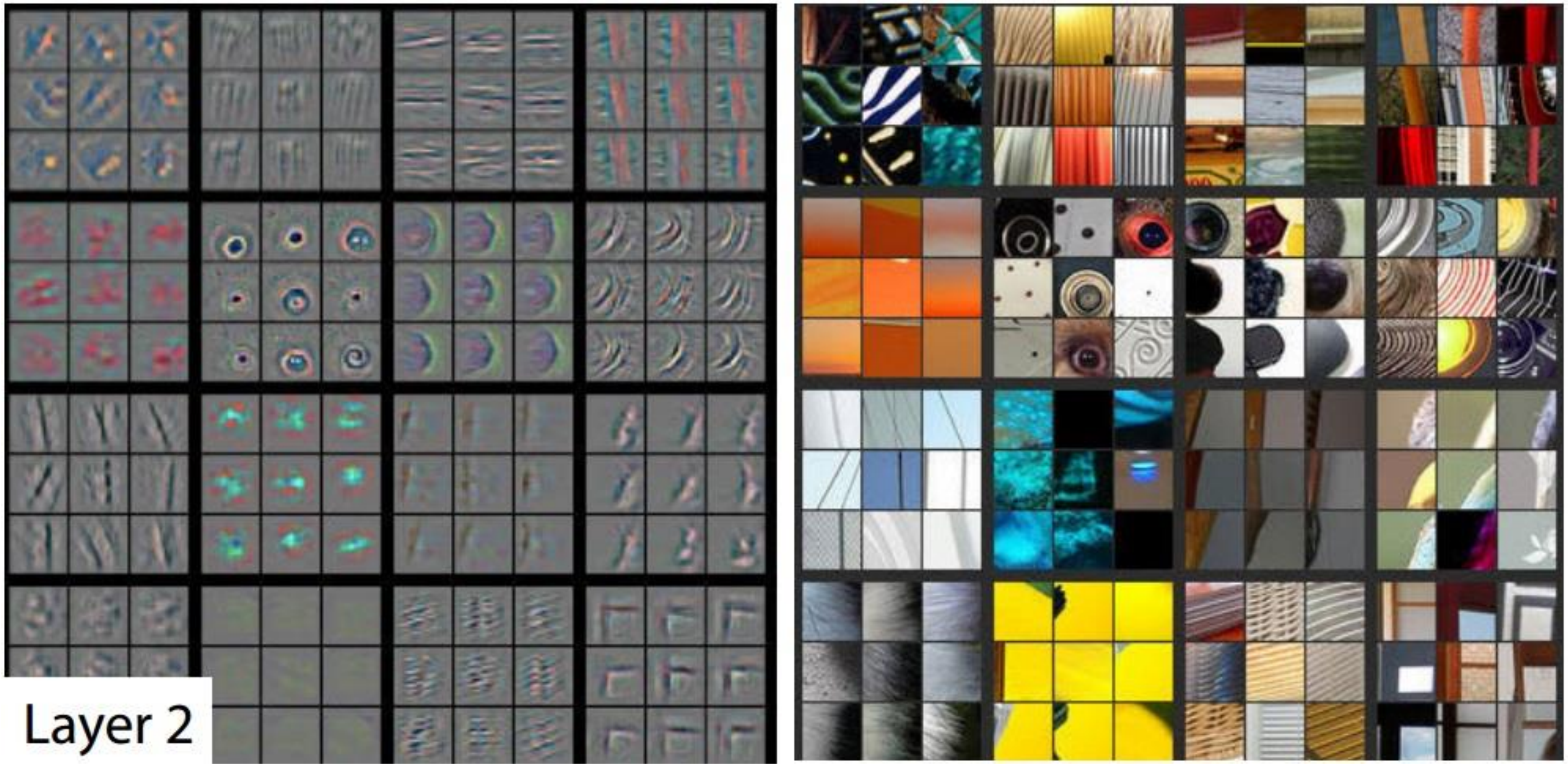
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 1



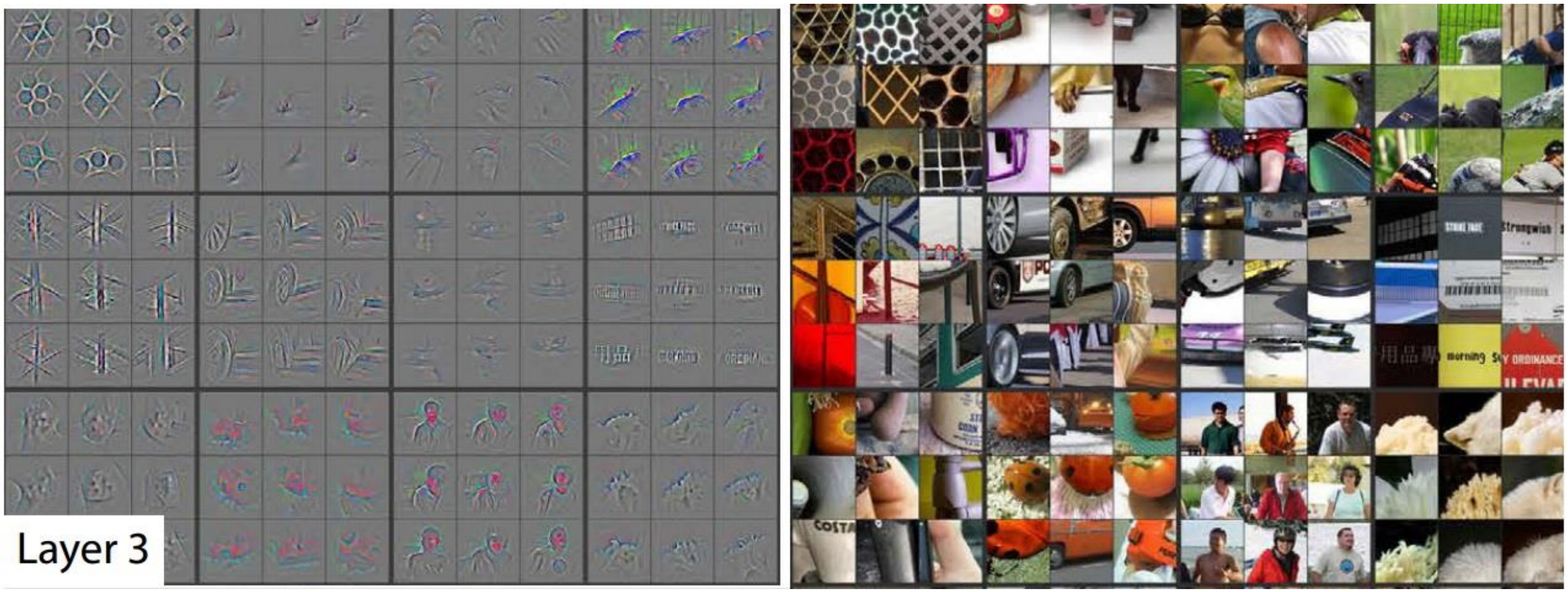
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 2



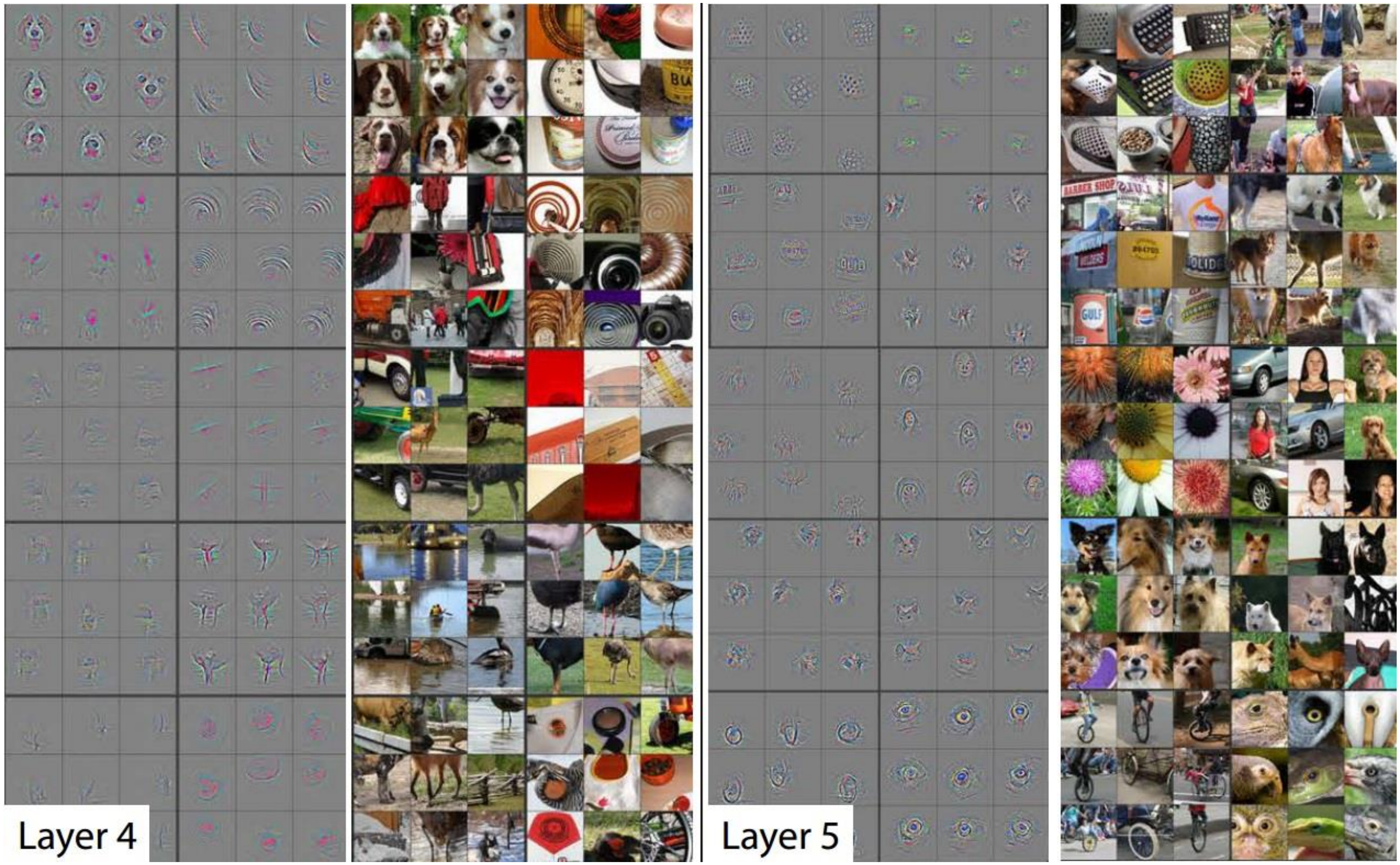
Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Layer 3



