Convolutional Neural Networs for Image Classification

Marcello Pelillo

University of Venice, Italy

Image and Video Understanding

a.y. 2018/19

The Age of "Deep Learning"

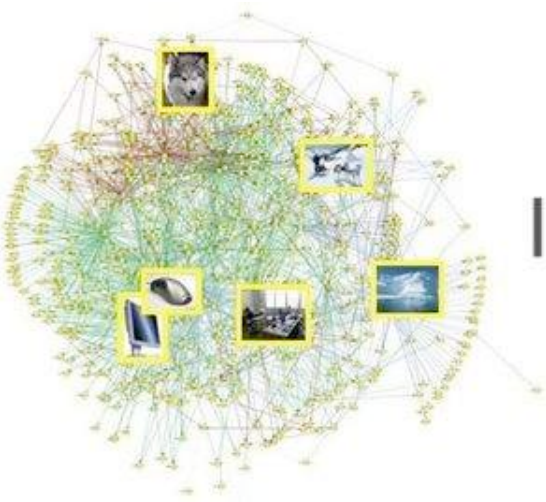
News & Analysis

Microsoft, Google Beat Humans at Image Recognition

Deep learning algorithms compete at ImageNet challenge

R. Colin Johnson
2/18/2015 08:15 AM EST
14 comments

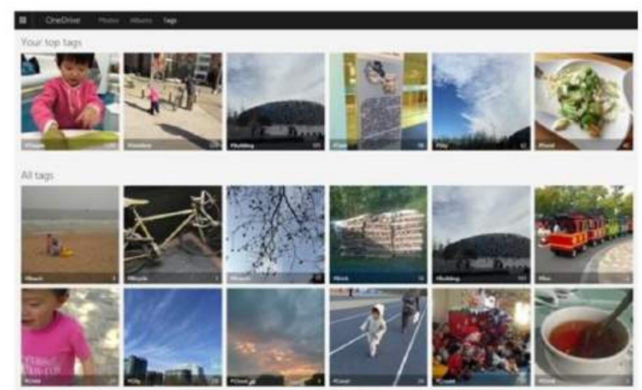
NO RATINGS
1 saves
LOGIN TO RATE



IMAGENET

PORTLAND, Ore. -- First computers beat the best of us at chess, then poker, and finally Jeopardy. The next hurdle is image recognition — surely a computer can't do that as well as a human. Check that one off the list, too. Now Microsoft has programmed the first computer to beat the humans at image recognition.

The competition is fierce, with the ImageNet Large Scale Visual Recognition Challenge doing the judging for the 2015 championship on December 17. Between now and then expect to see a stream of papers claiming they have one-upped humans too. For instance, only 5 days after Microsoft announced it had beat the human benchmark of 5.1% errors with a 4.94% error grabbing neural network, Google announced it had one-upped Microsoft by 0.04%.

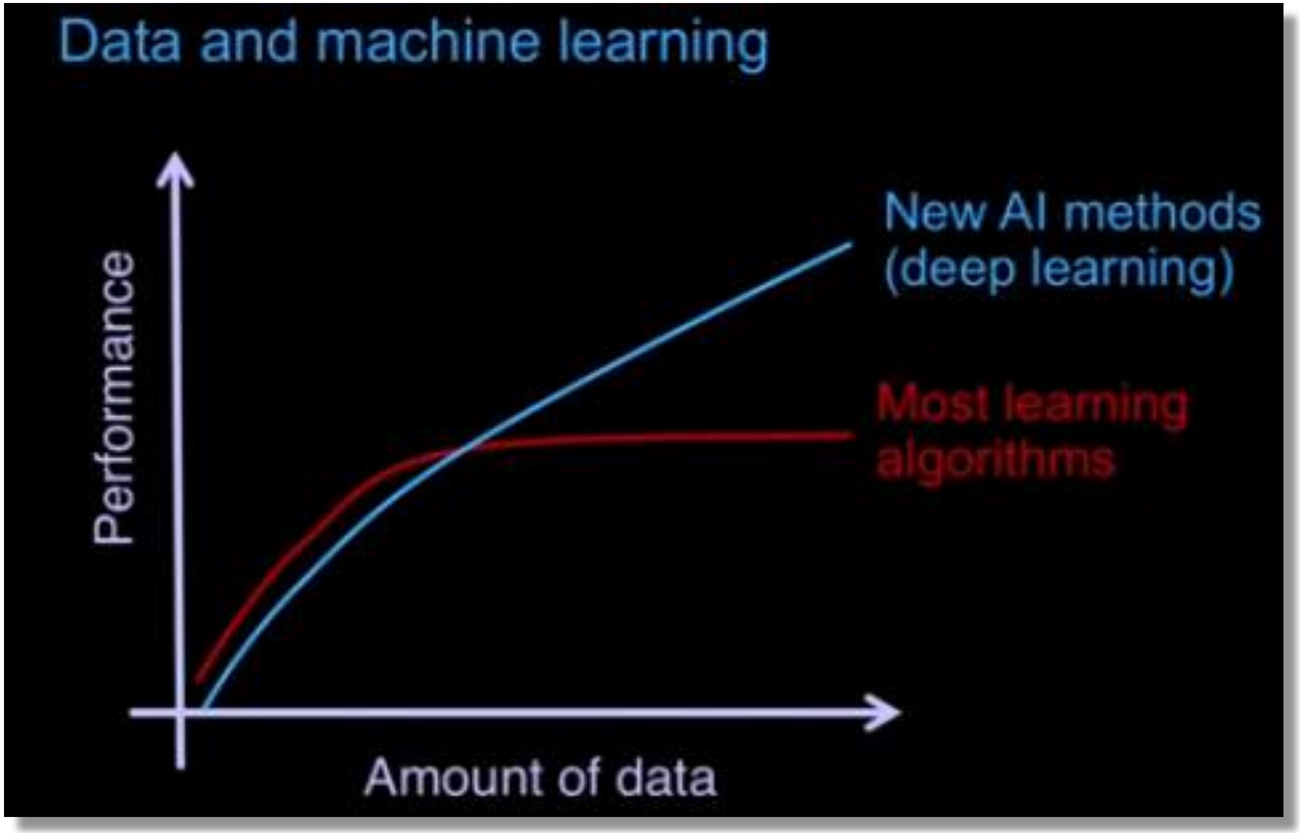


The top row is a representative of the categories that Microsoft's algorithm found in the database and the image columns below are examples that fit. (Source: Microsoft)

The Deep Learning "Philosophy"

- Learn a feature hierarchy all the way from pixels to classifier
- Each layer extracts features from the output of previous layer
- Train all layers jointly

Performance Improves with More Data



Old Idea… Why Now?

1. We have more data - from Lena to ImageNet.

2. We have more computing power, GPUs are really good at this.

3. Last but not least, we have new ideas

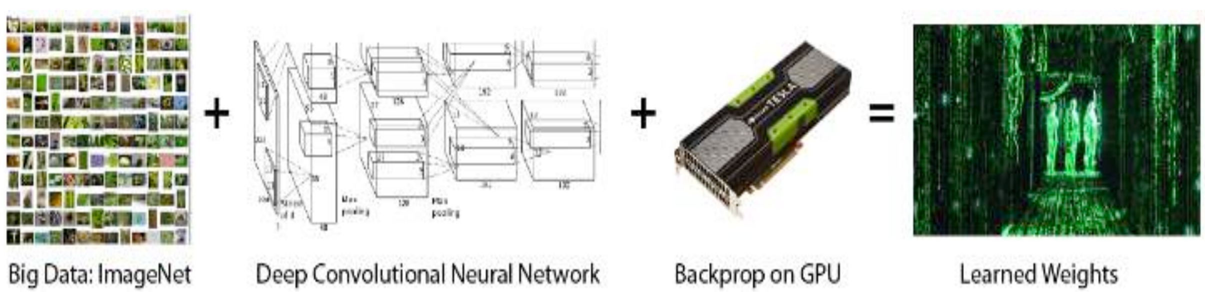
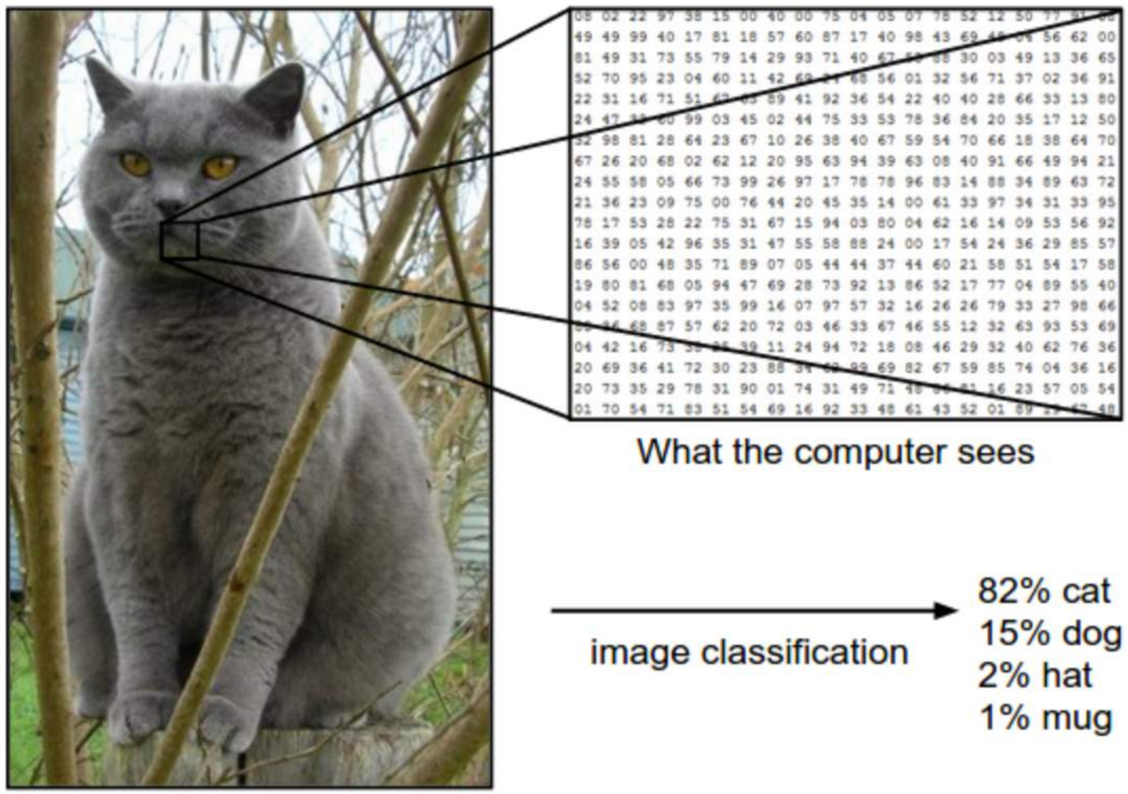


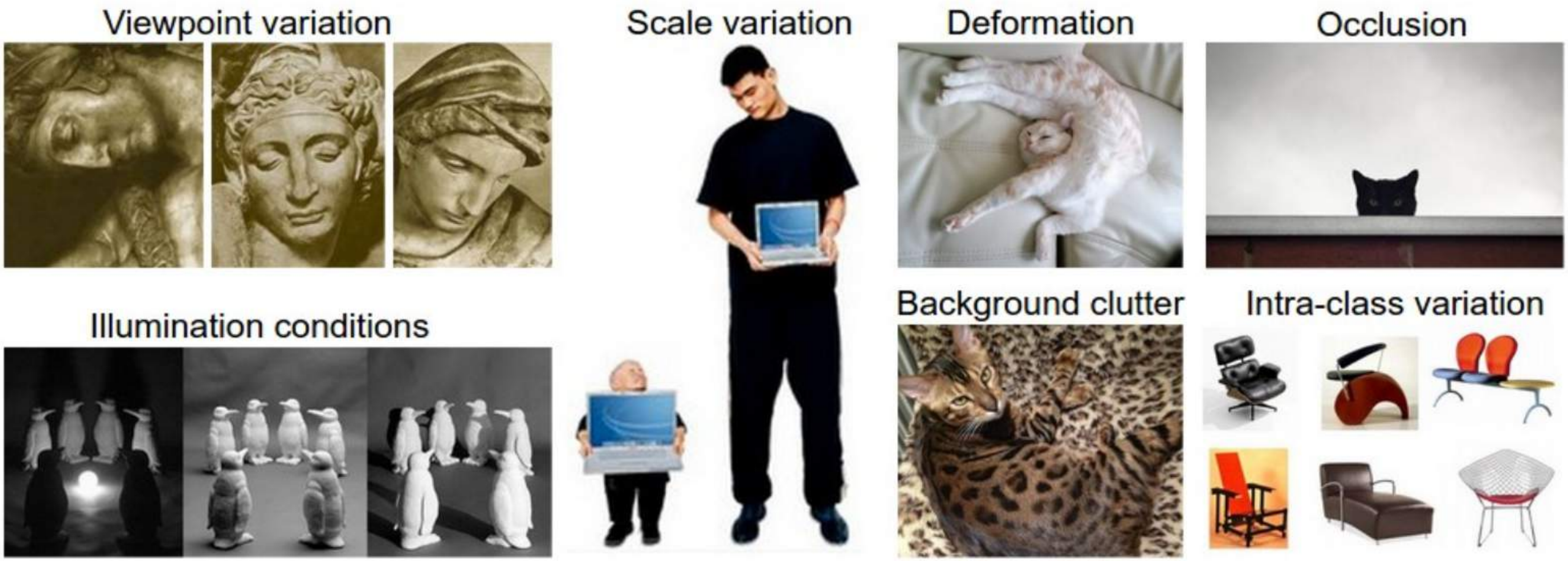
Image Classification



Predict a single label (or a distribution over labels as shown here to indicate our confidence) for a given image. Images are 3-dimensional arrays of integers from 0 to 255, of size Width x Height x 3. The 3 represents the three color channels Red, Green, Blue.