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

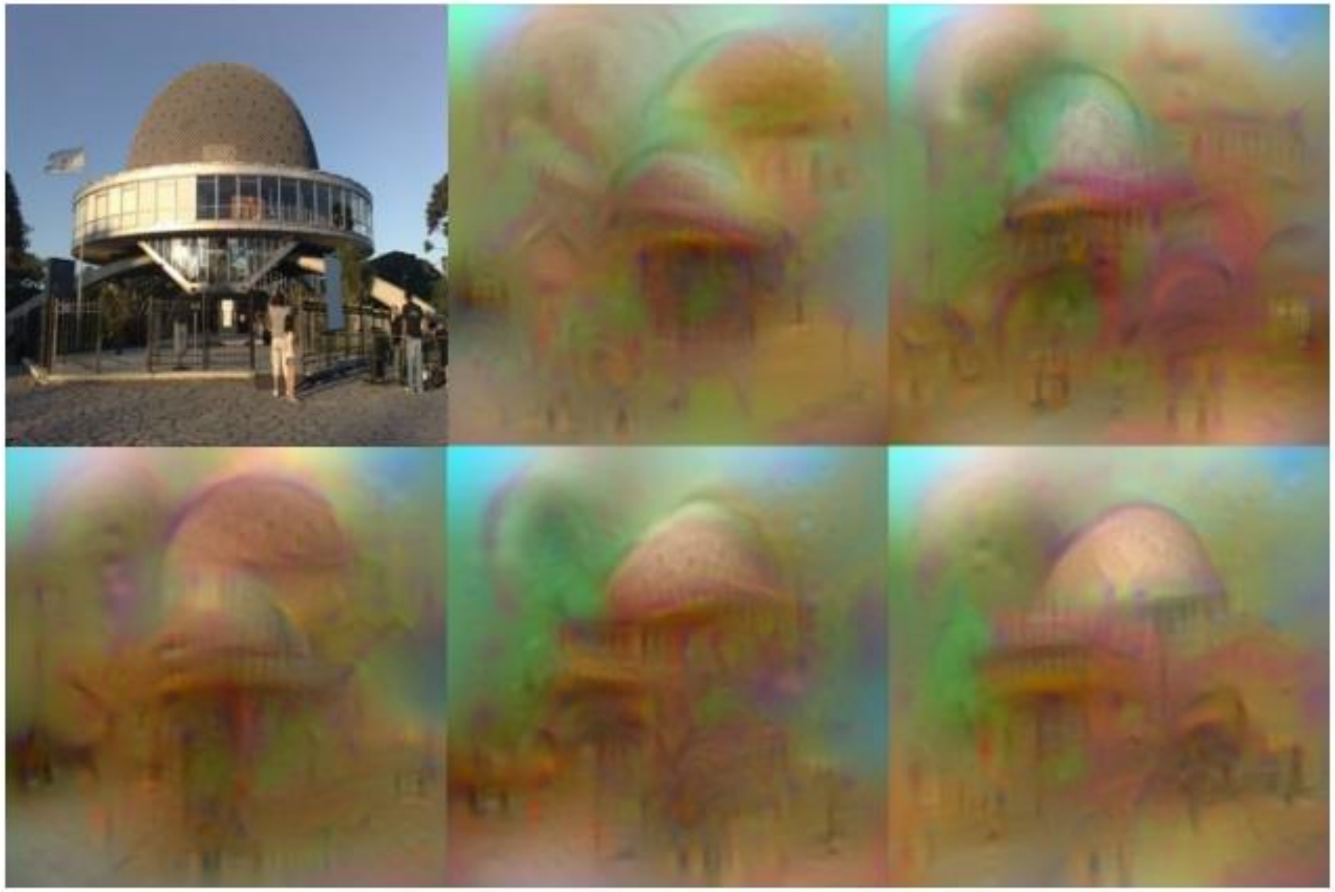
Layer 4 and 5



Visualizing and Understanding Convolutional Networks [Zeiler and Fergus, ECCV 2014]

Invert CNN features

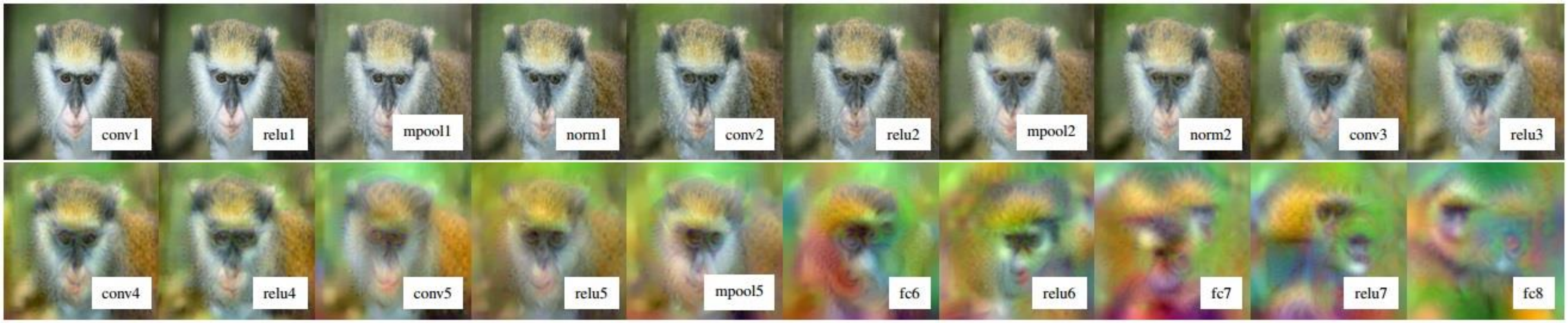
- Reconstruct an image from CNN features



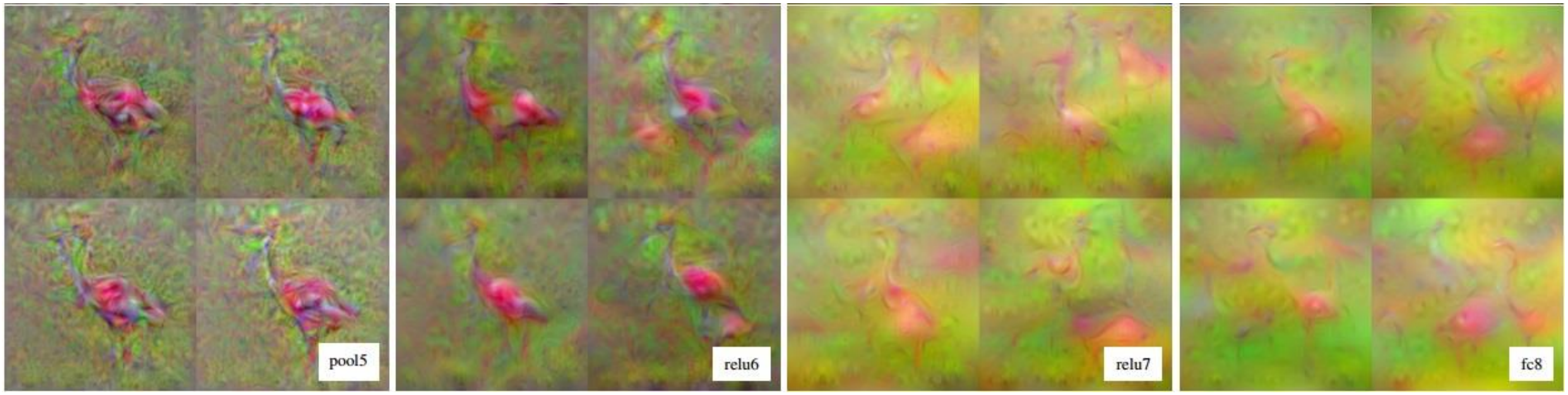
Understanding deep image representations by inverting them
[Mahendran and Vedaldi CVPR 2015]

CNN Reconstruction

Reconstruction from different layers

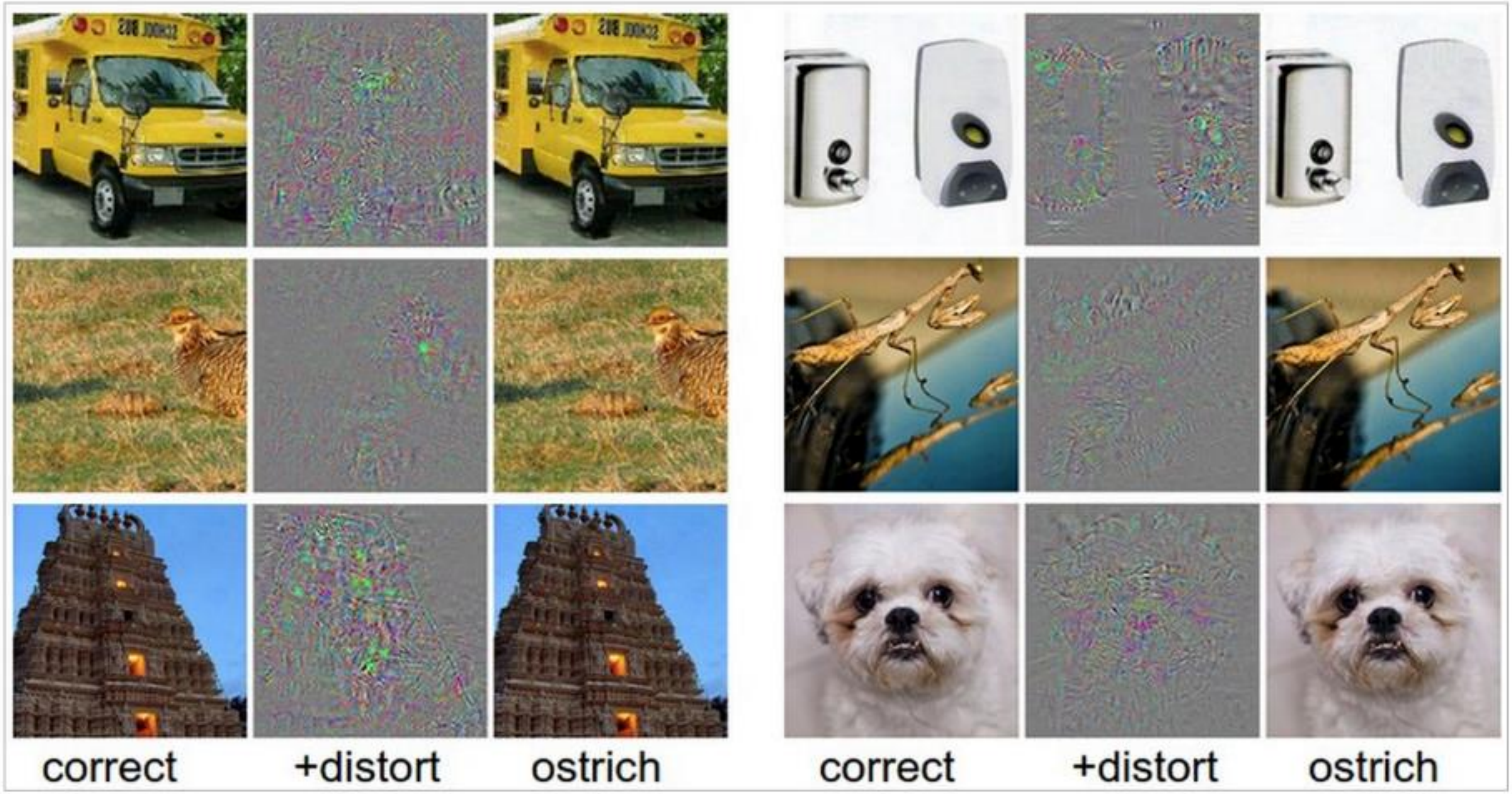


Multiple reconstructions



Understanding deep image representations by inverting them
[Mahendran and Vedaldi CVPR 2015]

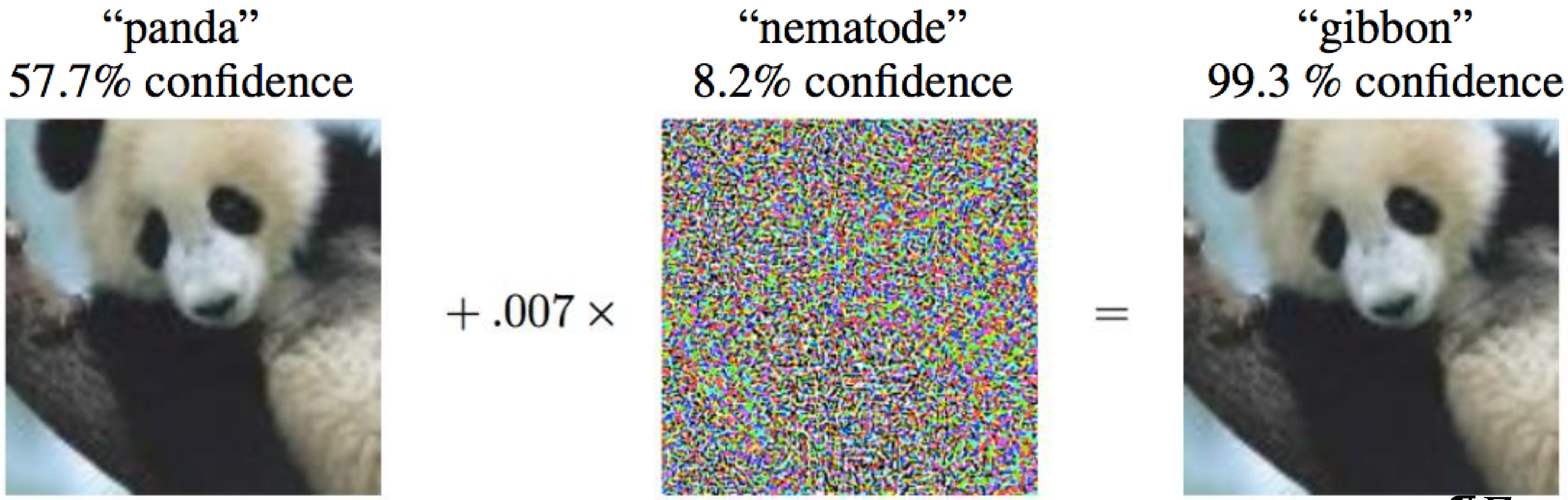
Breaking CNNs



Take a correctly classified image (left image in both columns), and add a tiny distortion (middle) to fool the ConvNet with the resulting image (right).

Intriguing properties of neural networks [Szegedy ICLR 2014]

What is going on?



x

$$\frac{¶E}{¶\mathbf{x}}$$

$$\mathbf{x} ¬ \quad \mathbf{x} + a\frac{¶E}{¶\mathbf{x}}$$

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]

http://karpathy.github.io/2015/03/30/breaking-convnets/

What is going on?

- Recall gradient descent training: modify the weights to reduce classifier error

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial E}{\partial \mathbf{w}}$$

- Adversarial examples: modify the *image* to *increase* classifier error

$$\mathbf{x} \neg \;\; \mathbf{x} + a \frac{\P E}{\P \mathbf{x}}$$

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]

http://karpathy.github.io/2015/03/30/breaking-convnets/

Fooling a linear classifier

- Perceptron weight update: add a small multiple of the example to the weight vector:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha\mathbf{x}$$

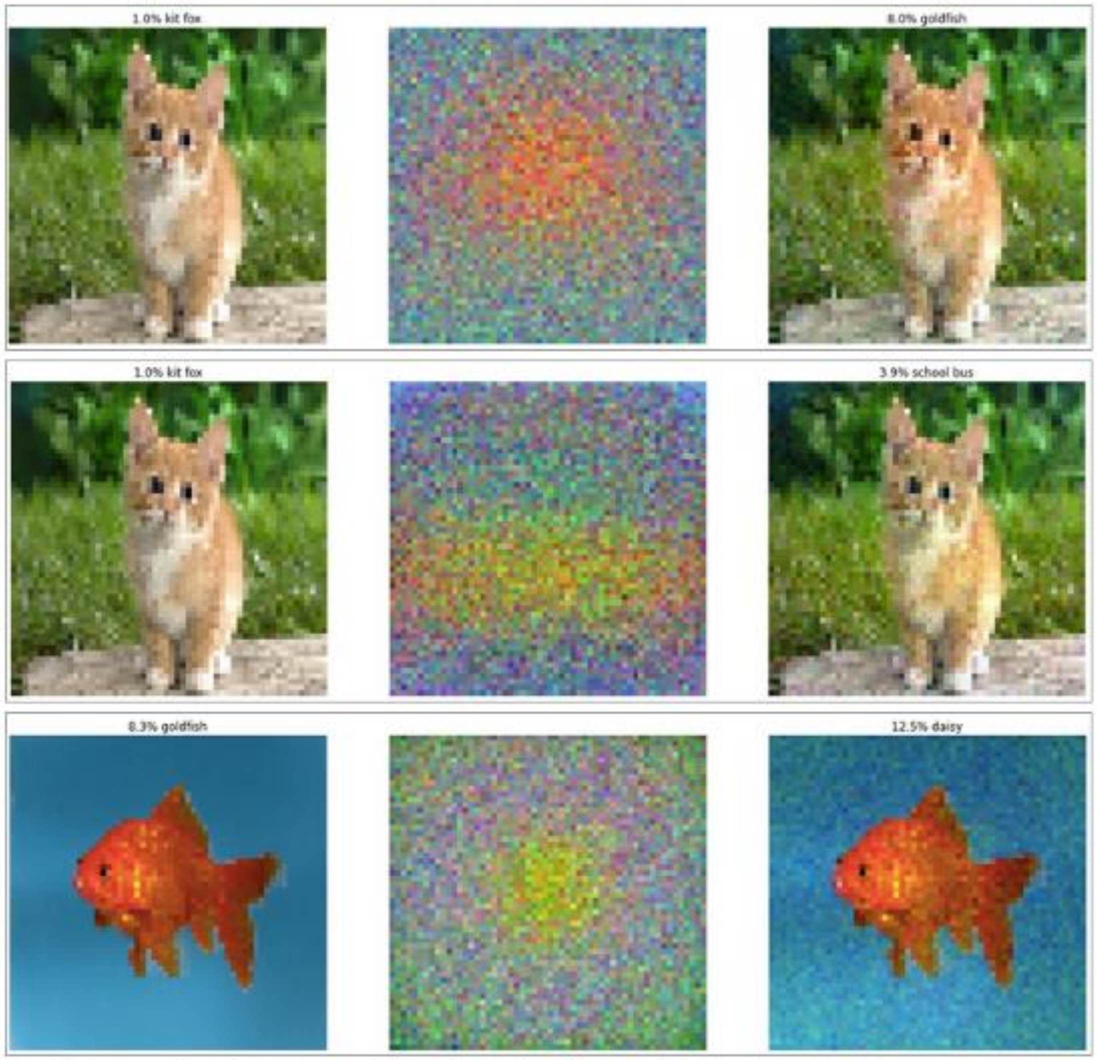
- To fool a linear classifier, add a small multiple of the weight vector to the training example:

$$\mathbf{x} \leftarrow \mathbf{x} + \alpha\mathbf{w}$$

Explaining and Harnessing Adversarial Examples [Goodfellow ICLR 2015]

http://karpathy.github.io/2015/03/30/breaking-convnets/

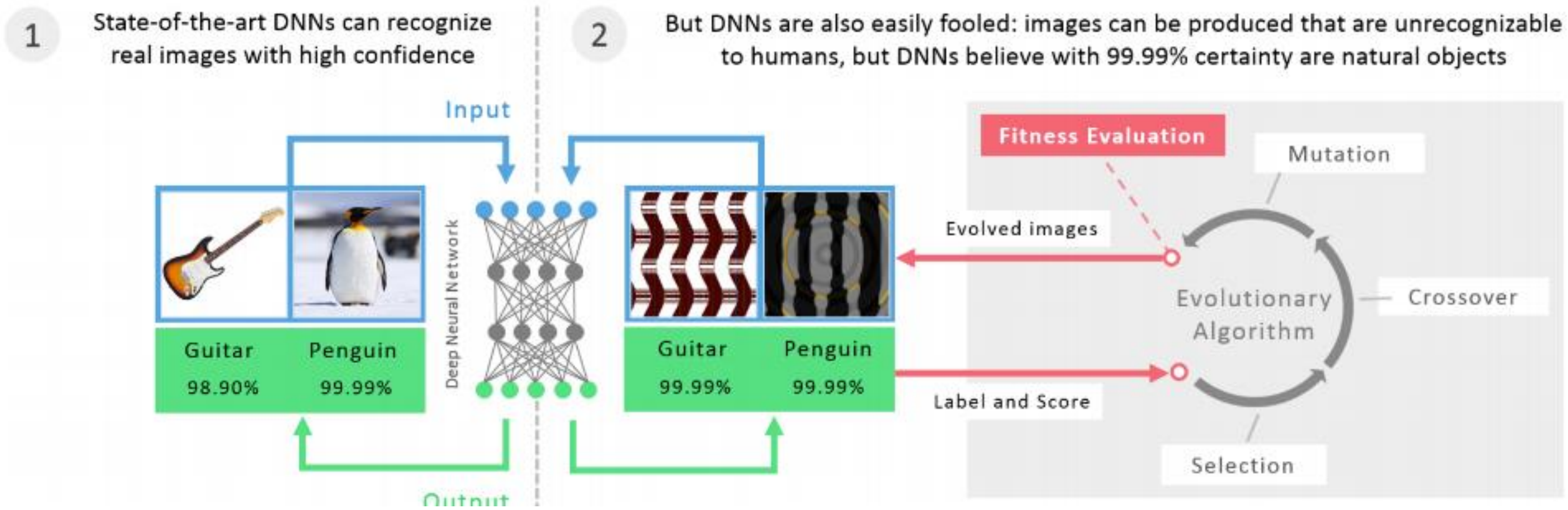
Fooling a linear classifier



Fooled linear classifier: The starting image (left) is classified as a kit fox. That's incorrect, but then what can you expect from a linear classifier? However, if we add a small amount "goldfish" weights to the image (top row, middle), suddenly the classifier is convinced that it's looking at one with high confidence. We can distort it with the school bus template instead if we wanted to.

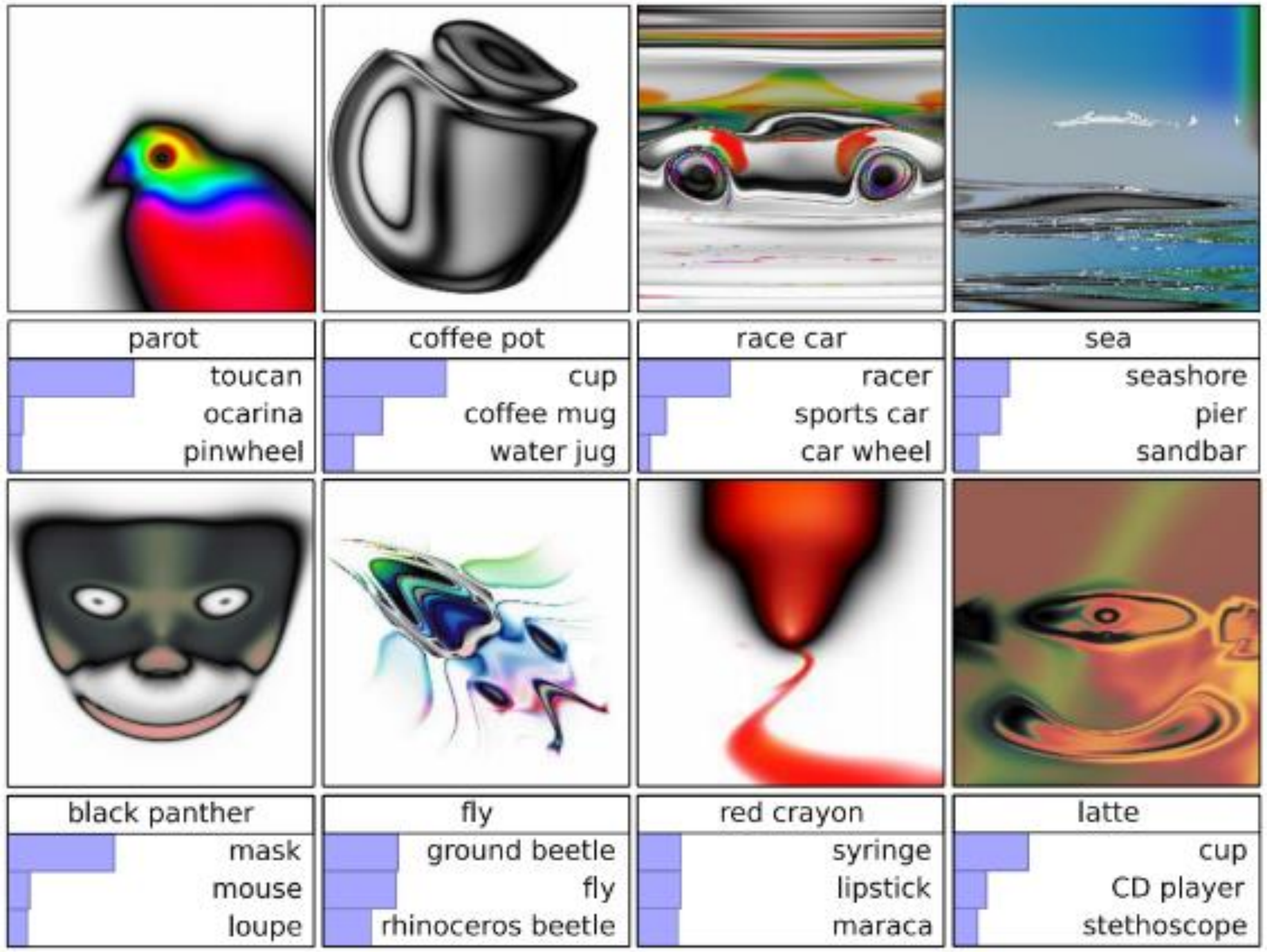
http://karpathy.github.io/2015/03/30/breaking-convnets/