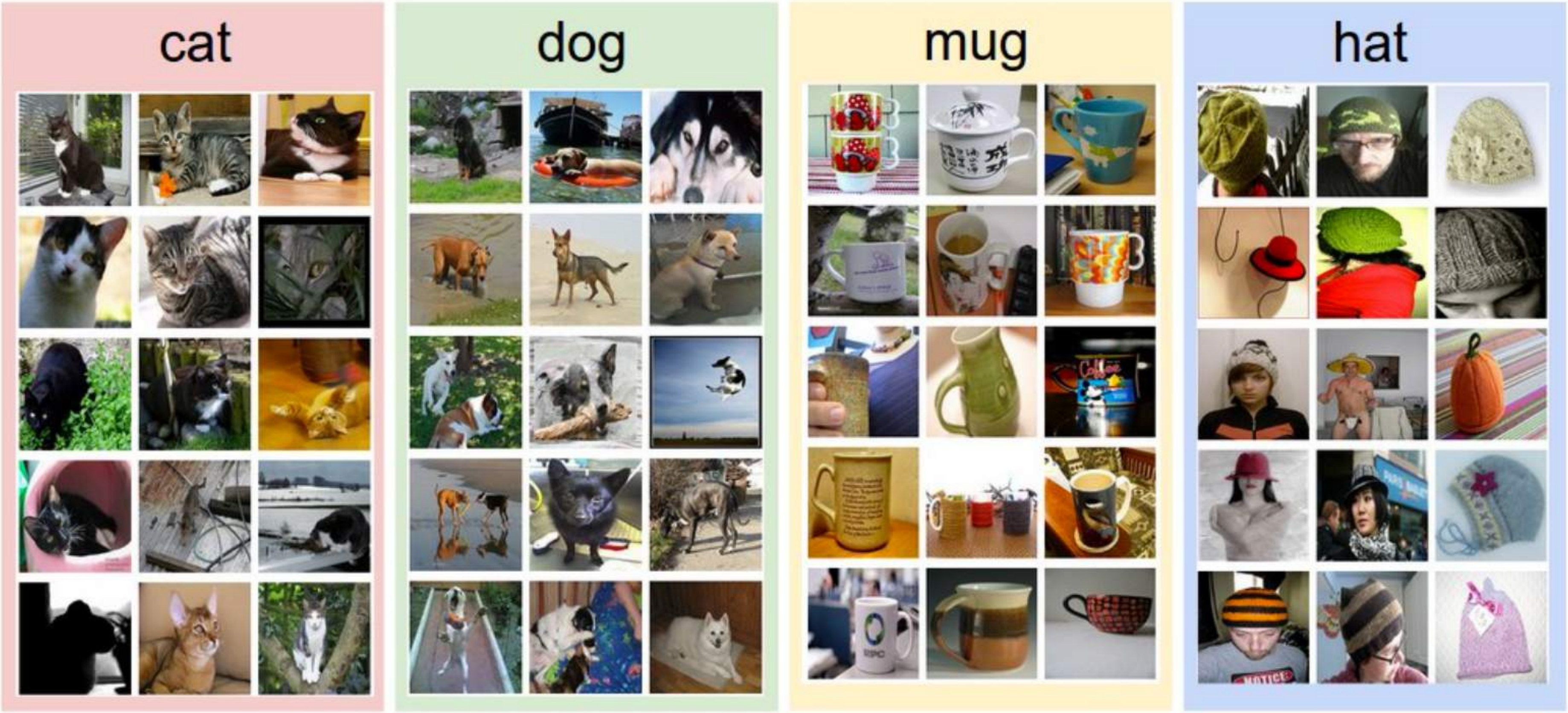
From: A. Karpathy

Challenges



From: A. Karpathy

The Data-Driven Approach

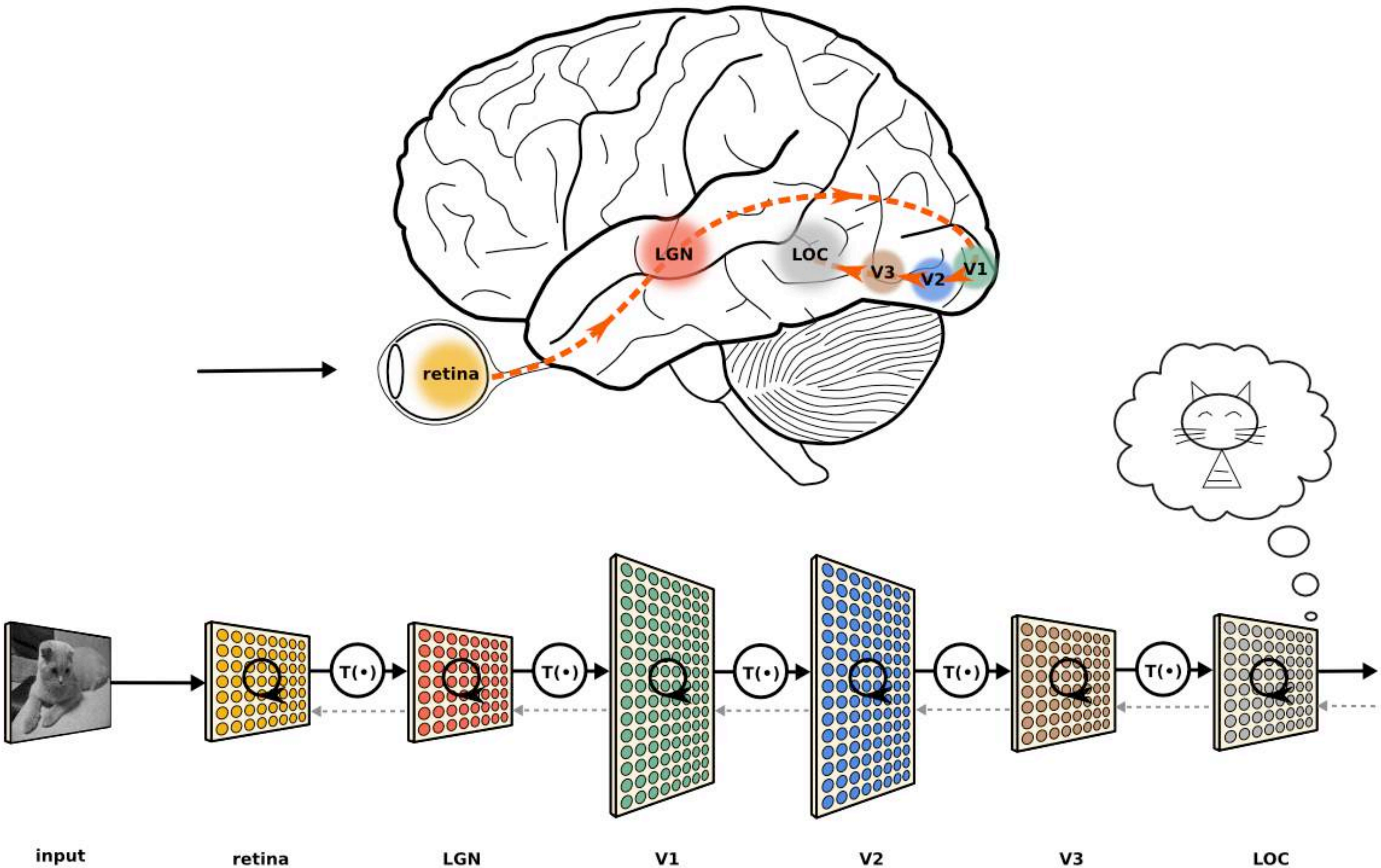


An example training set for four visual categories.

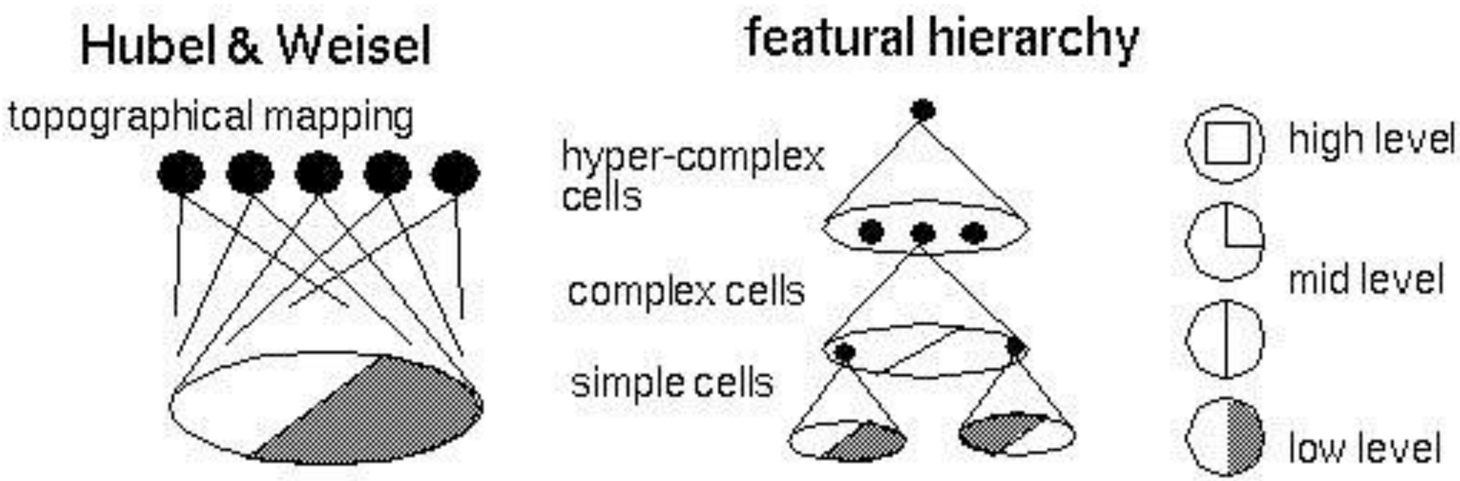
In practice we may have thousands of categories and hundreds of thousands of images for each category.