Breaking CNNs


1
State-of-the-art DNNs can recognize real images with high confidence
2
But DNNs are also easily fooled: images can be produced that are unrecognizable to humans, but DNNs believe with 99.99% certainty are natural objects
Input
Deep Neural Network
Guitar
98.90%
Penguin
99.99%
Guitar
99.99%
Penguin
99.99%
Output
Fitness Evaluation
Mutation
Evolved images
Evolutionary Algorithm
Crossover
Label and Score
Selection

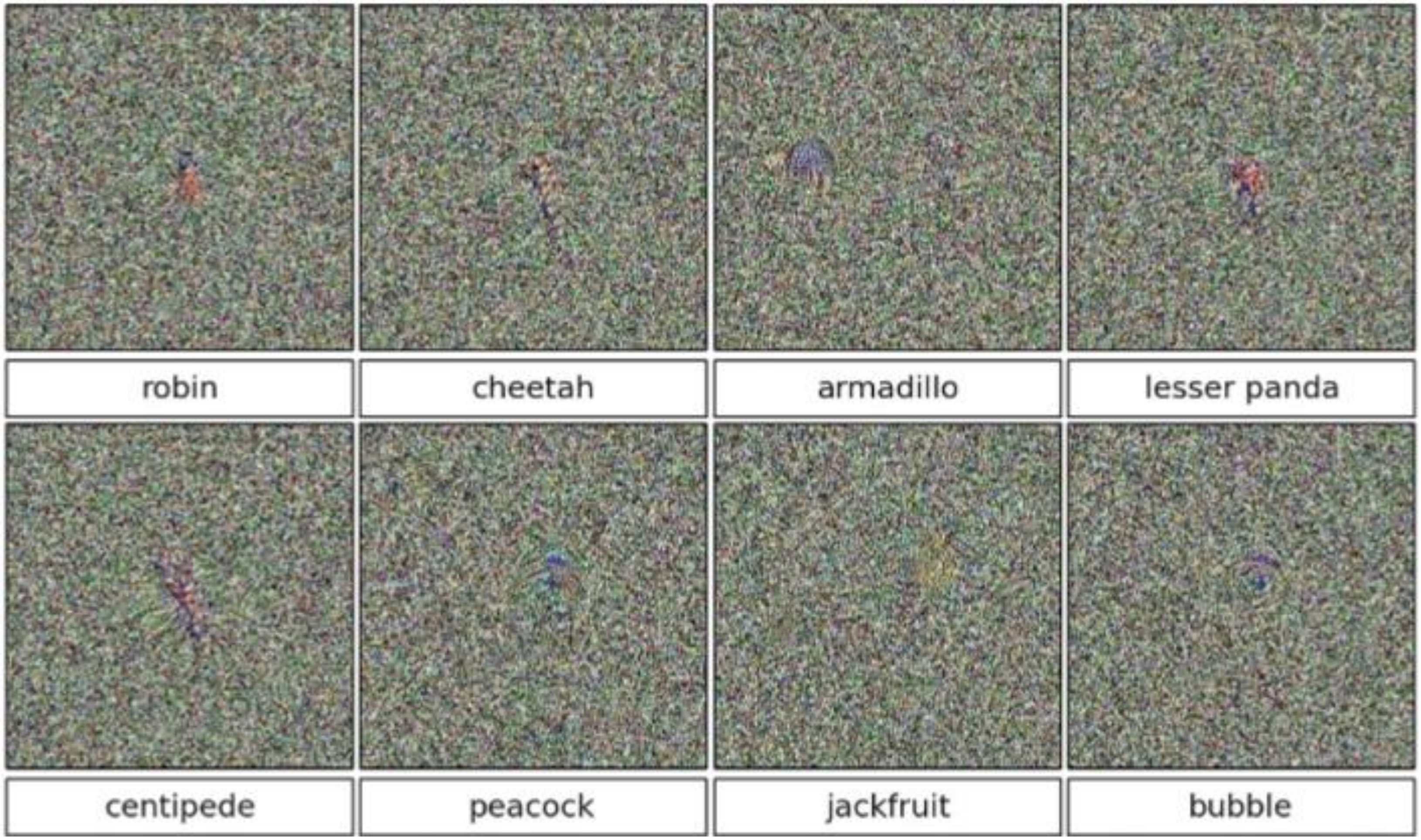
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Images that both CNN and Human can recognize



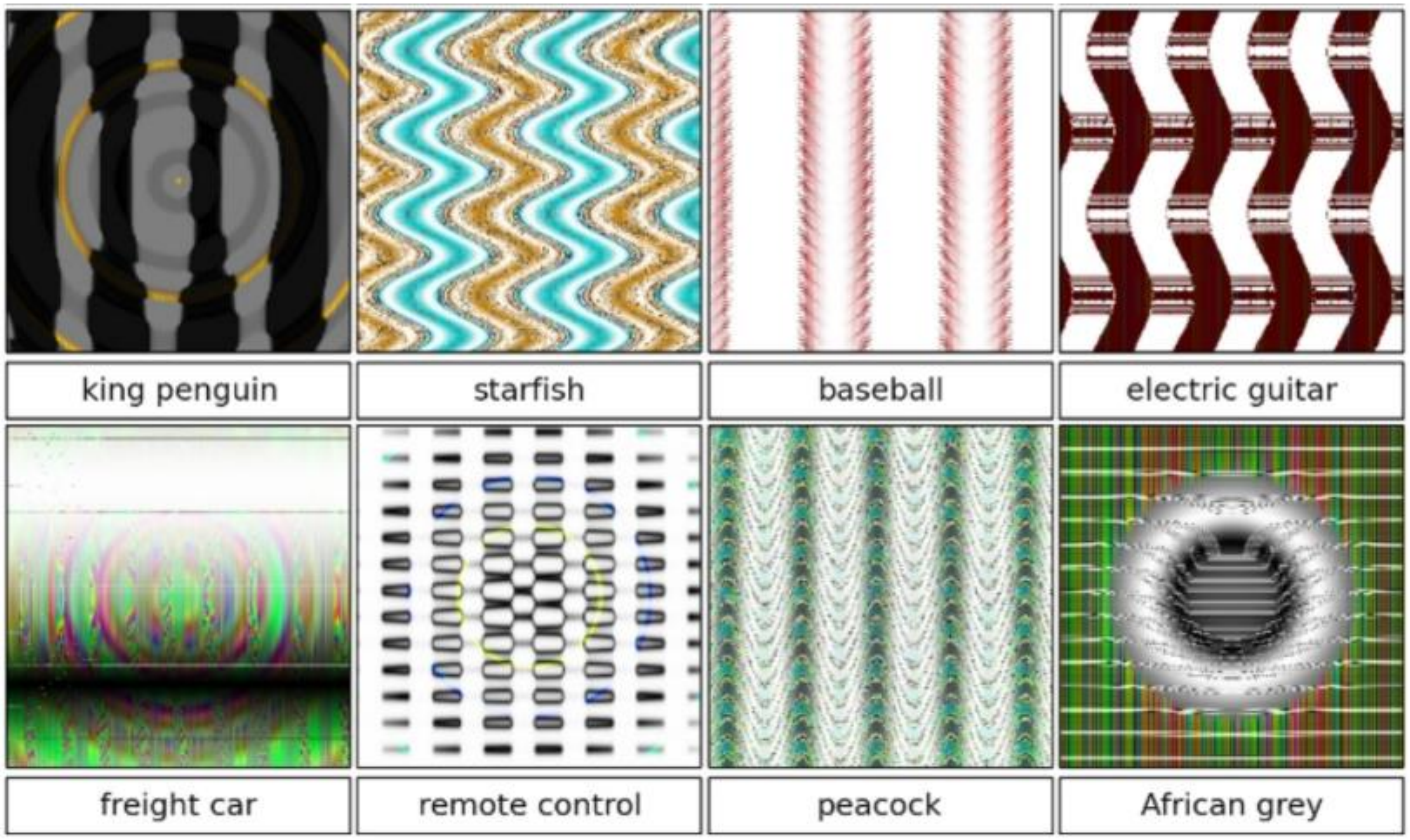
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Direct Encoding



Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

Indirect Encoding

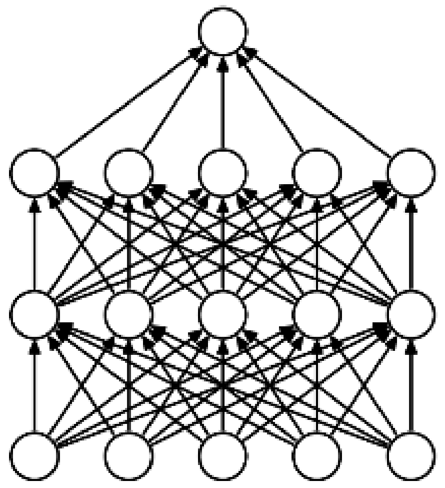


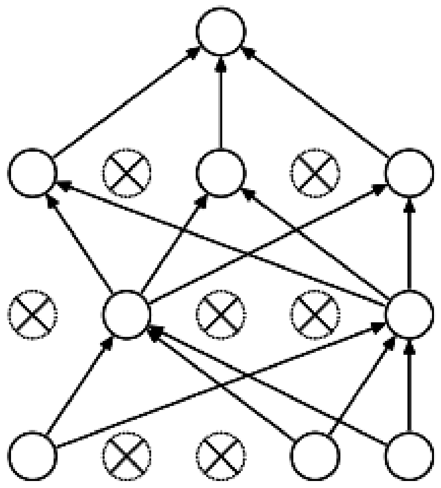
Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images [Nguyen et al. CVPR 2015]

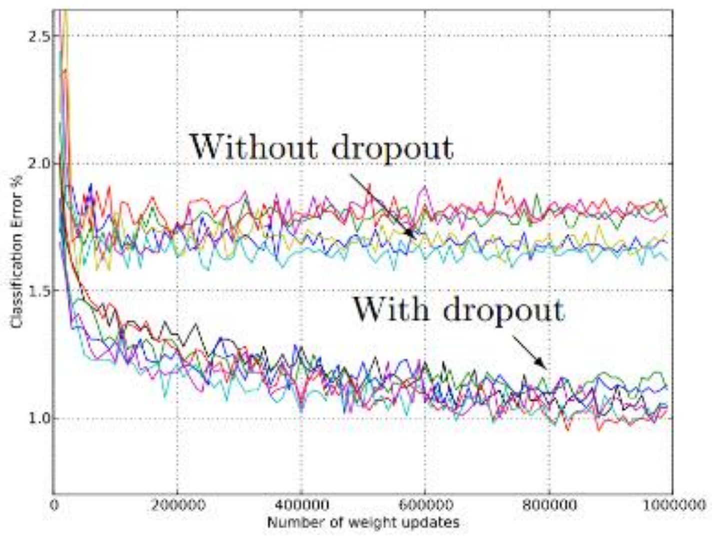
Training Convolutional Neural Networks

- Backpropagation + stochastic gradient descent with momentum
  - Neural Networks: Tricks of the Trade
- Dropout
- Data augmentation
- Batch normalization
- Initialization
  - Transfer learning

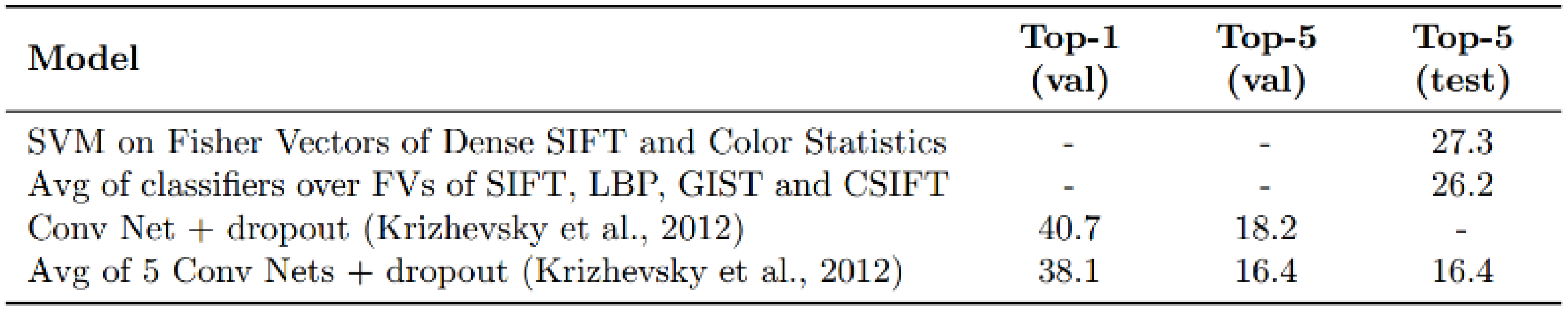
Dropout


(a) Standard Neural Net


(b) After applying dropout.



Main Idea: approximately combining exponentially many different neural network architectures efficiently



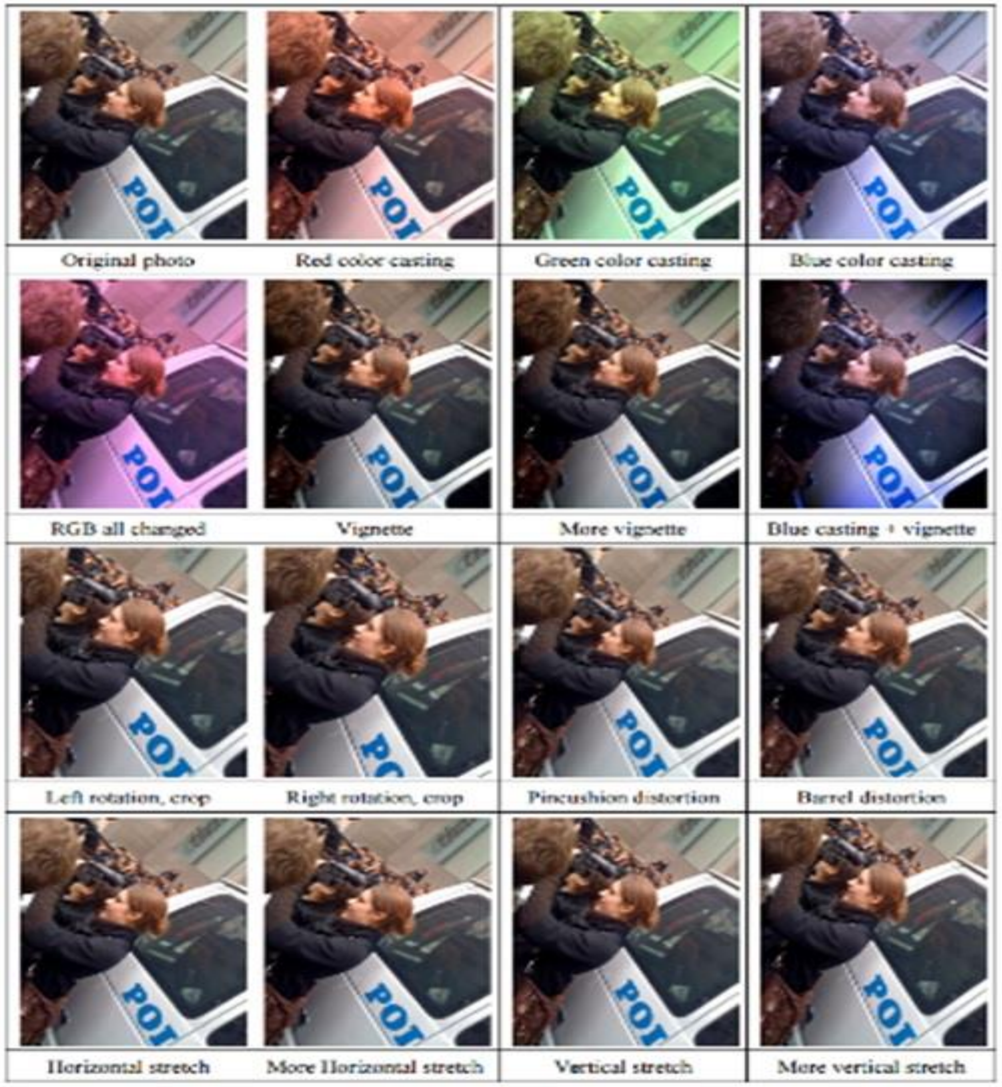
Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SVM on Fisher Vectors of Dense SIFT and Color Statistics	-	-	27.3
Avg of classifiers over FVs of SIFT, LBP, GIST and CSIFT	-	-	26.2
Conv Net + dropout (Krizhevsky et al., 2012)	40.7	18.2	-
Avg of 5 Conv Nets + dropout (Krizhevsky et al., 2012)	38.1	16.4	16.4

Table 6: Results on the ILSVRC-2012 validation/test set.

Dropout: A simple way to prevent neural networks from overfitting [Srivastava JMLR 2014]

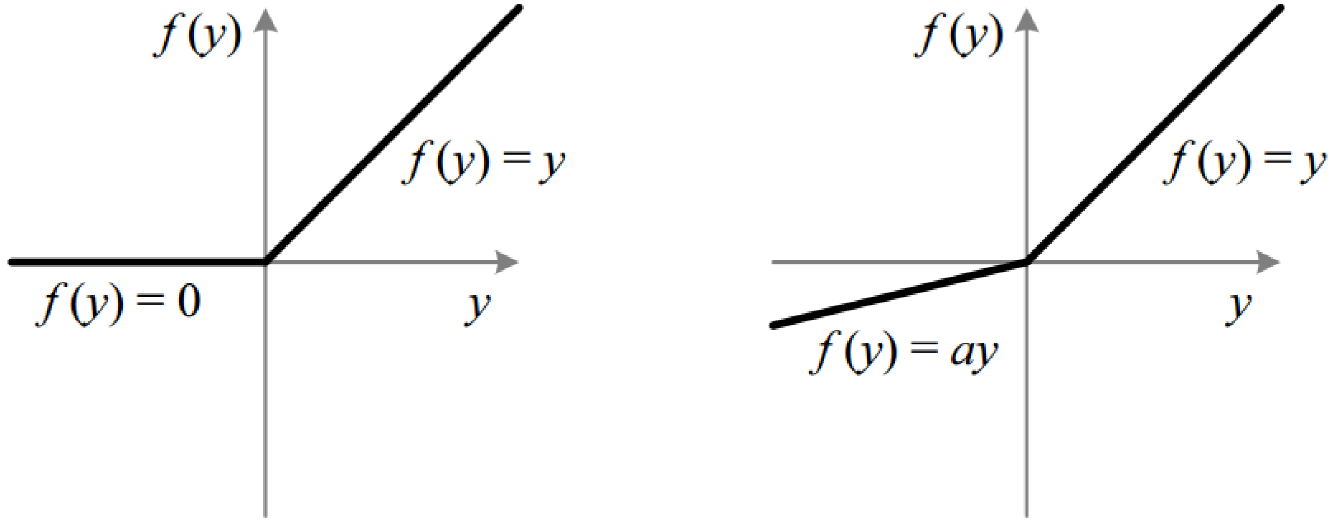
Data Augmentation (Jittering)

- Create *virtual* training samples
  - Horizontal flip
  - Random crop
  - Color casting
  - Geometric distortion



Deep Image [Wu et al. 2015]

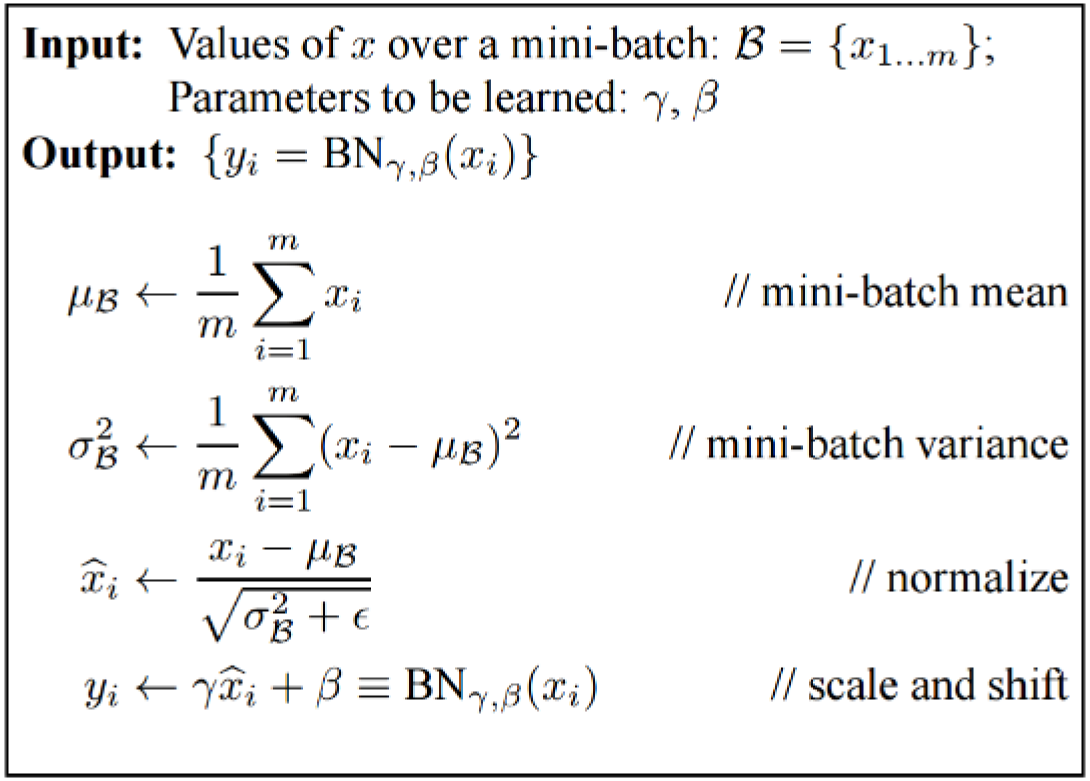
Parametric Rectified Linear Unit



	team	top-5 (**test**)
in competition ILSVRC 14	MSRA, SPP-nets [11]	8.06
	VGG [25]	7.32
	GoogLeNet [29]	6.66
post-competition	VGG [25] (arXiv v5)	6.8
	Baidu [32]	5.98
	MSRA, PReLU-nets	**4.94**

Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification [He et al. 2015]

Batch Normalization



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\dots m}\}$;
Parameters to be learned: γ, β

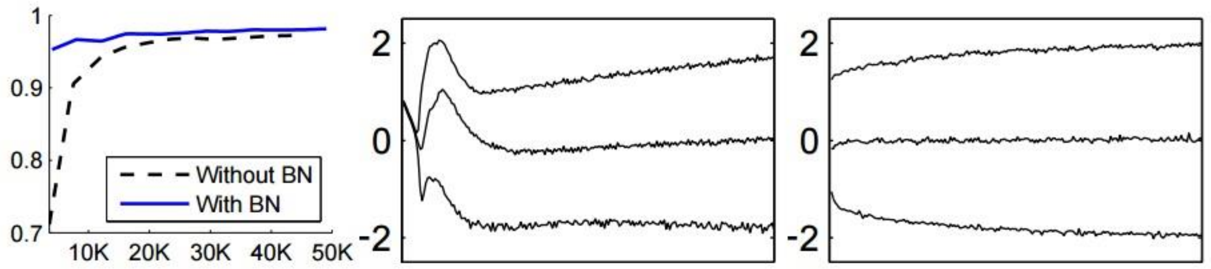
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