From: A. Karpathy

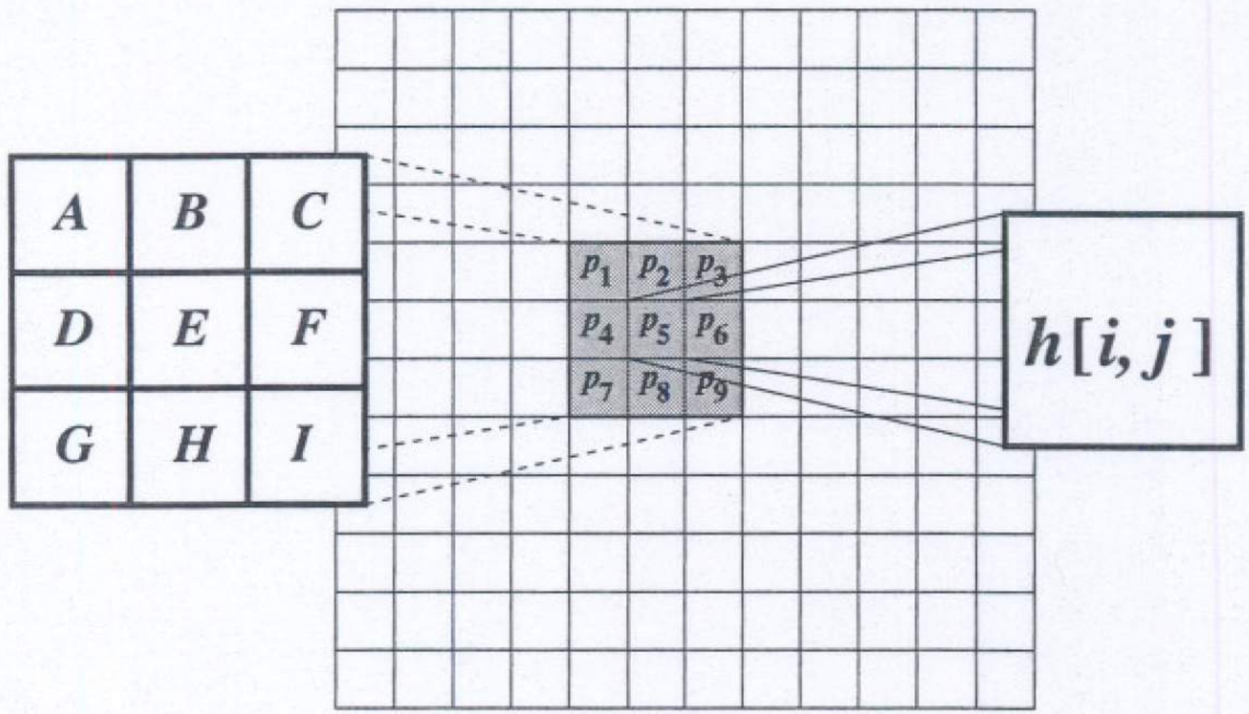
Inspiration from Biology



The Visual System as a Hierarchy of Feature Detectors

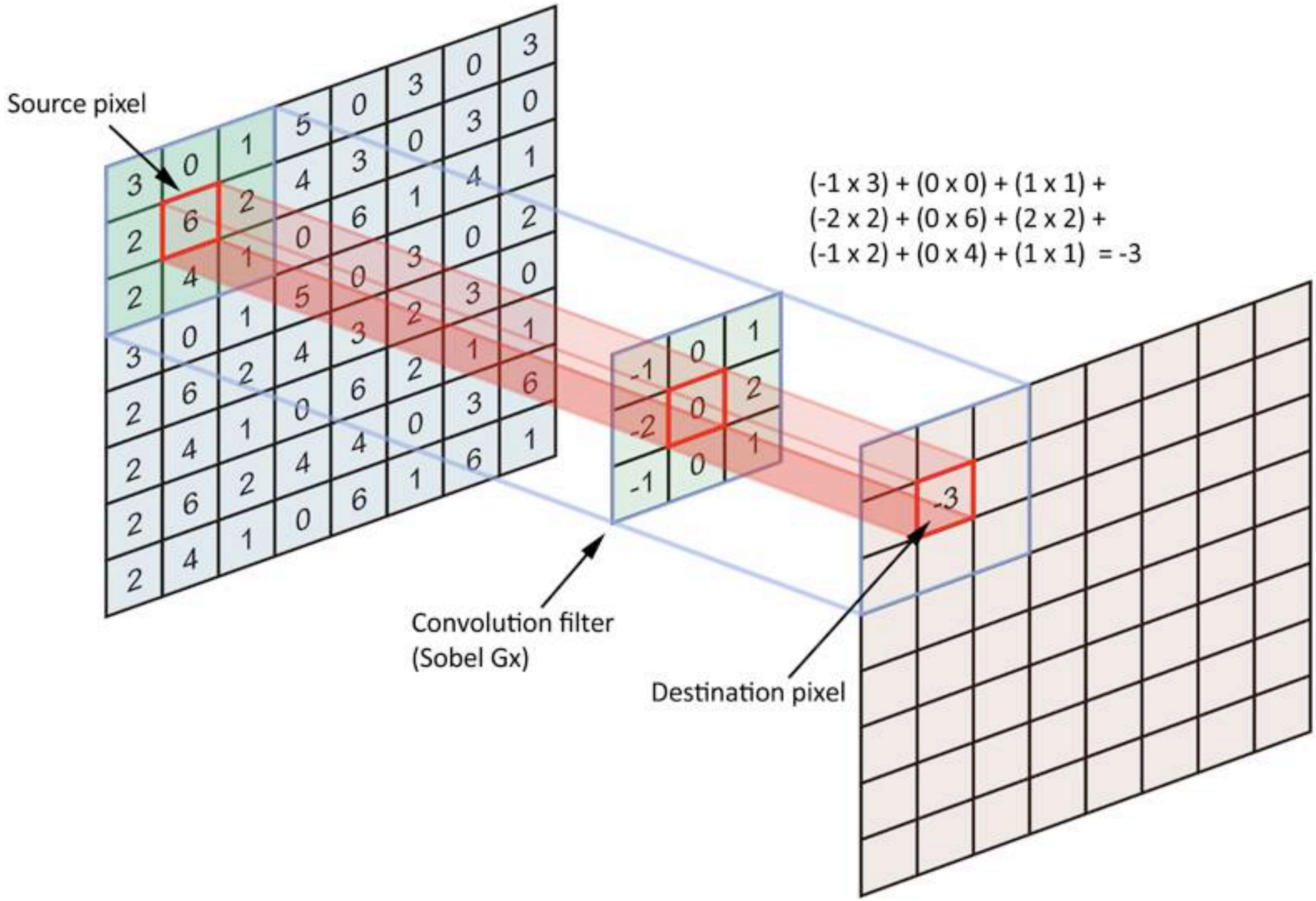


Convolution

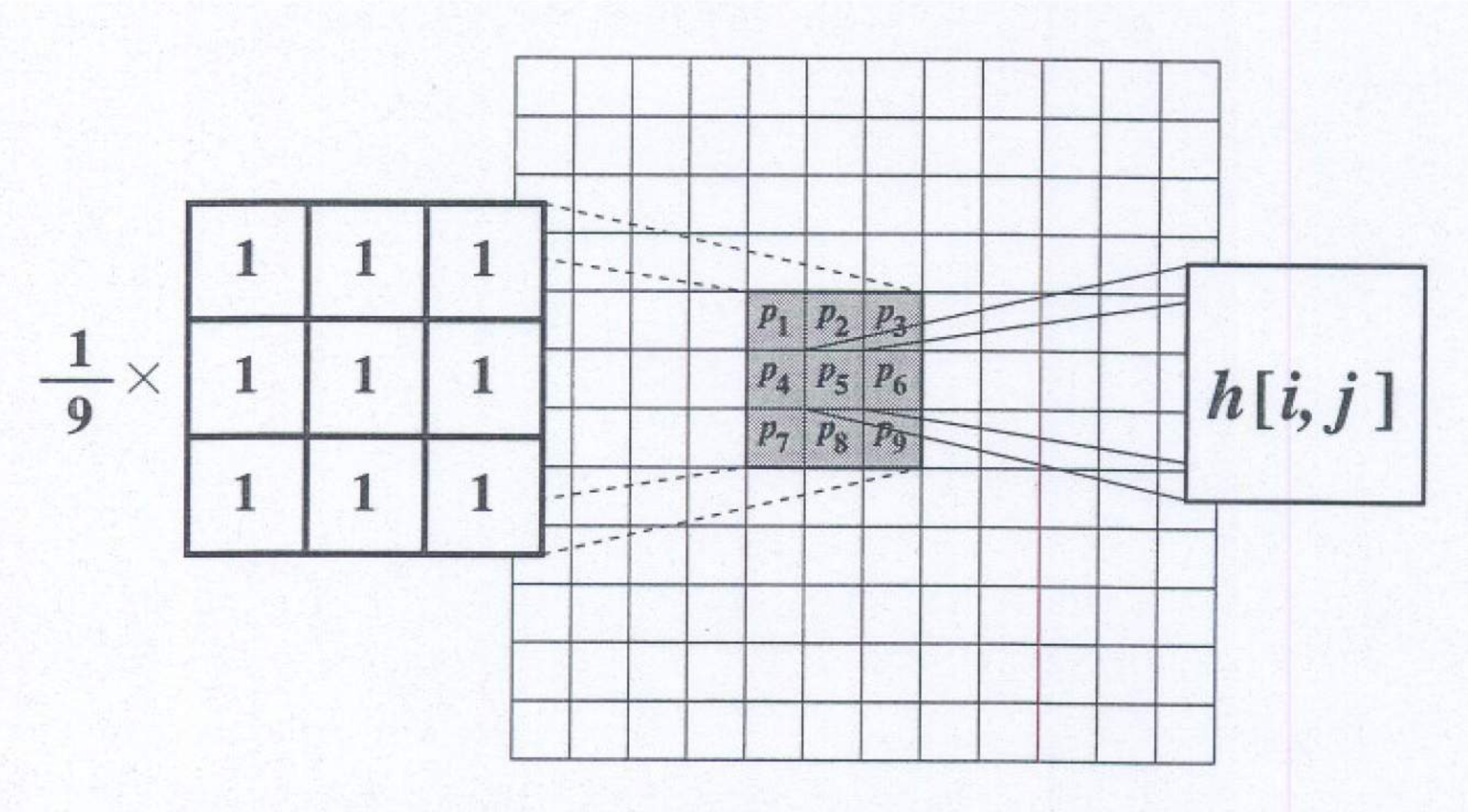


$$h[i,j] = A\,p_1 + B\,p_2 + C\,p_3 + D\,p_4 + E\,p_5 + F\,p_6 + G\,p_7 + H\,p_8 + I\,p_9$$

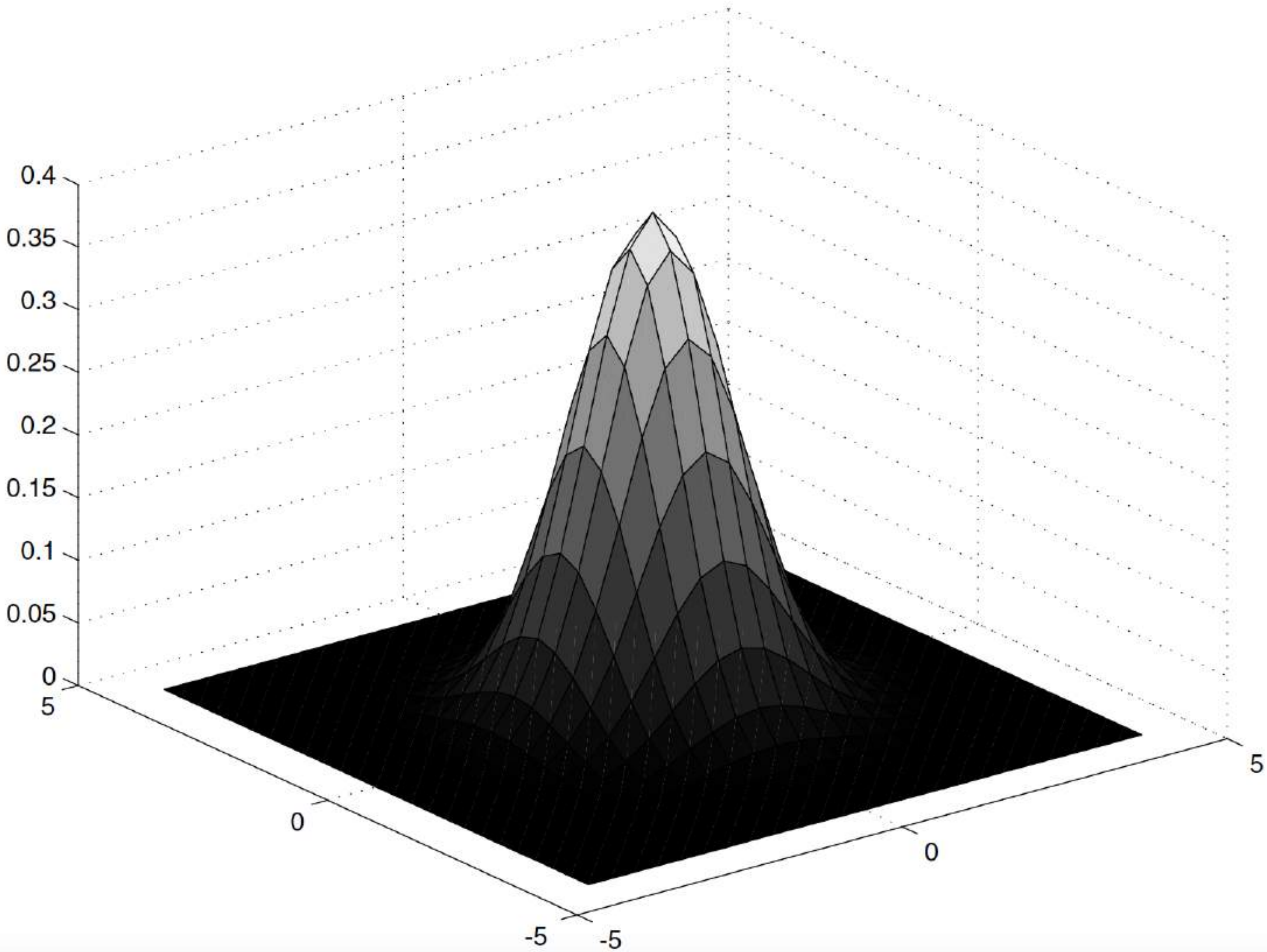
Convolution



Mean Filters



Gaussian Filters



Gaussian Filters

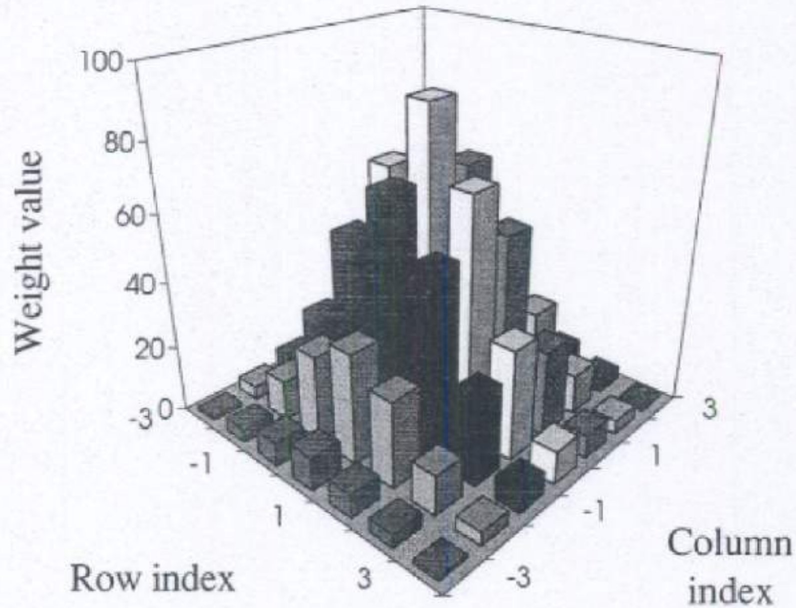
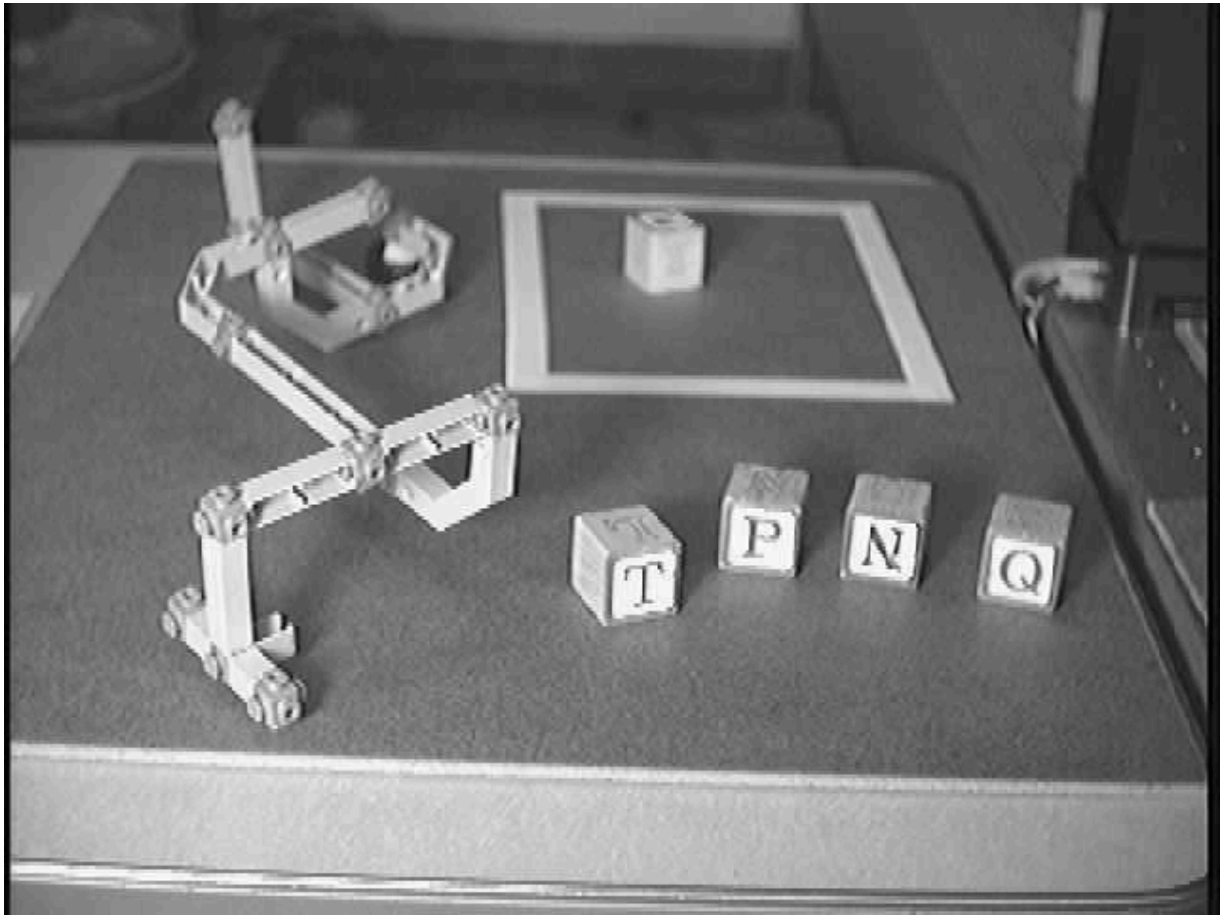


Figure 4.15: A 3-D plot of the 7×7 Gaussian mask.

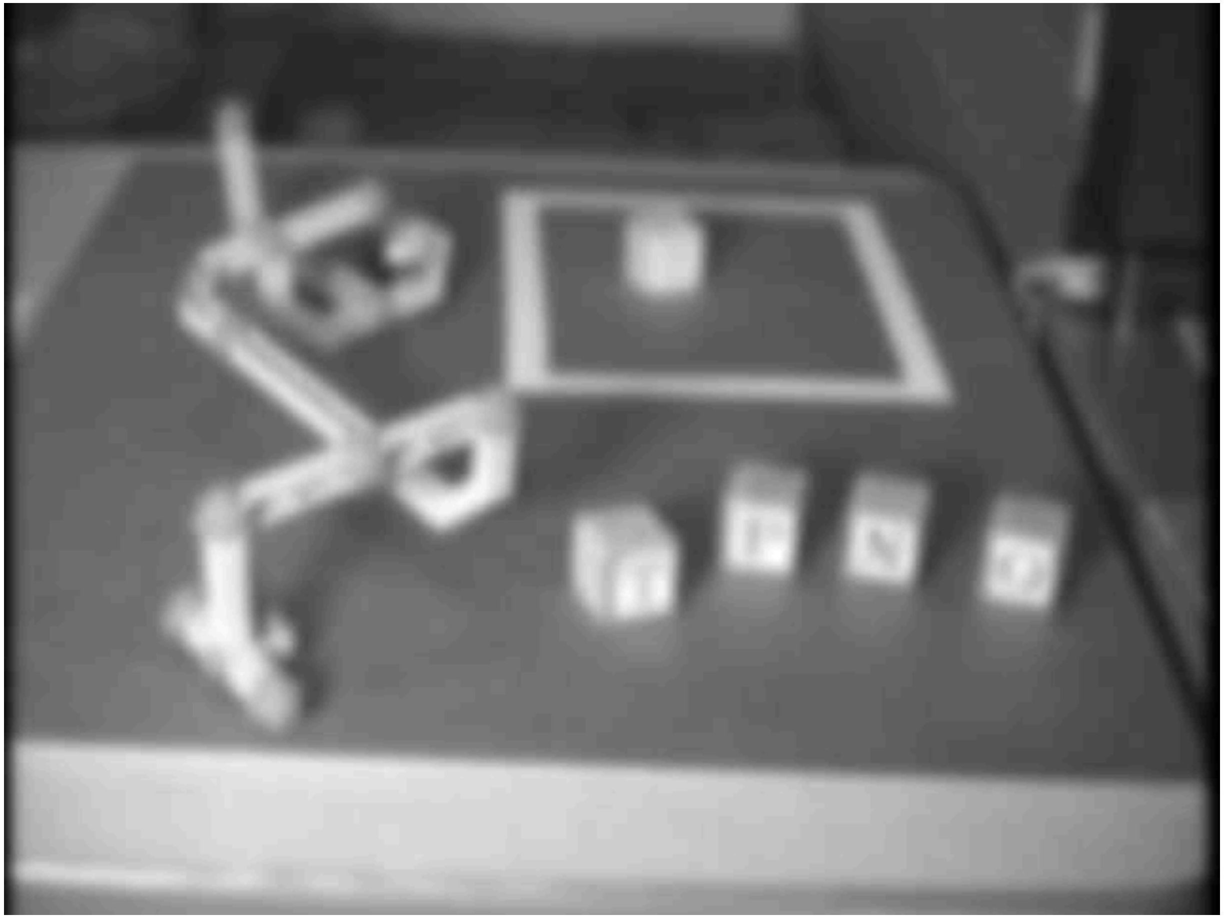
7×7 Gaussian mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

The Effect of Gaussian Filters



The Effect of Gaussian Filters



Kernel Width Affects Scale

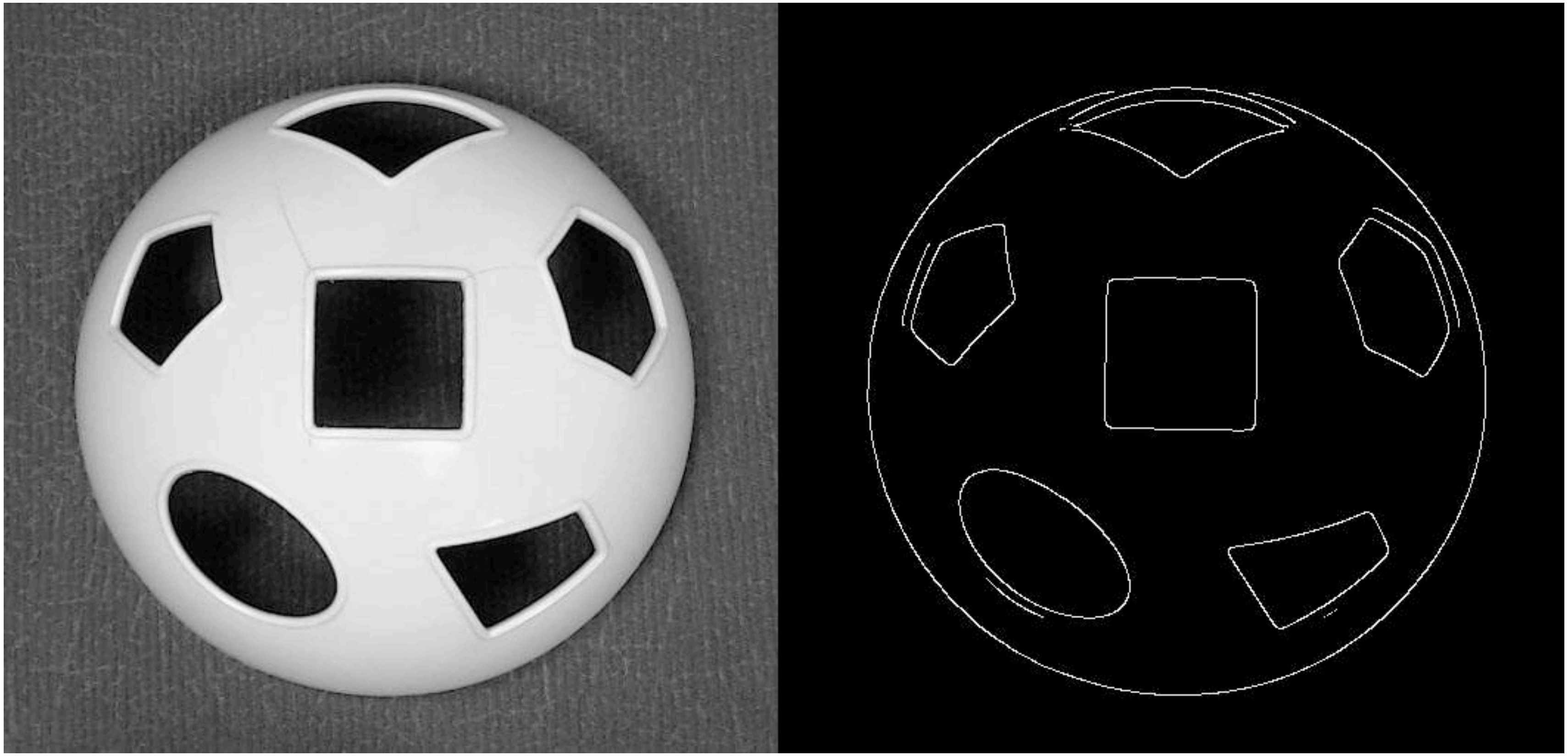
Width = 3

Width = 7

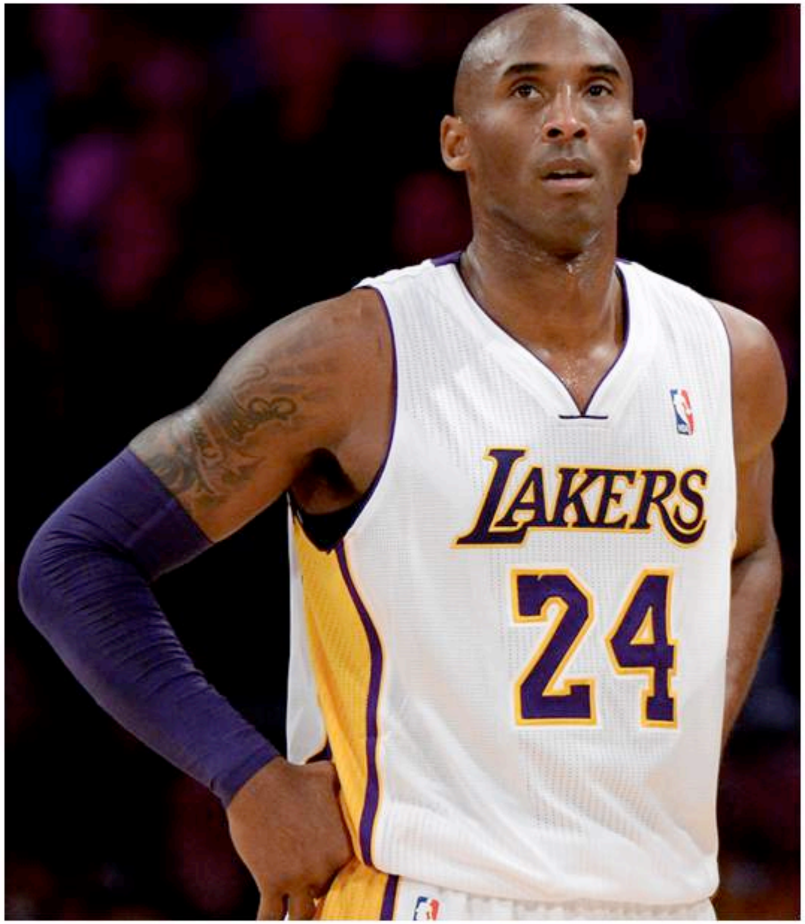
Width = 13

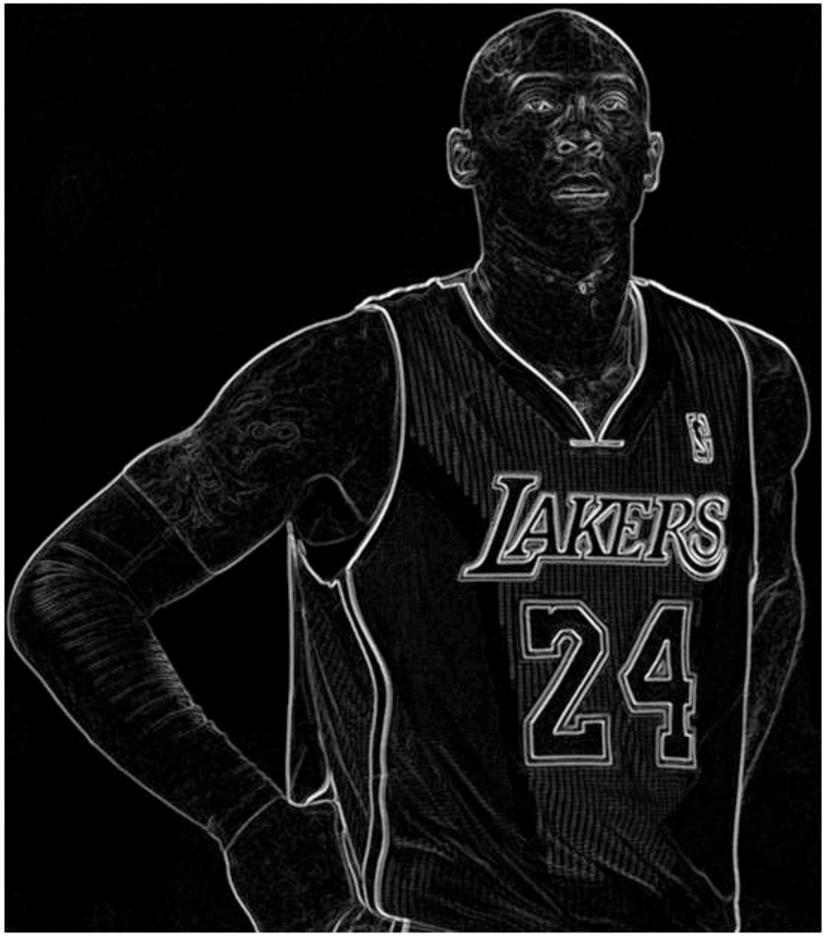
Width = 19

Edge detection



Edge detection





Using Convolution for Edge Detection

Roberts Operator

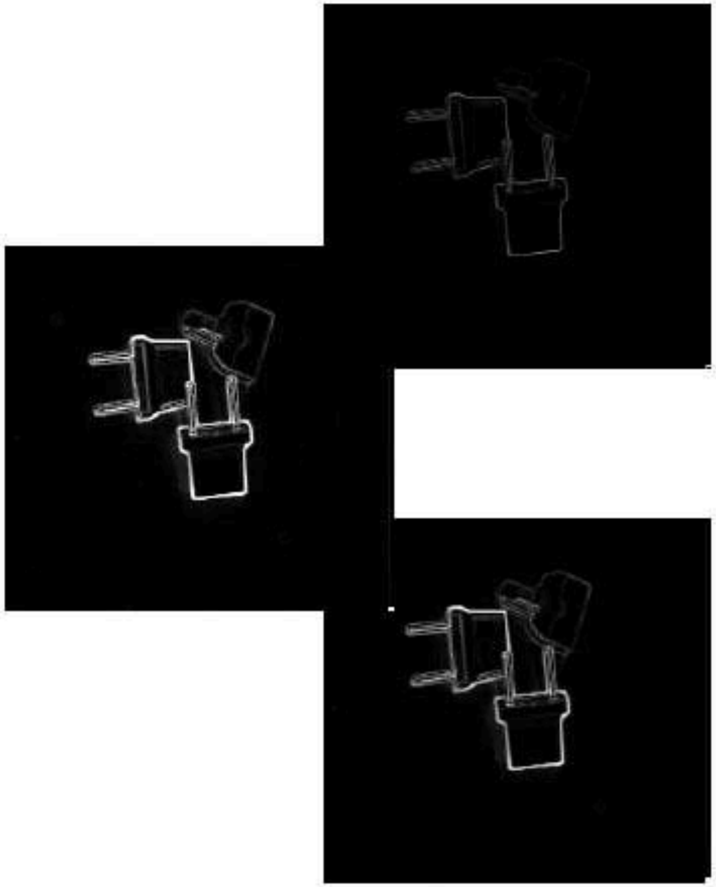
$$G_x \approx \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Sobel Operator

$$G_x \approx \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

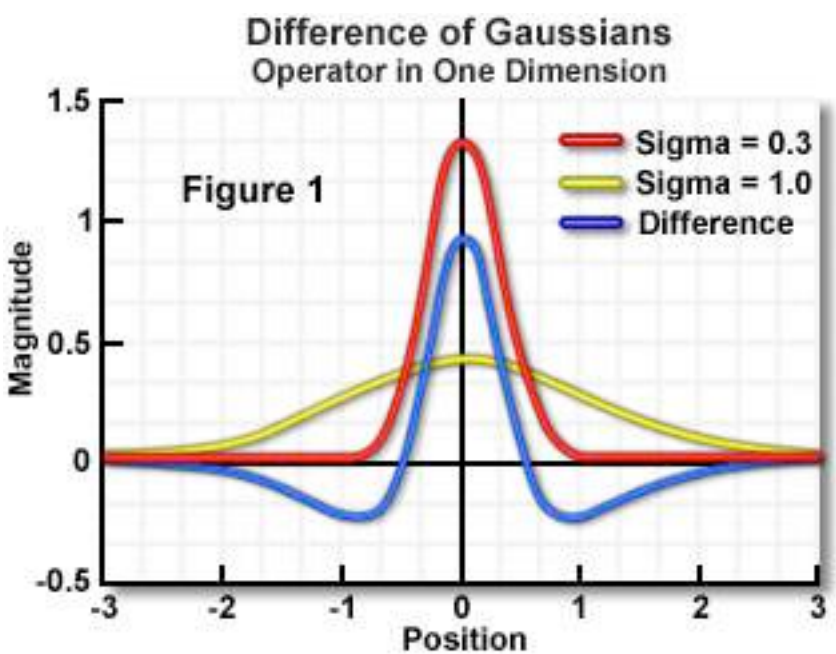
Prewitt Operator

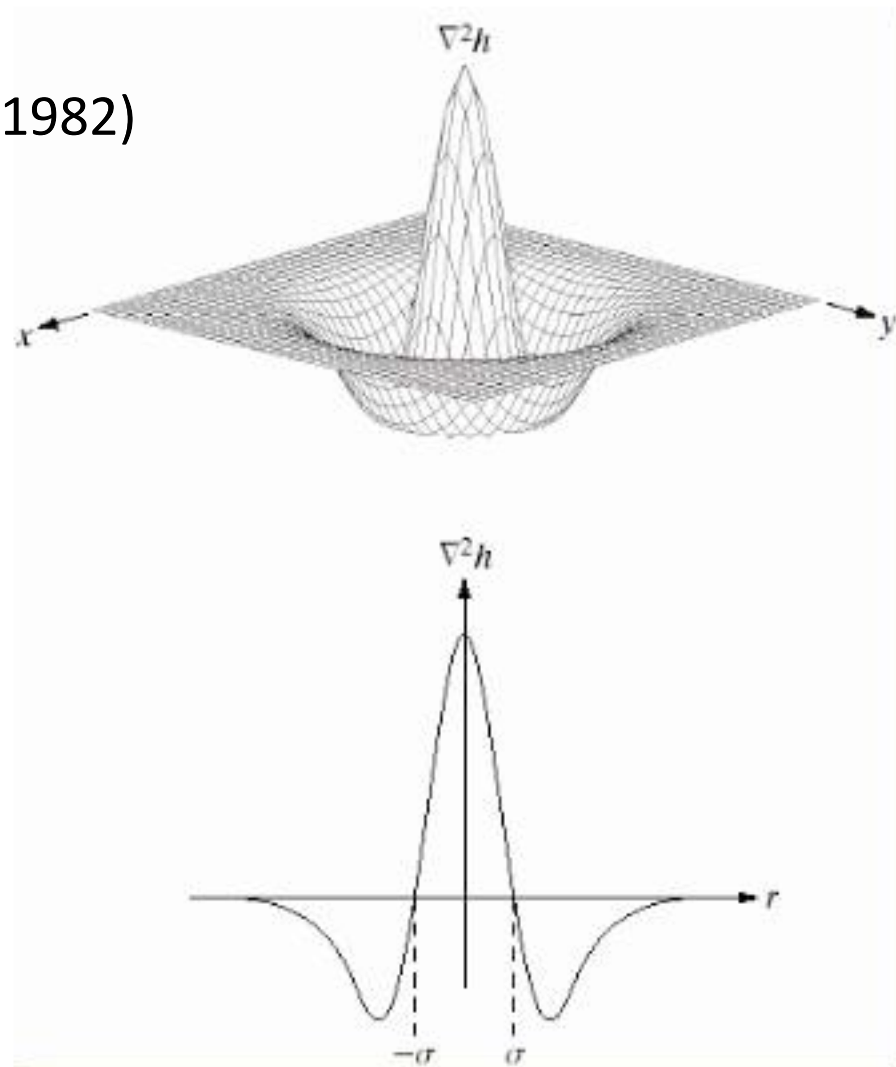
$$G_x \approx \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \qquad G_y \approx \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