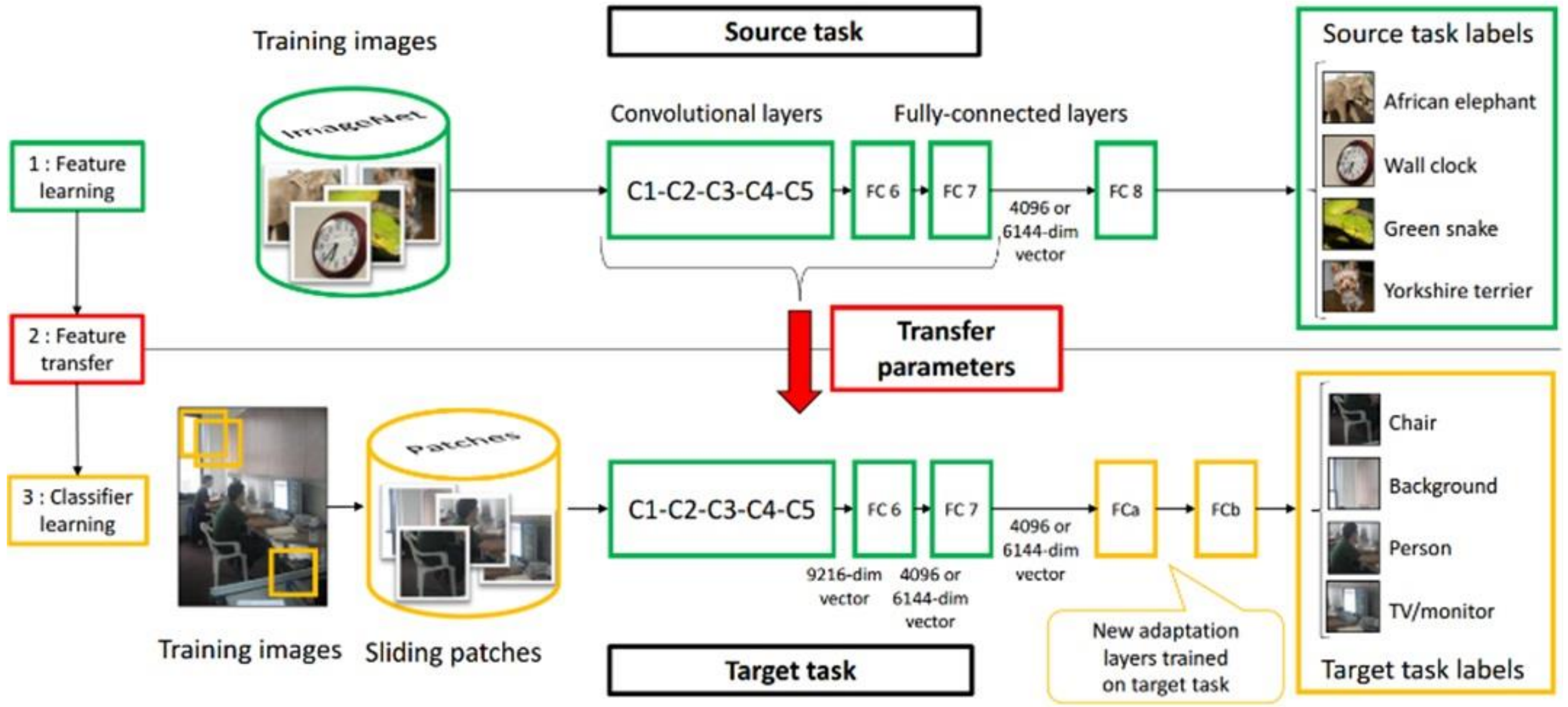


(a) (b) Without BN (c) With BN

Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [Ioffe and Szegedy 2015]

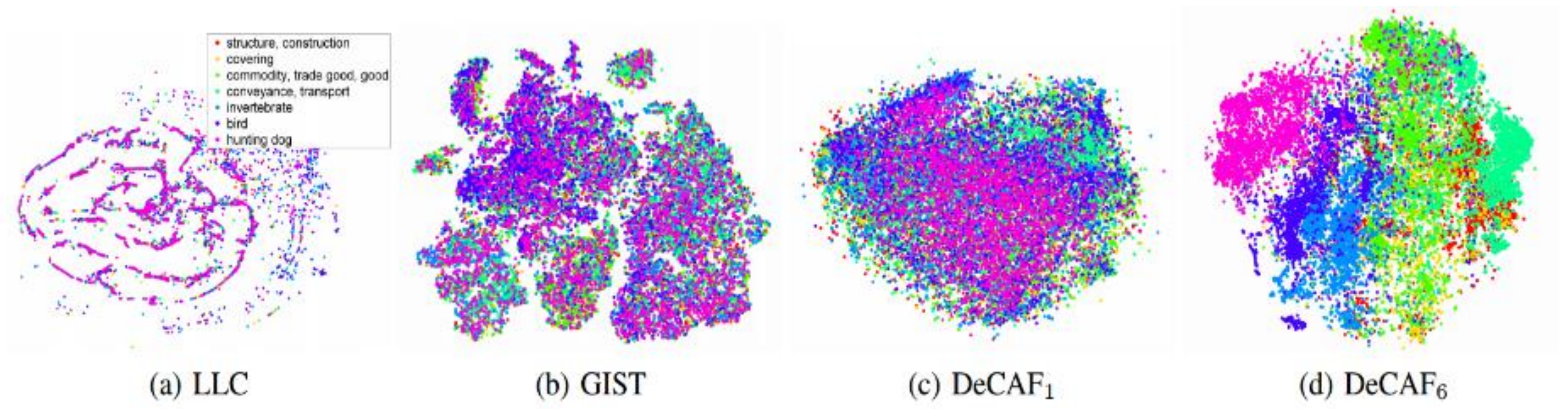
Transfer Learning

- Improvement of learning in a **new** task through the *transfer of knowledge* from a **related** task that has already been learned.
- Weight initialization for CNN



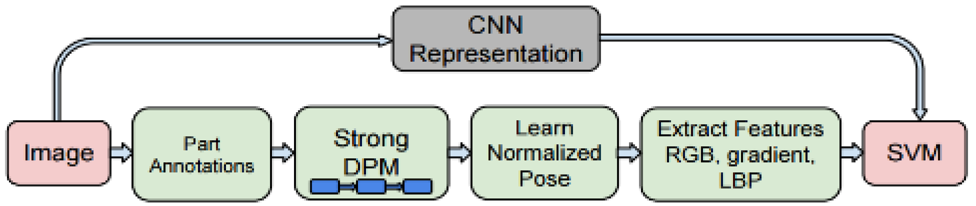
Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks [Oquab et al. CVPR 2014]

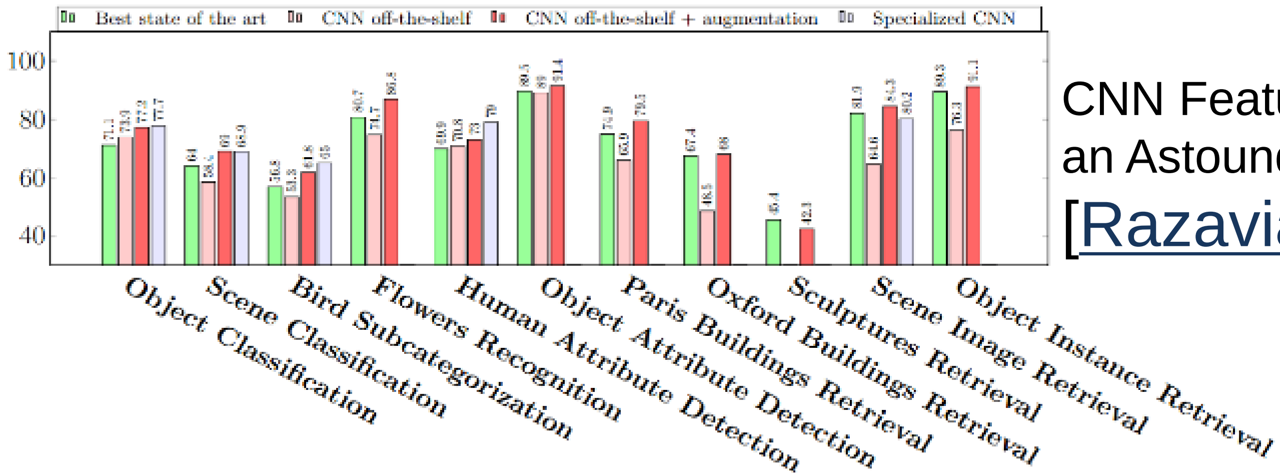
Convolutional activation features



(a) LLC (b) GIST (c) DeCAF_1 (d) DeCAF_6

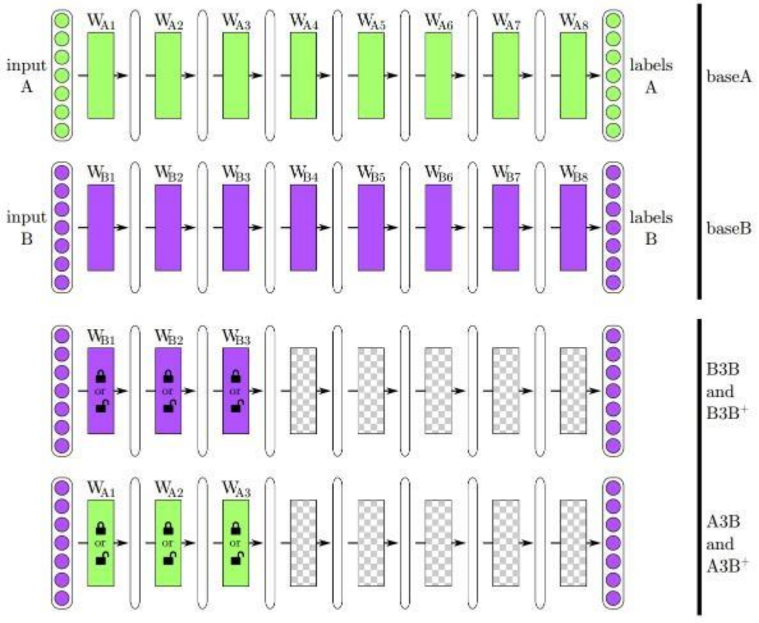
[Donahue et al. ICML 2013]