A Variety of Image Filters

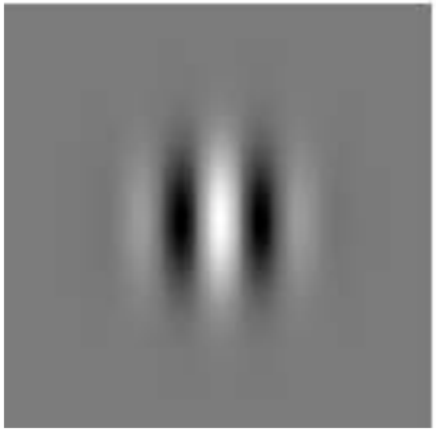
Laplacian of Gaussians (LoG) (Marr 1982)

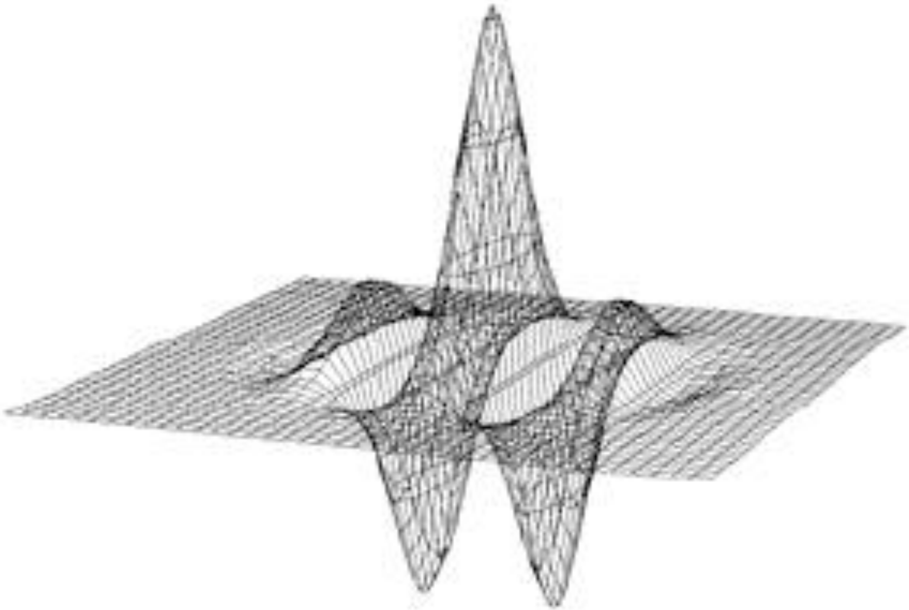




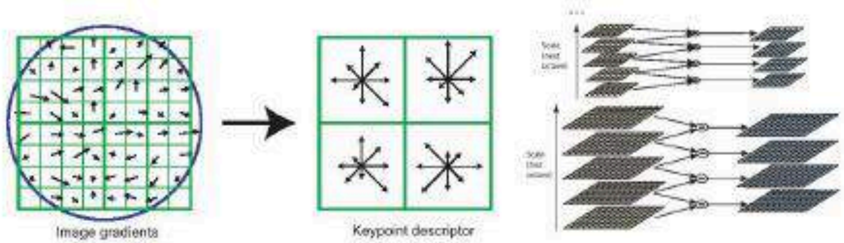
A Variety of Image Filters

Gabor filters (directional) (Daugman 1985)

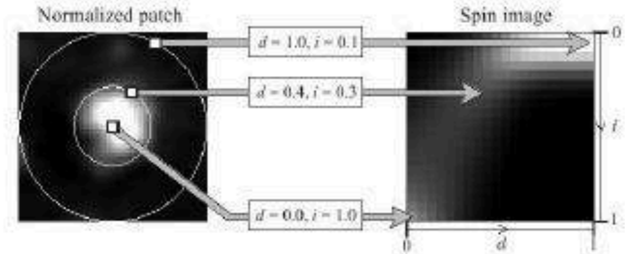




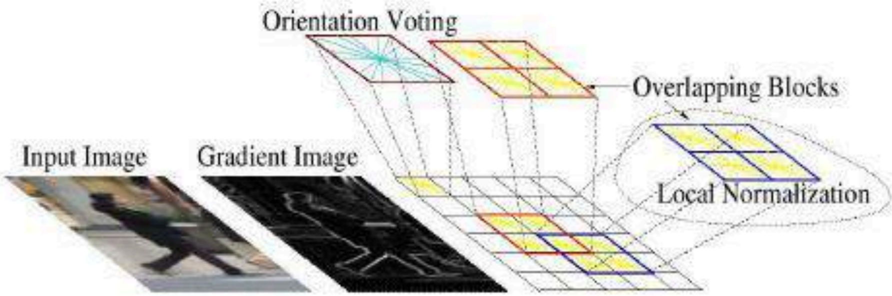
A Variety of Image Filters



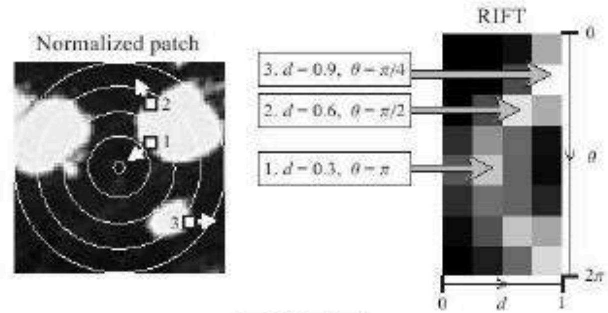
SIFT



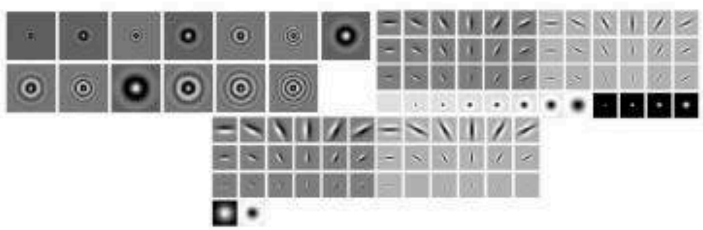
Spin image

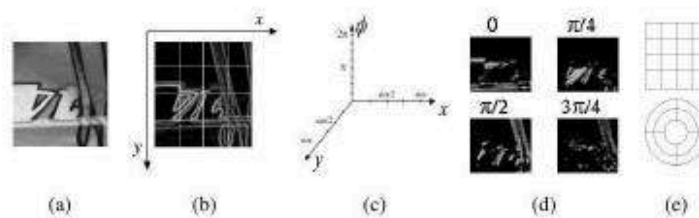


HoG



RIFT


Textons



GLOH

From: M. Sebag

Traditional *vs* Deep Learning Approach

Traditional approach

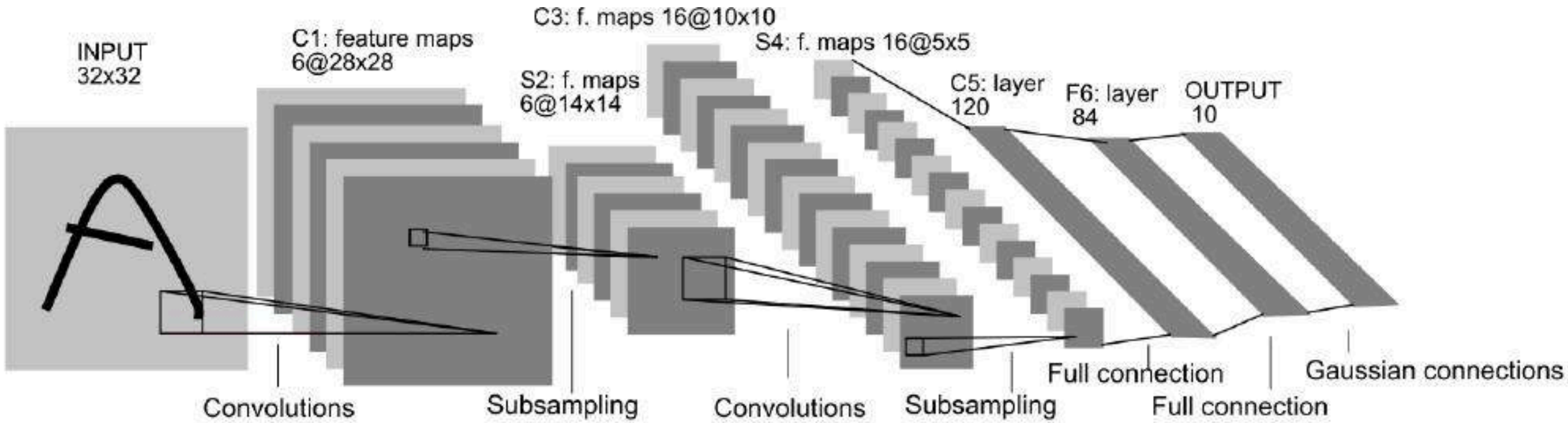
→ Manually crafted features → Trainable classifier

Deep learning

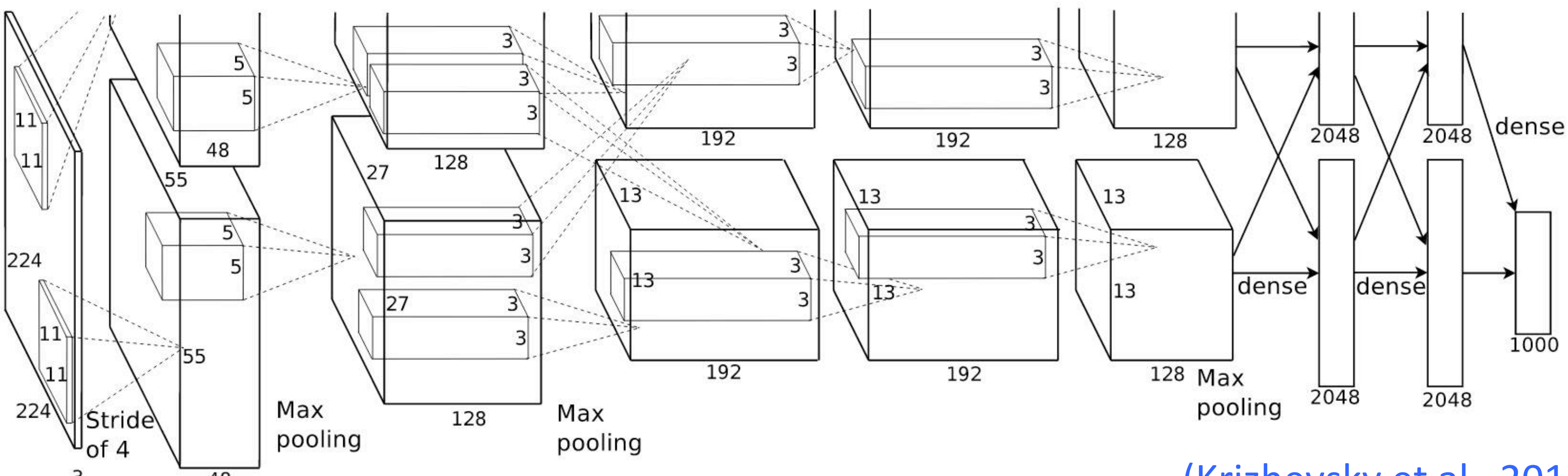
→ Trainable feature extractor → Trainable classifier

From: M. Sebag

Convolutional Neural Networks (CNNs)



(LeCun 1998)



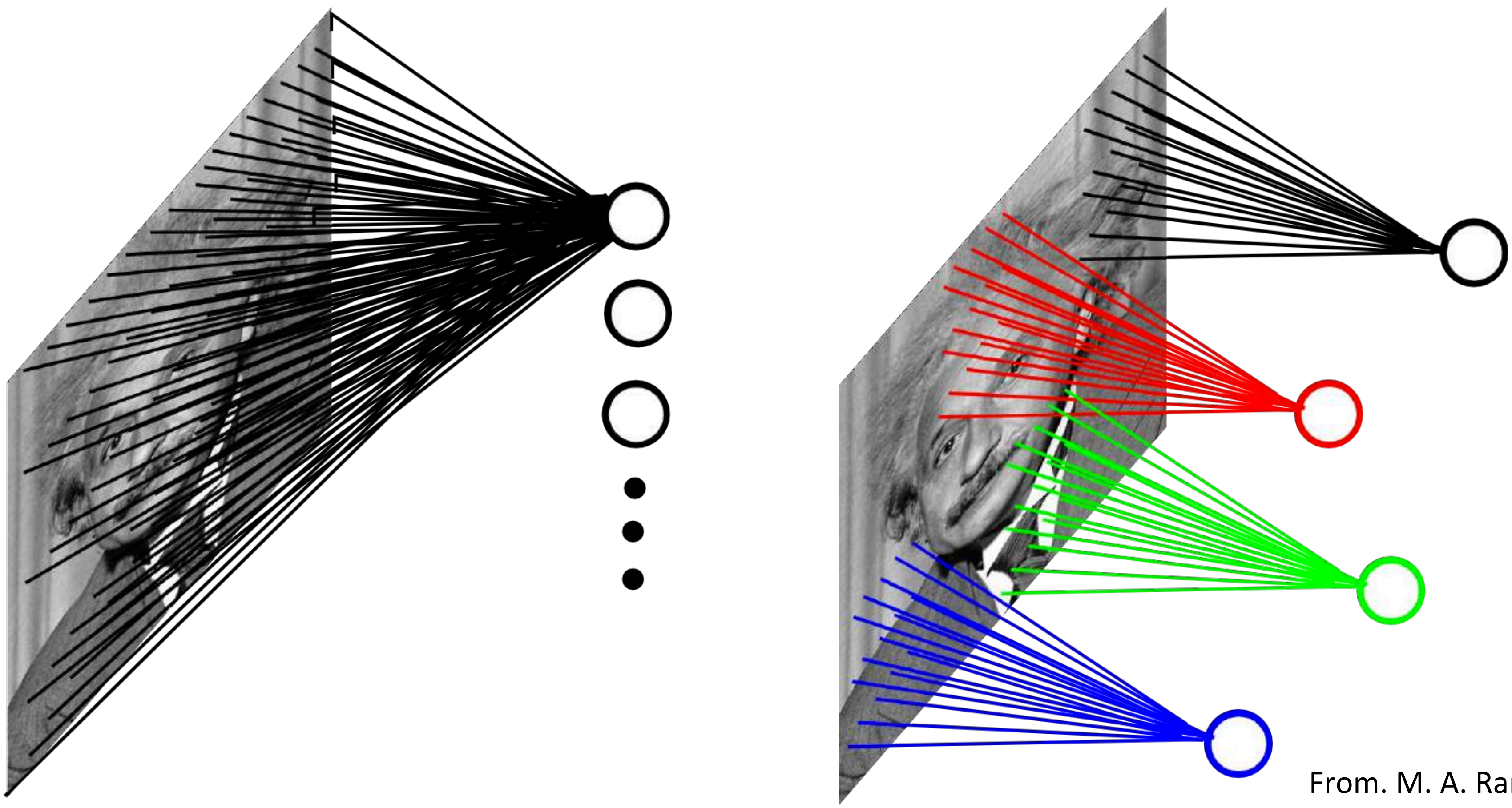
(Krizhevsky et al. 2012)