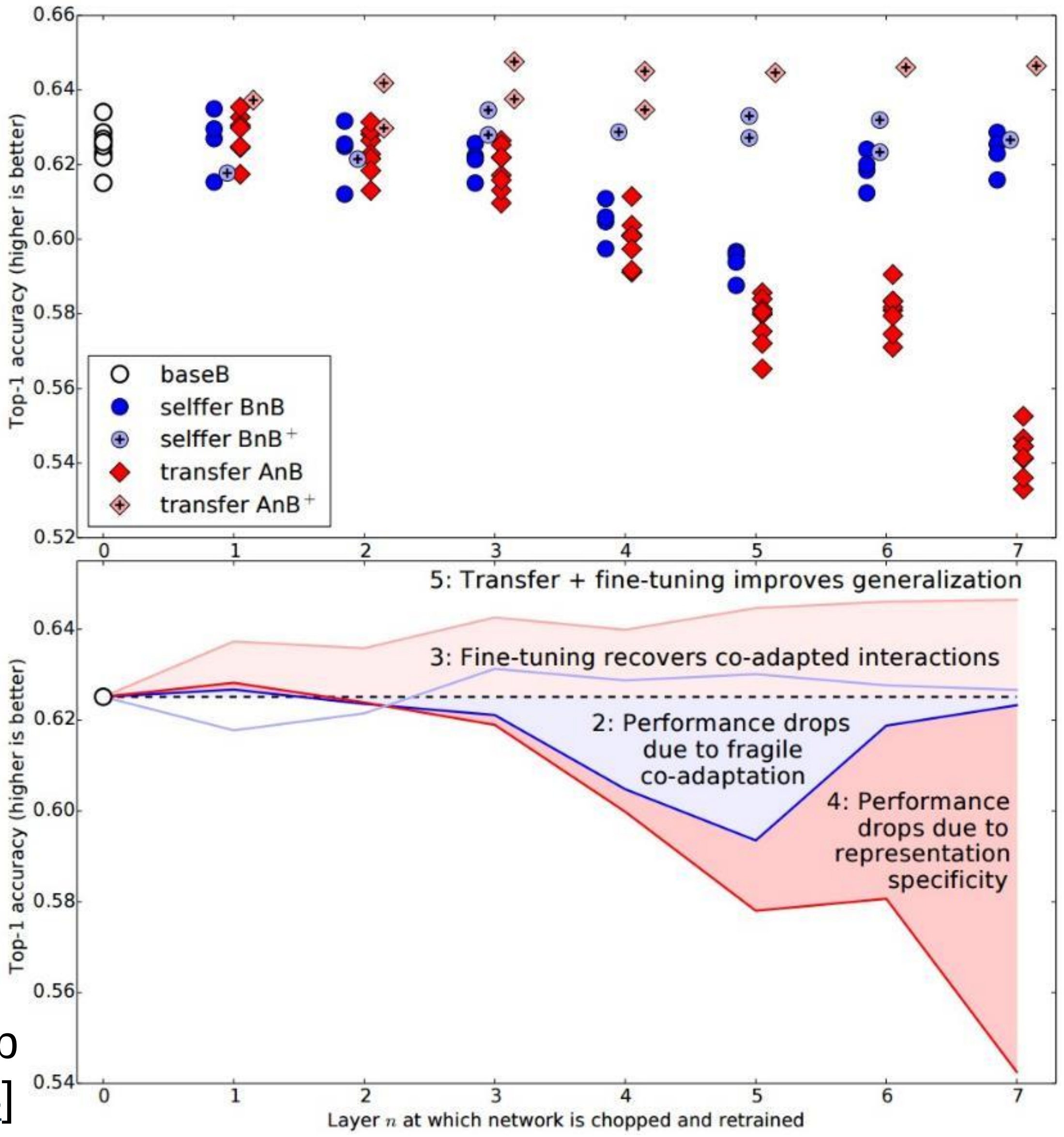




CNN Features off-the-shelf:
an Astounding Baseline for Recognition
[Razavian et al. 2014]

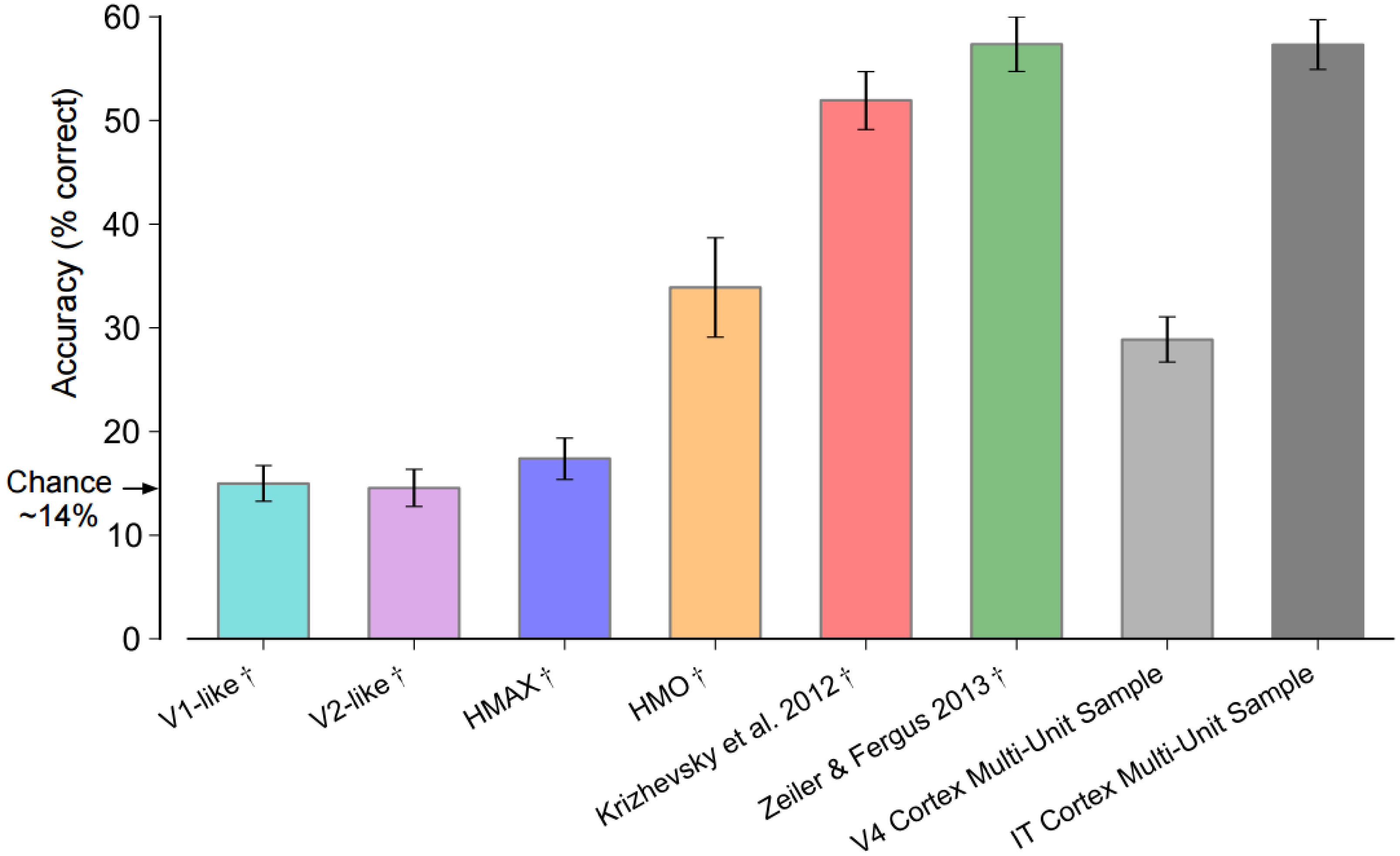
How transferable are features in CNN?





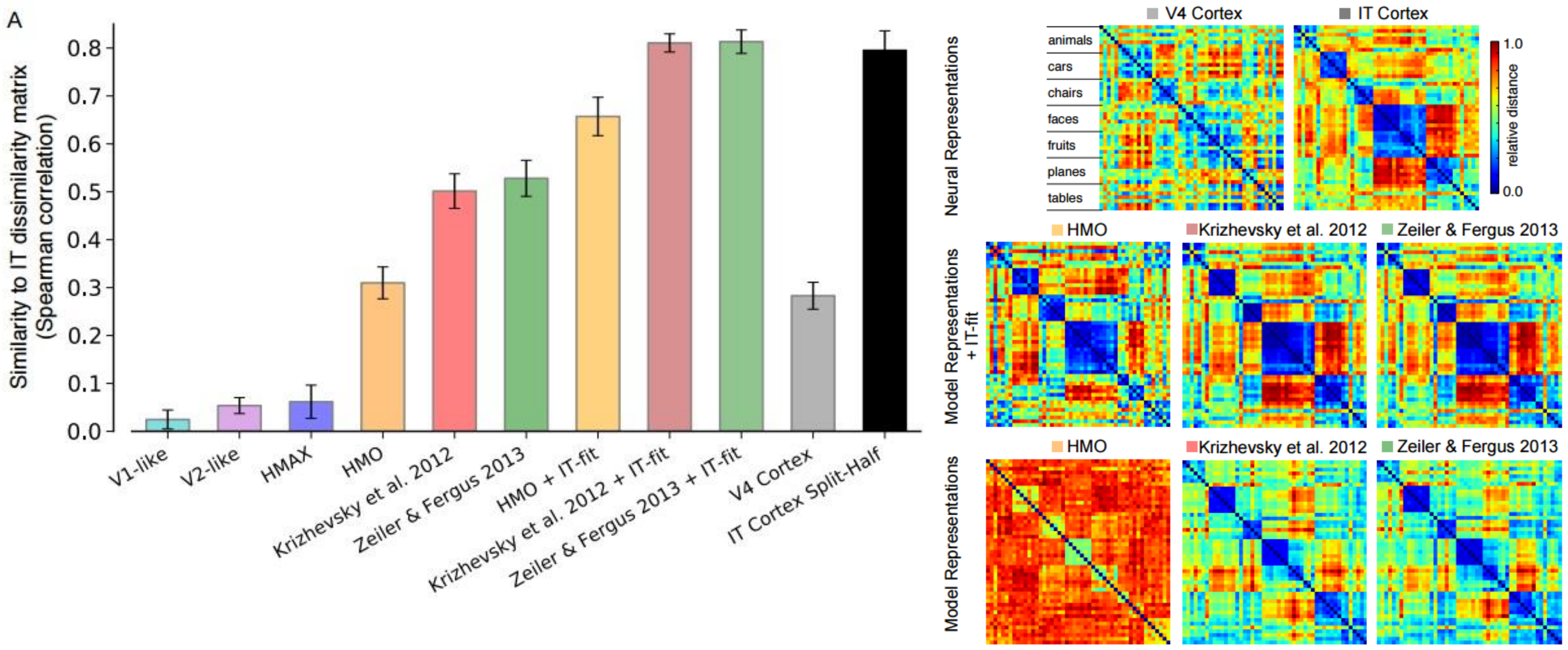
How transferable are features in deep neural networks [Yosinski NIPS 2014]

Deep Neural Networks Rival the Representation of Primate Inferior Temporal Cortex



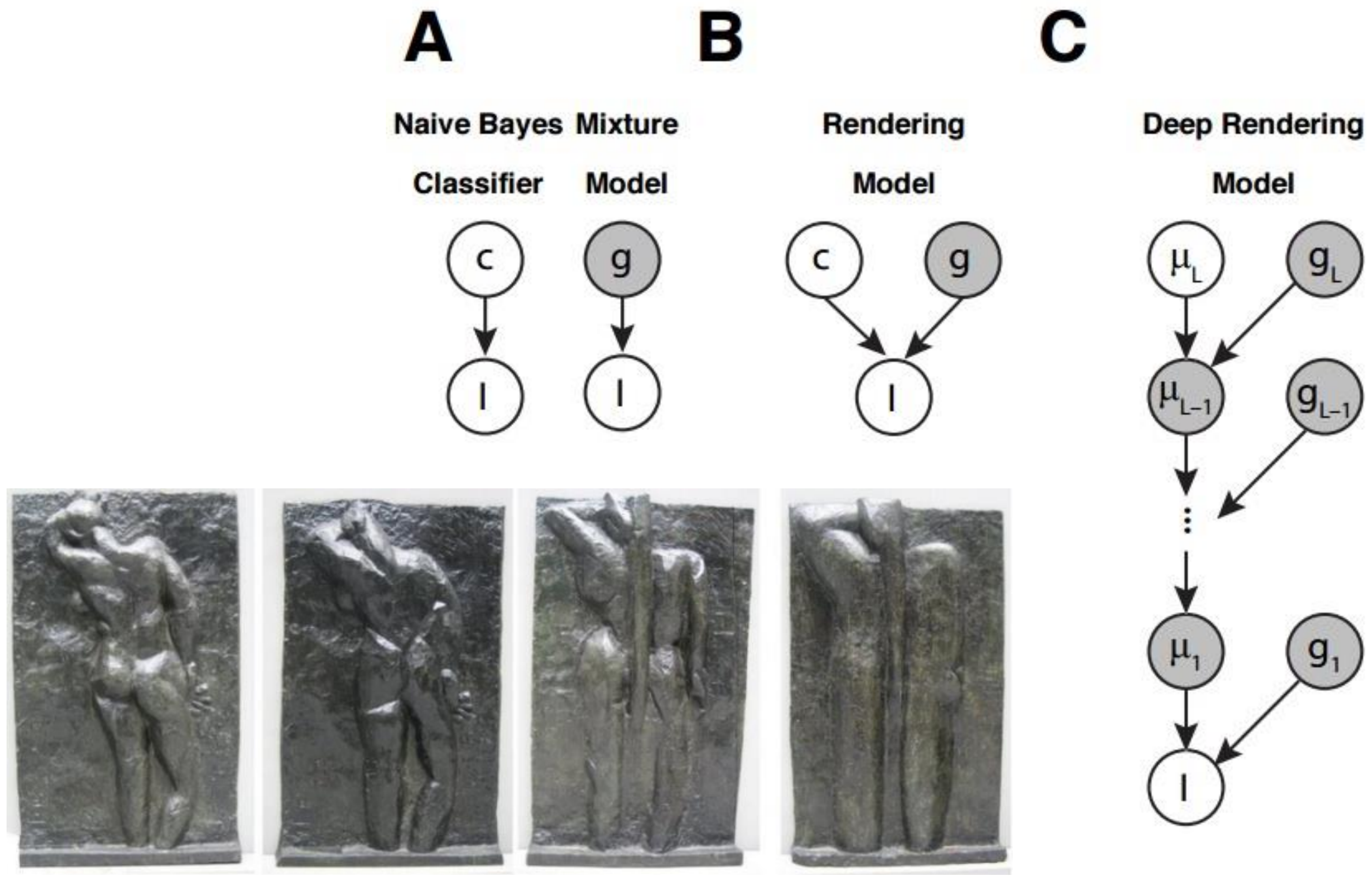
Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition [Cadieu et al. PLOS 2014]

Deep Neural Networks Rival the Representation of Primate Inferior Temporal Cortex



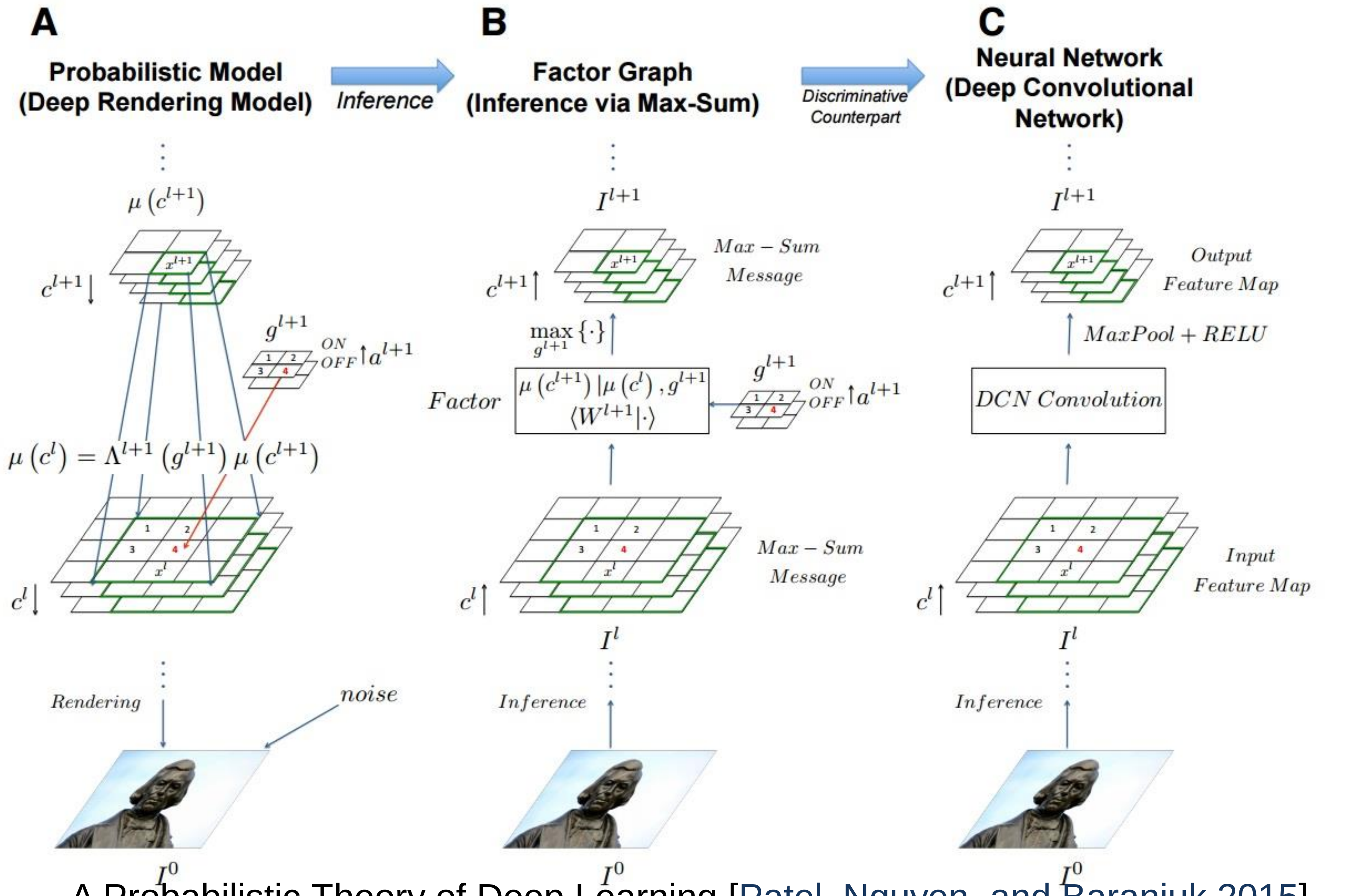
Deep Neural Networks Rival the Representation of Primate IT Cortex for Core Visual Object Recognition [Cadieu et al. PLOS 2014]

Deep Rendering Model (DRM)



A Probabilistic Theory of Deep Learning [Patel, Nguyen, and Baraniuk 2015]

CNN as a Max-Sum Inference

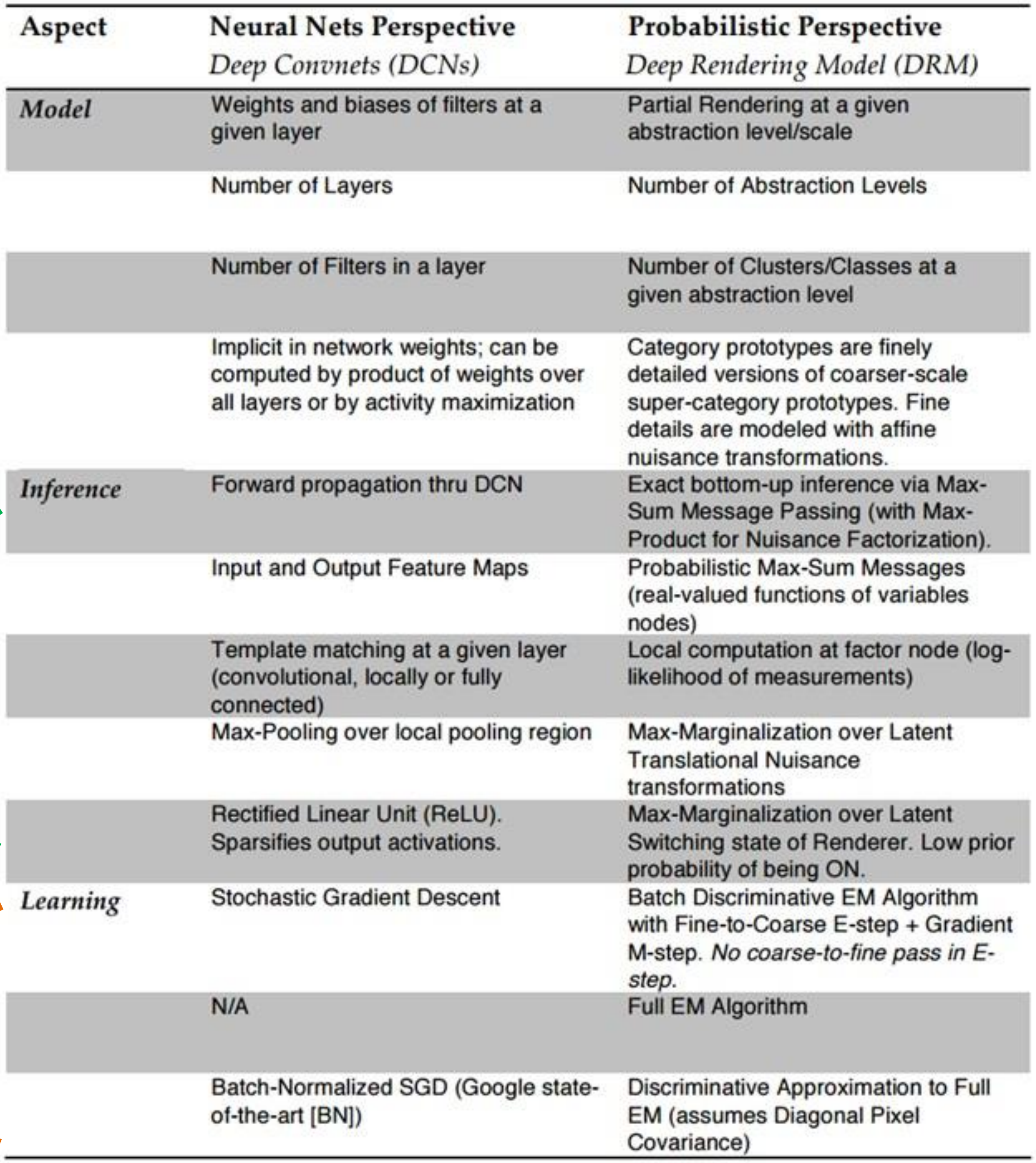


A Probabilistic Theory of Deep Learning [Patel, Nguyen, and Baraniuk 2015]

Model

Inference

Learning



Aspect	**Neural Nets Perspective** *Deep Convnets (DCNs)*	**Probabilistic Perspective** *Deep Rendering Model (DRM)*
Model	Weights and biases of filters at a given layer	Partial Rendering at a given abstraction level/scale
	Number of Layers	Number of Abstraction Levels
	Number of Filters in a layer	Number of Clusters/Classes at a given abstraction level
	Implicit in network weights; can be computed by product of weights over all layers or by activity maximization	Category prototypes are finely detailed versions of coarser-scale super-category prototypes. Fine details are modeled with affine nuisance transformations.
Inference	Forward propagation thru DCN	Exact bottom-up inference via Max-Sum Message Passing (with Max-Product for Nuisance Factorization).
	Input and Output Feature Maps	Probabilistic Max-Sum Messages (real-valued functions of variables nodes)
	Template matching at a given layer (convolutional, locally or fully connected)	Local computation at factor node (log-likelihood of measurements)
	Max-Pooling over local pooling region	Max-Marginalization over Latent Translational Nuisance transformations
	Rectified Linear Unit (ReLU). Sparsifies output activations.	Max-Marginalization over Latent Switching state of Renderer. Low prior probability of being ON.
Learning	Stochastic Gradient Descent	Batch Discriminative EM Algorithm with Fine-to-Coarse E-step + Gradient M-step. *No coarse-to-fine pass in E-step.*
	N/A	Full EM Algorithm
	Batch-Normalized SGD (Google state-of-the-art [BN])	Discriminative Approximation to Full EM (assumes Diagonal Pixel Covariance)

Tools

- Caffe
- cuda-convnet2
- Torch
- MatConvNet
- Pylearn2

Resources

- http://deeplearning.net/
- https://github.com/ChristosChristofidis/awesome-deep-learning

Things to remember

- Overview
  - Neuroscience, Perceptron, multi-layer neural networks
- Convolutional neural network (CNN)
  - Convolution, nonlinearity, max pooling
  - CNN for classification and beyond
- Understanding and visualizing CNN
  - Find images that maximize some class scores; visualize individual neuron activation, input pattern and images; breaking CNNs
- Training CNN
  - Dropout; data augmentation; batch normalization; transfer learning
- Probabilistic interpretation
  - Deep rendering model; CNN forward-propagation as max-sum inference; training as an EM algorithm