Fully- *vs* Locally-Connected Networks

Fully-connected: 400,000 hidden units = 16 billion parameters

Locally-connected: 400,000 hidden units 10 x 10 fields = 40 million parameters

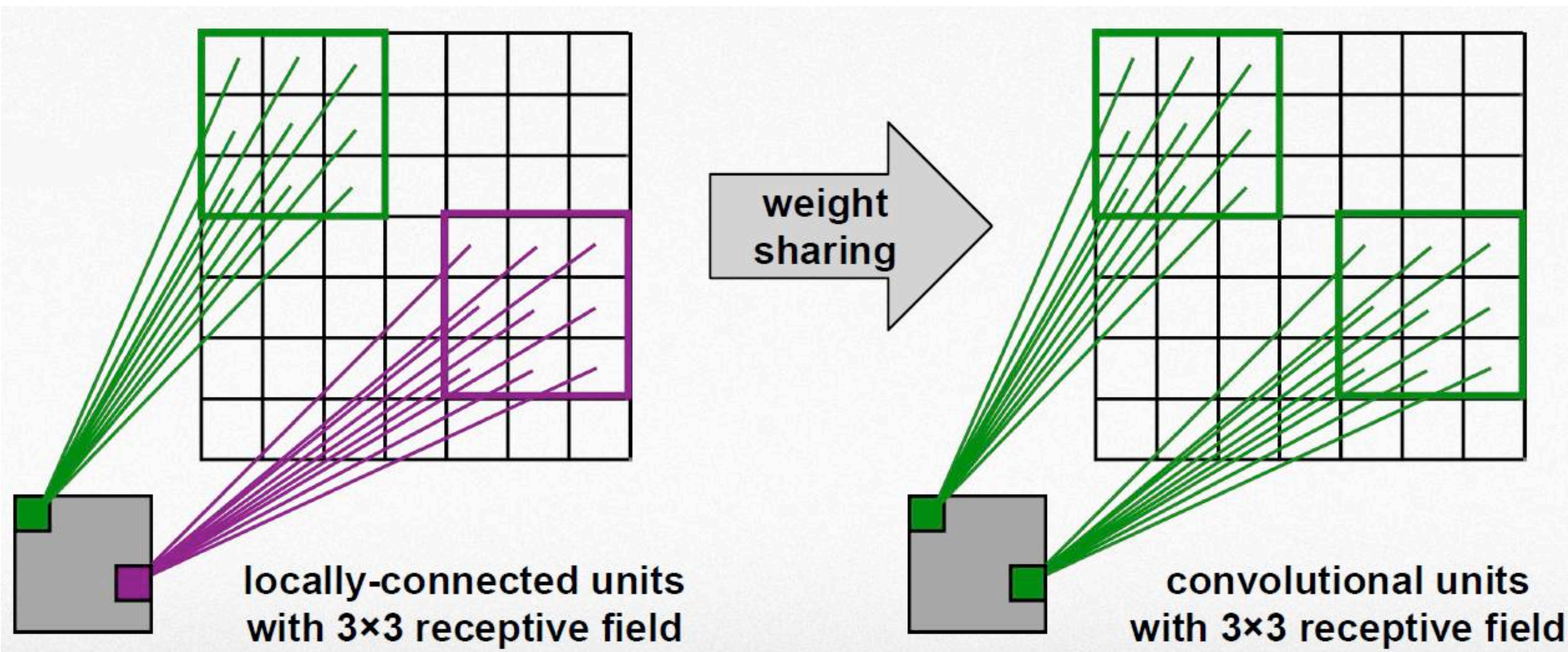
Local connections capture local dependencies



From. M. A. Ranzato

Weight Sharing

We can dramatically reduce the number of parameters by making one reasonable assumption: That if one feature is useful to compute at some spatial position (x1,y1), then it should also be useful to compute at a different position (x2,y2).



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

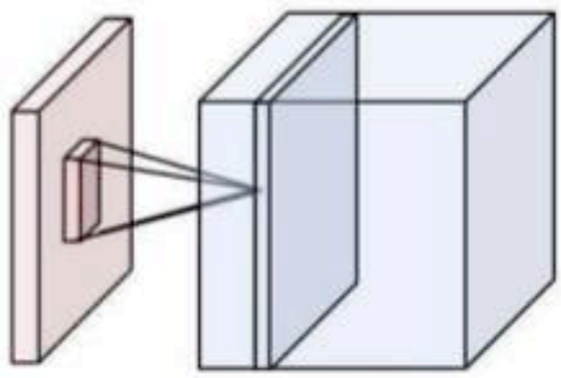
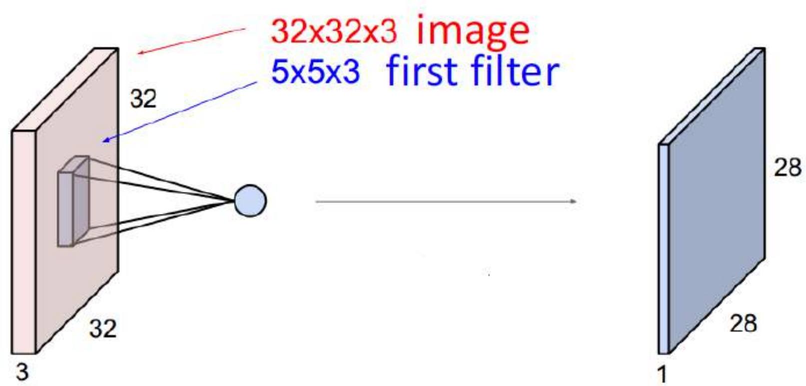
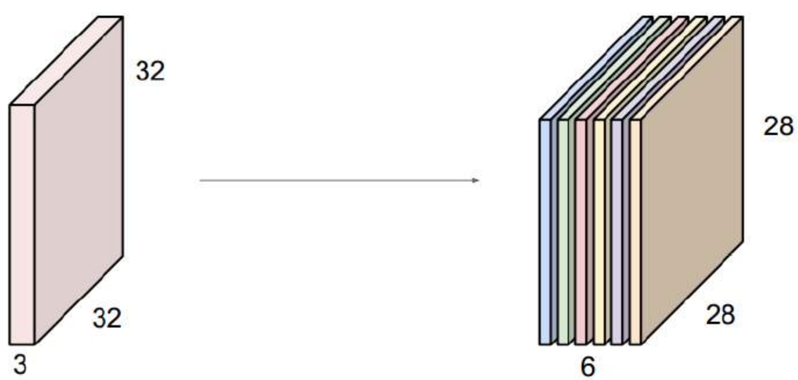


Image credit: A. Karpathy

Using Several Trainable Filters

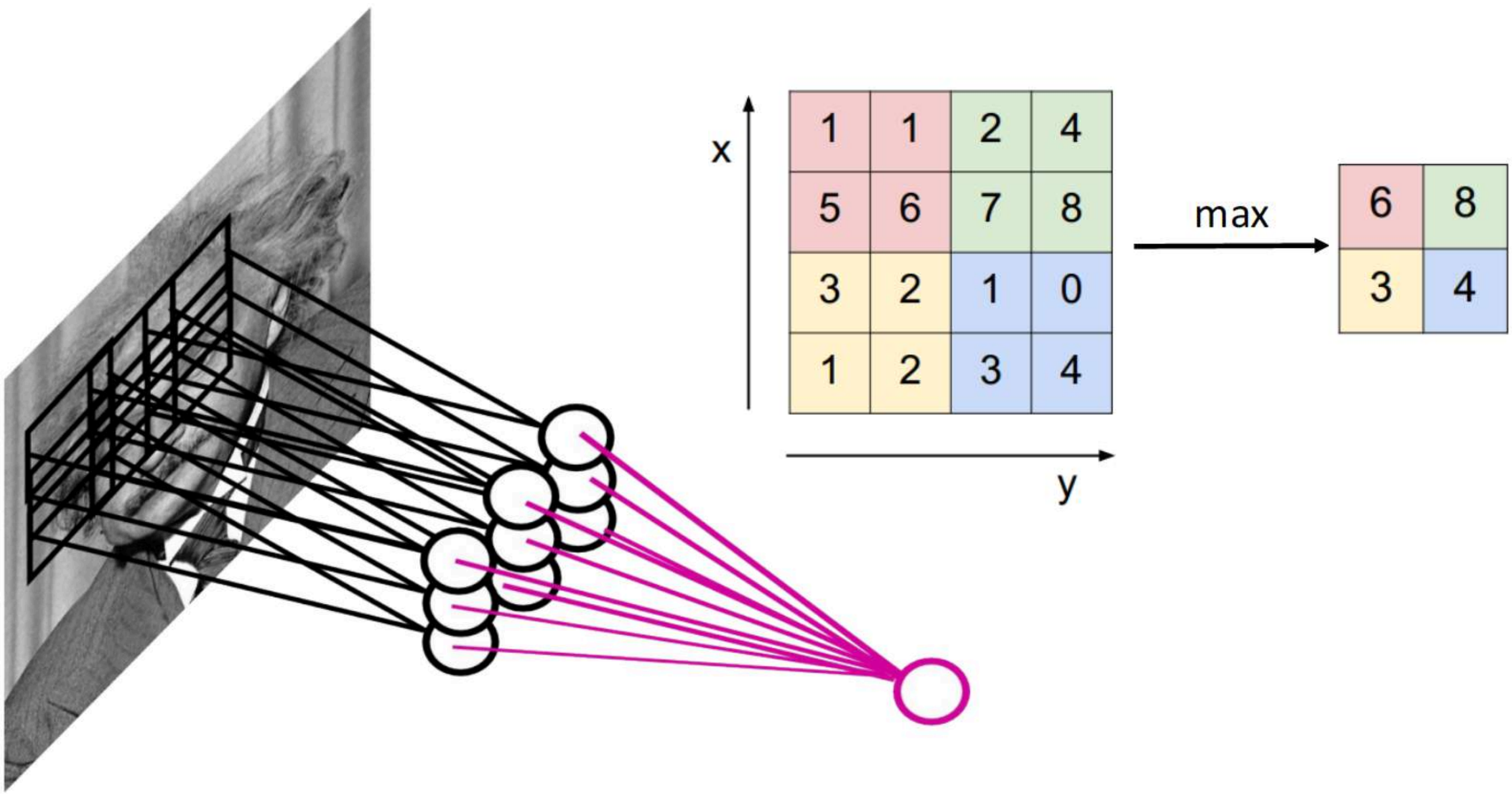
Normally, several filters are packed together and learnt automatically during training



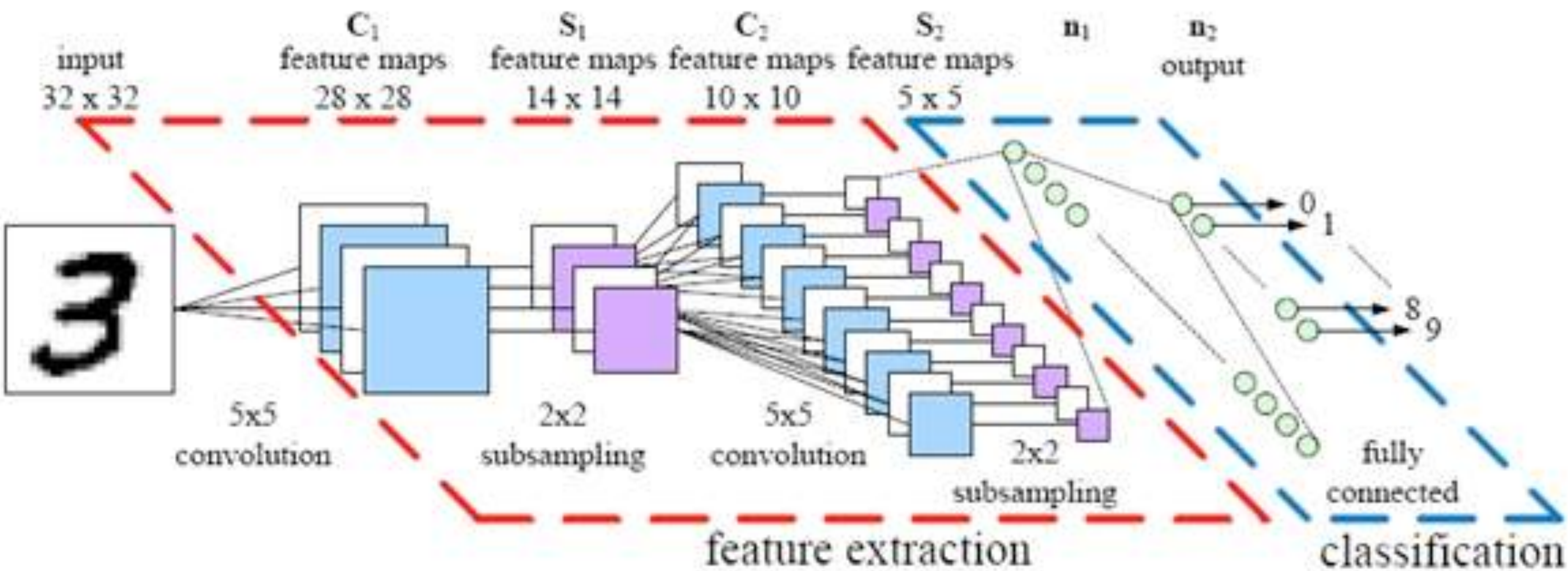


Pooling

Max pooling is a way to simplify the network architecture, by downsampling the number of neurons resulting from filtering operations.



Combining Feature Extraction and Classification



AlexNet (2012)

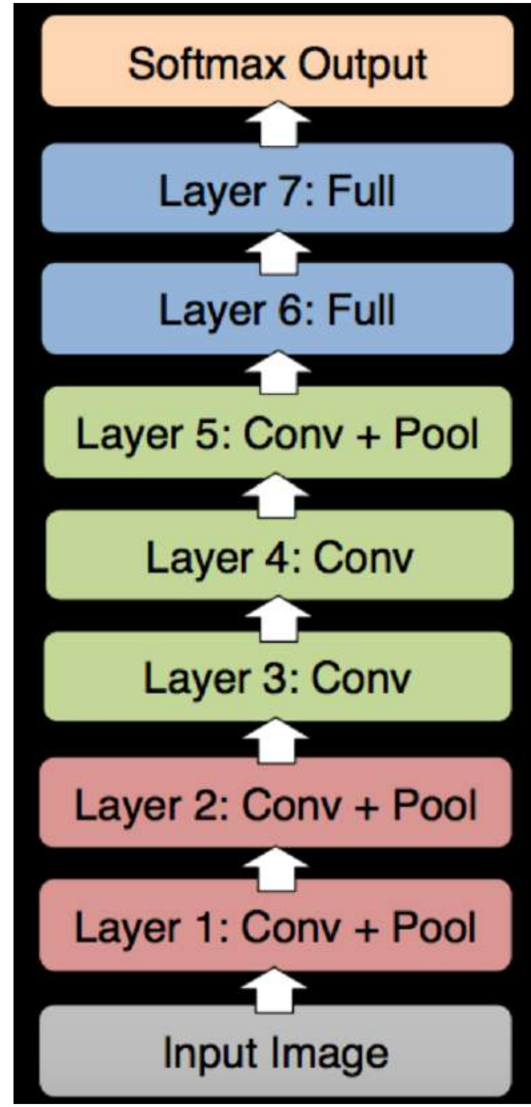
ImageNet Classification with Deep Convolutional Neural Networks

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

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

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

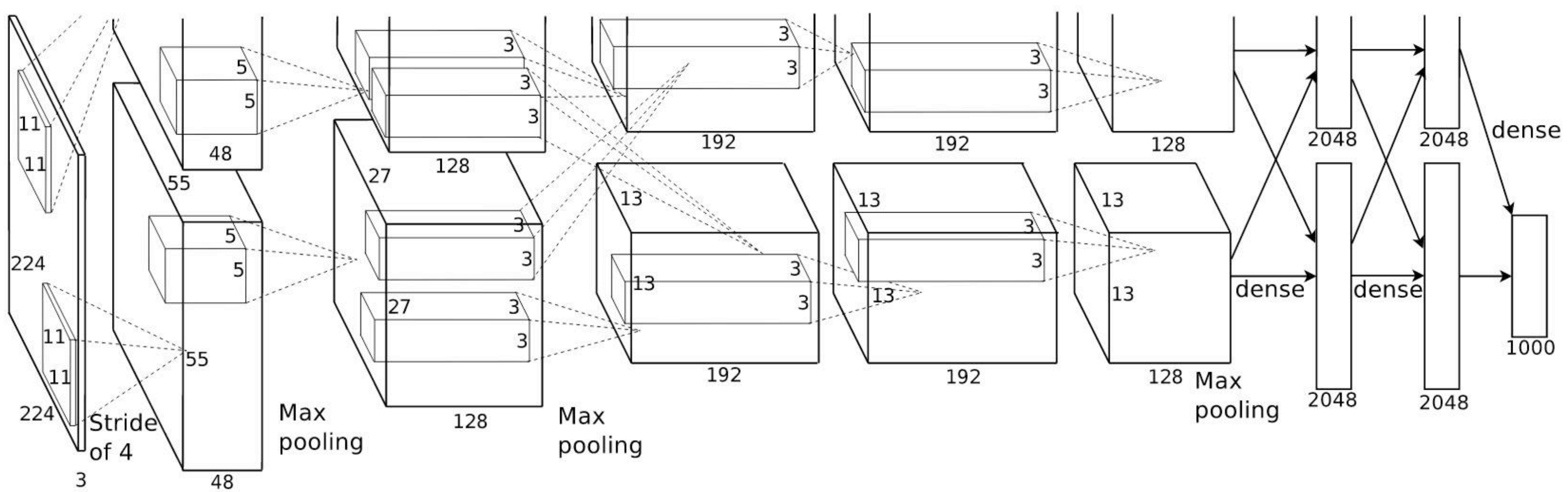
- 8 layers total
- Trained on Imagenet Dataset (1000 categories, 1.2M training images, 150k test images)



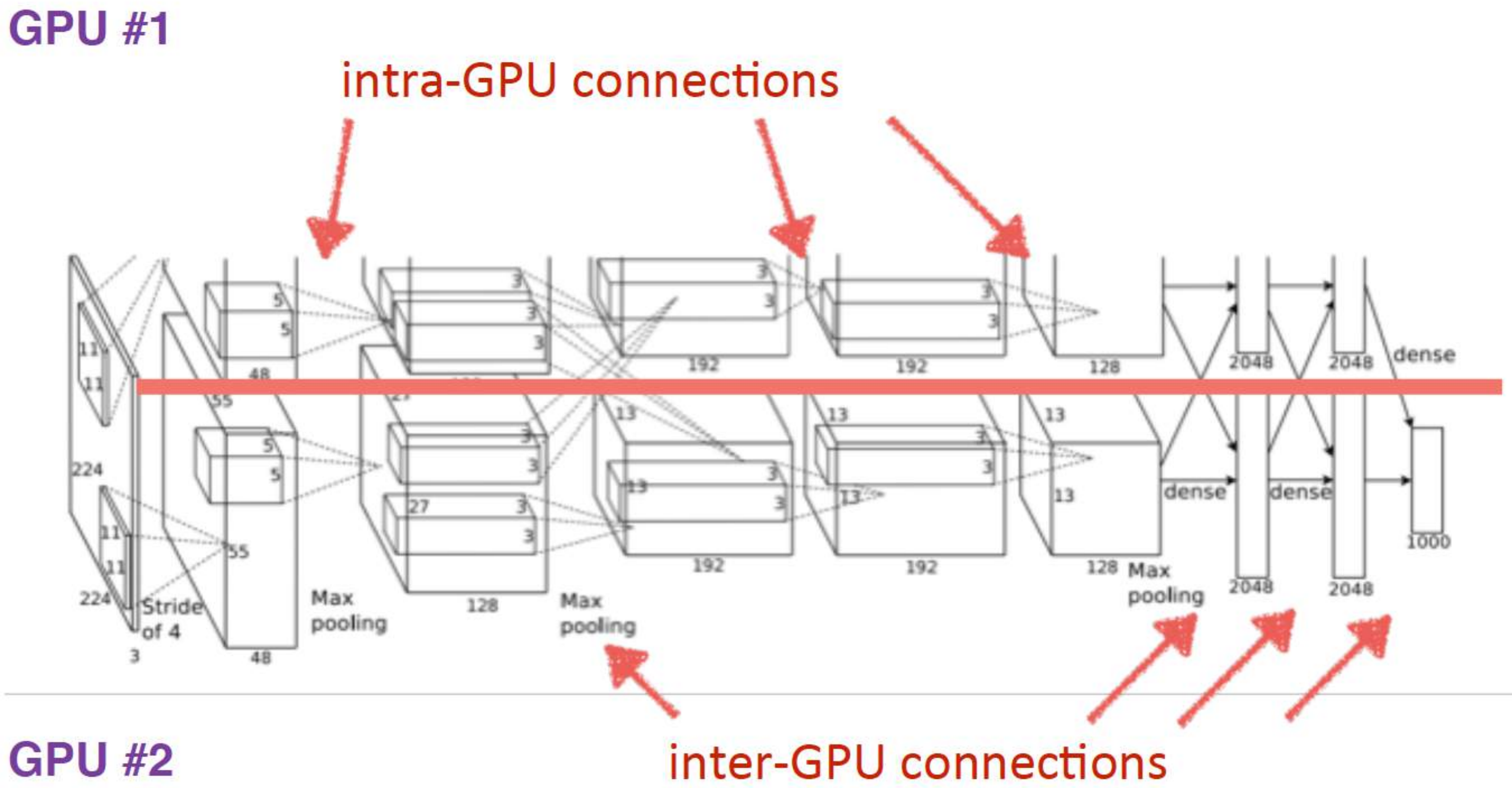
AlexNet Architecture

- 1st layer: 96 kernels (11 x 11 x 3)
- Normalized, pooled
- 2nd layer: 256 kernels (5 x 5 x 48)
- Normalized, pooled
- 3rd layer: 384 kernels (3 x 3 x 256)
- 4th layer: 384 kernels (3 x 3 x 192)
- 5th layer: 256 kernels (3 x 3 x 192)
- Followed by 2 fully connected layers, 4096 neurons each
- Followed by a 1000-way SoftMax layer

650,000 neurons
60 million parameters

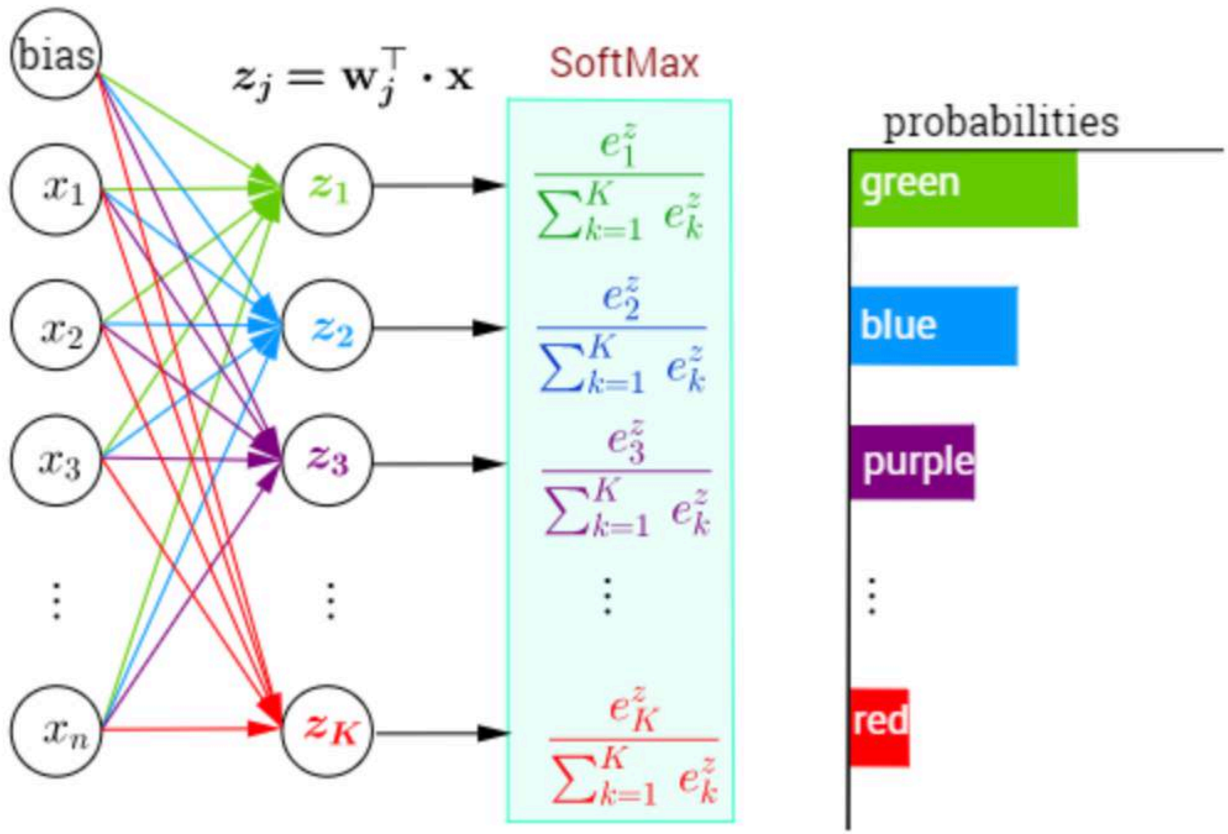


Training on Multiple GPU's



Output Layer: Softmax

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ z_K \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \\ \vdots \\ \mathbf{w}_K^\top \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

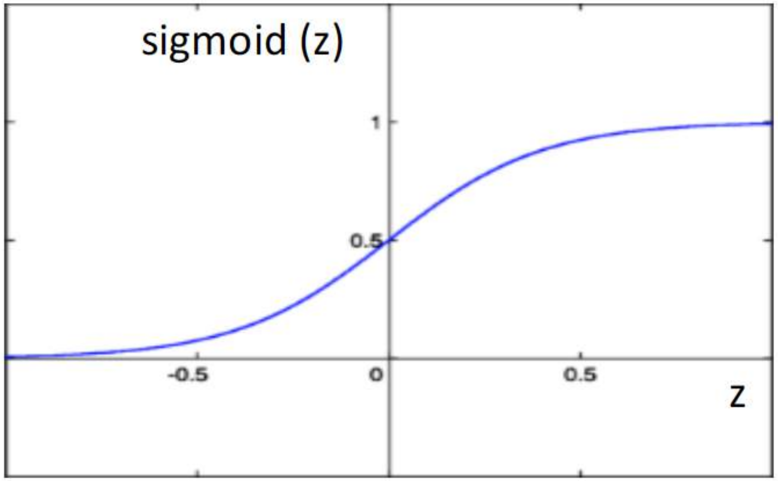


Rectified Linear Units (ReLU's)

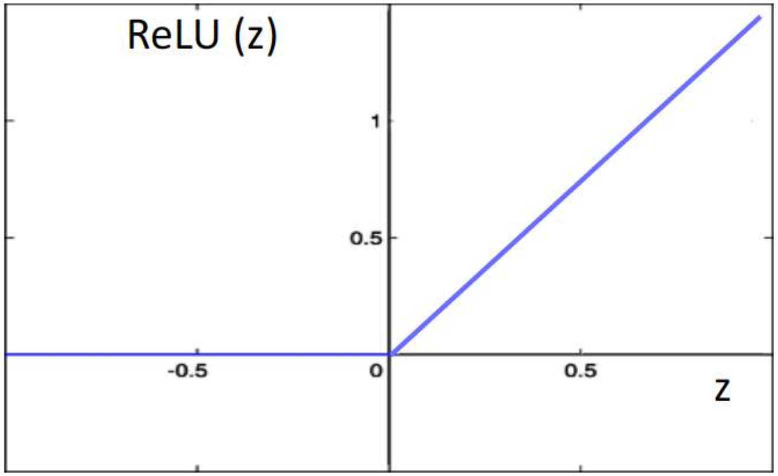
Problem: Sigmoid activation takes on values in (0,1). Propagating the gradient back to the initial layers, it tends to become 0 (vanishing gradient problem).

From a practical perspective, this slows down the training procedure of the initial layers of the network.

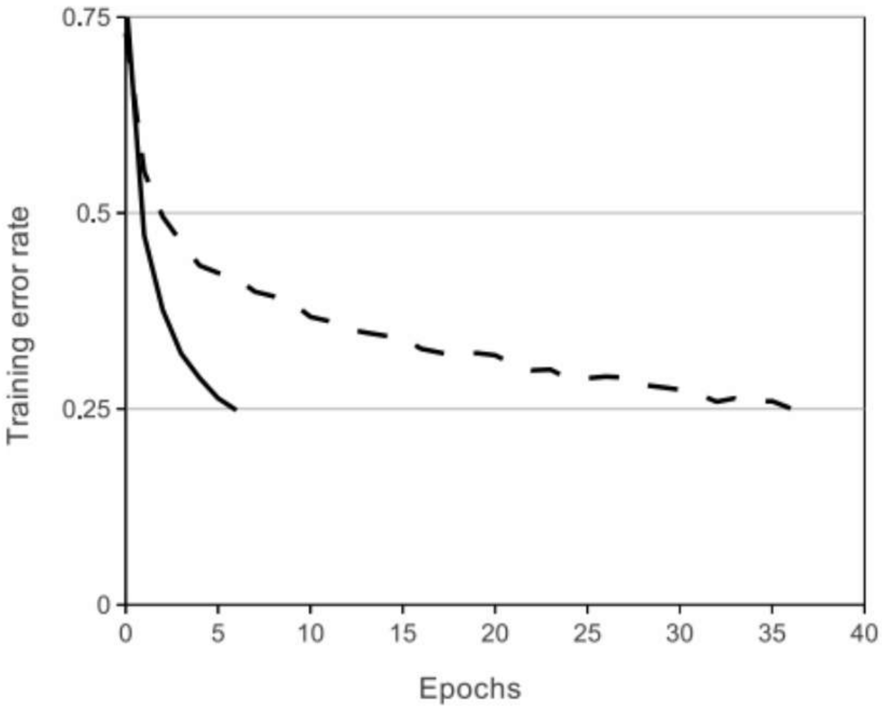
sigmoid(z) = $\frac{1}{1+e^{-z}}$



ReLU(z) = $\max(0, z)$



Rectified Linear Units (ReLU's)



A 4 layer CNN with ReLUs (solid line) converges six times faster than an equivalent network with tanh neurons (dashed line) on CIFAR-10 dataset

Mini-batch Stochastic Gradient Descent

Loop:

1. **Sample** a batch of data
2. **Forward** prop it through the graph, get loss
3. **Backprop** to calculate the gradients
4. **Update** the parameters using the gradient

Data Augmentation

The easiest and most common method to reduce overfitting on image data is to artificially enlarge the dataset using label-preserving transformations

AlexNet uses two forms of this **data augmentation**.

- The first form consists of generating image translations and horizontal reflections.
- The second form consists of altering the intensities of the RGB channels in training images.

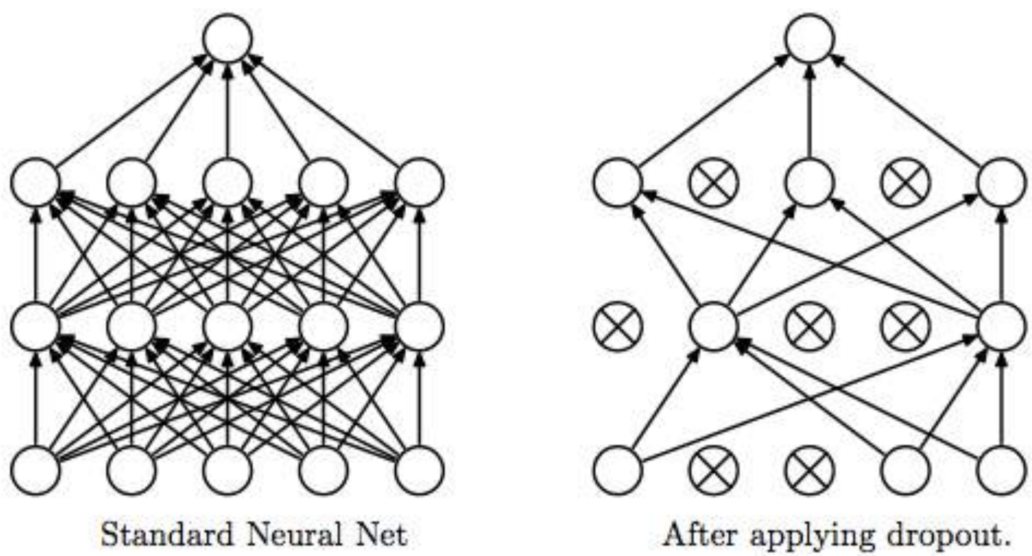
Dropout

Set to zero the output of each hidden neuron with probability 0.5.

The neurons which are "dropped out" in this way do not contribute to the forward pass and do not participate in backpropagation.

So every time an input is presented, the neural network samples a different architecture, but all these architectures share weights.

Reduces complex co-adaptations of neurons, since a neuron cannot rely on the presence of particular other neurons.



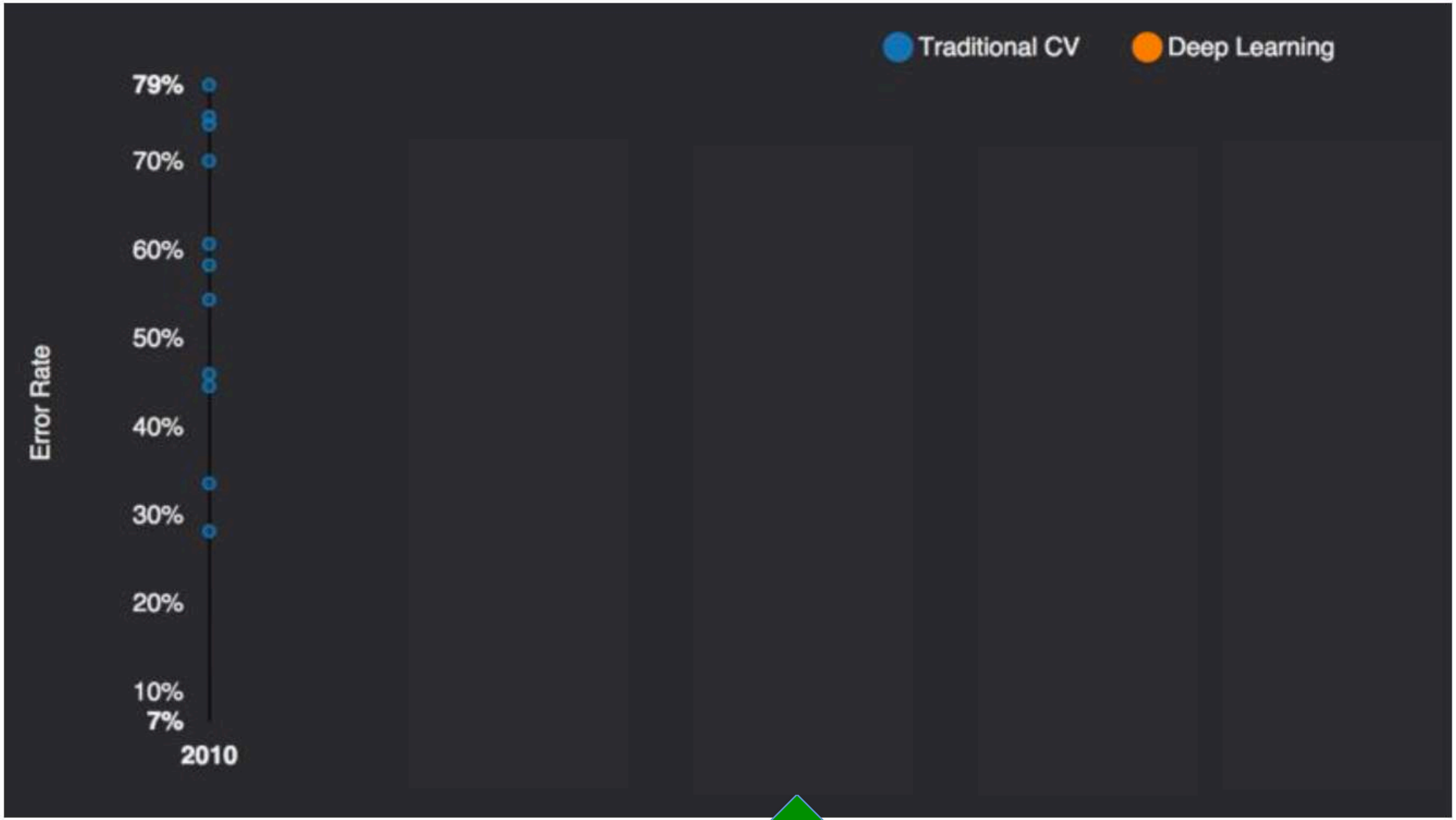
ImageNet

IMAGENET

[Deng et al. CVPR 2009]

- ~14 million labeled images, 20k classes
- Images gathered from Internet
- Human labels via Amazon Turk
- Challenge: 1.2 million training images, 1000 classes

ImageNet Challenges

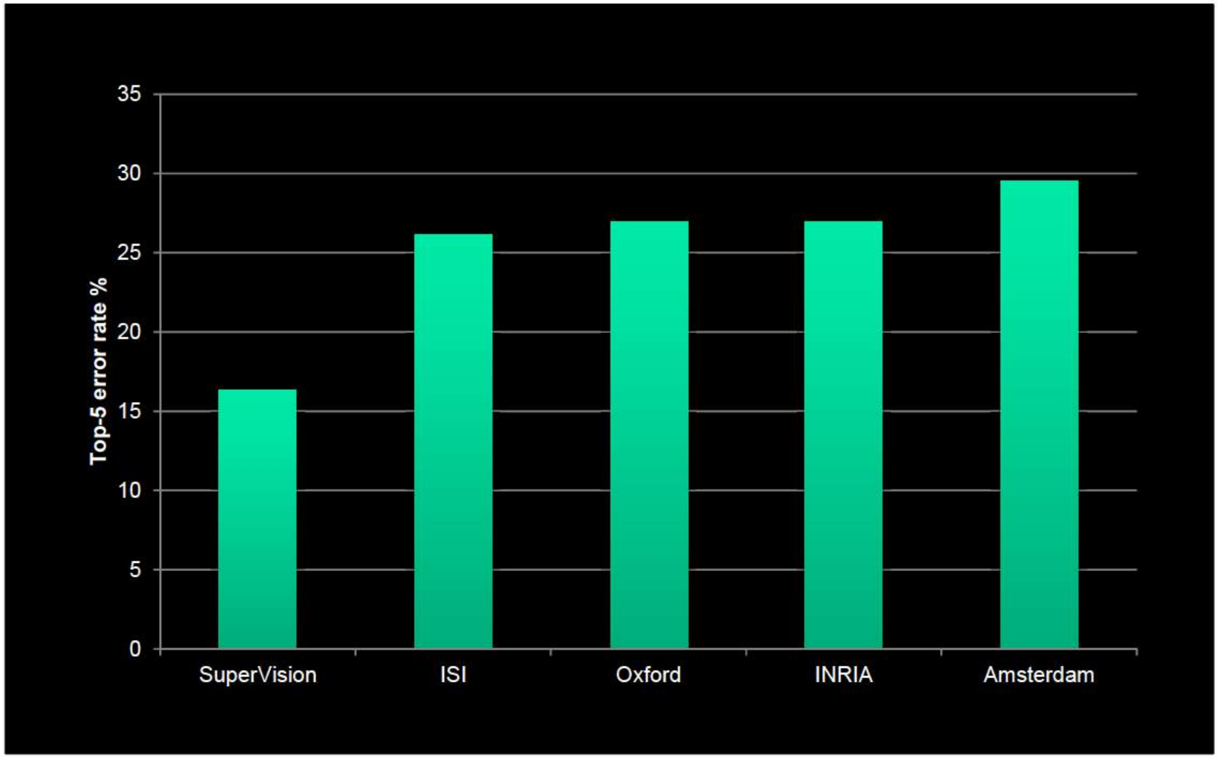


Deep learning!

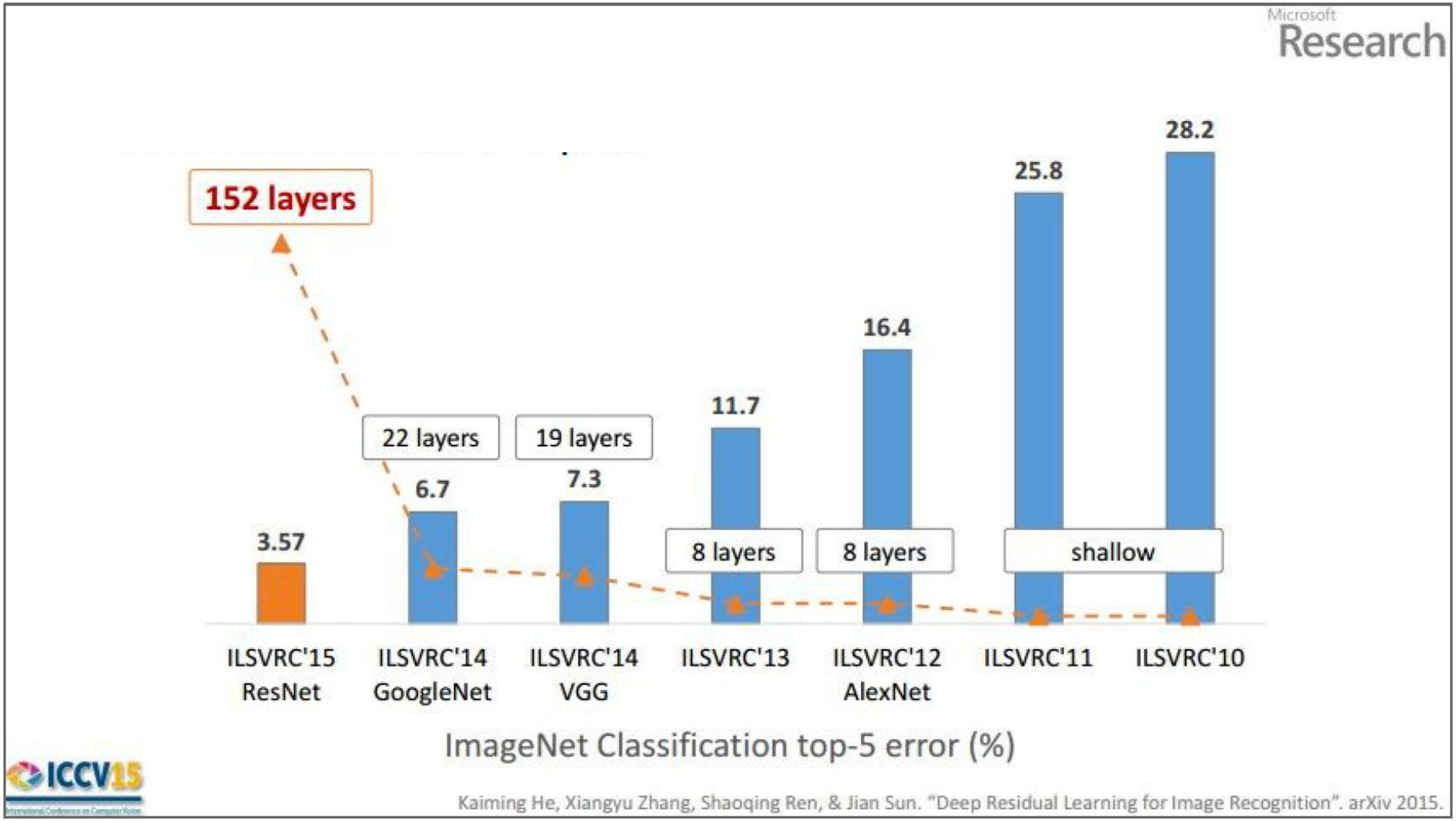
ImageNet Challenge 2012

Krizhevsky et al. -- **16.4% error** (top-5)

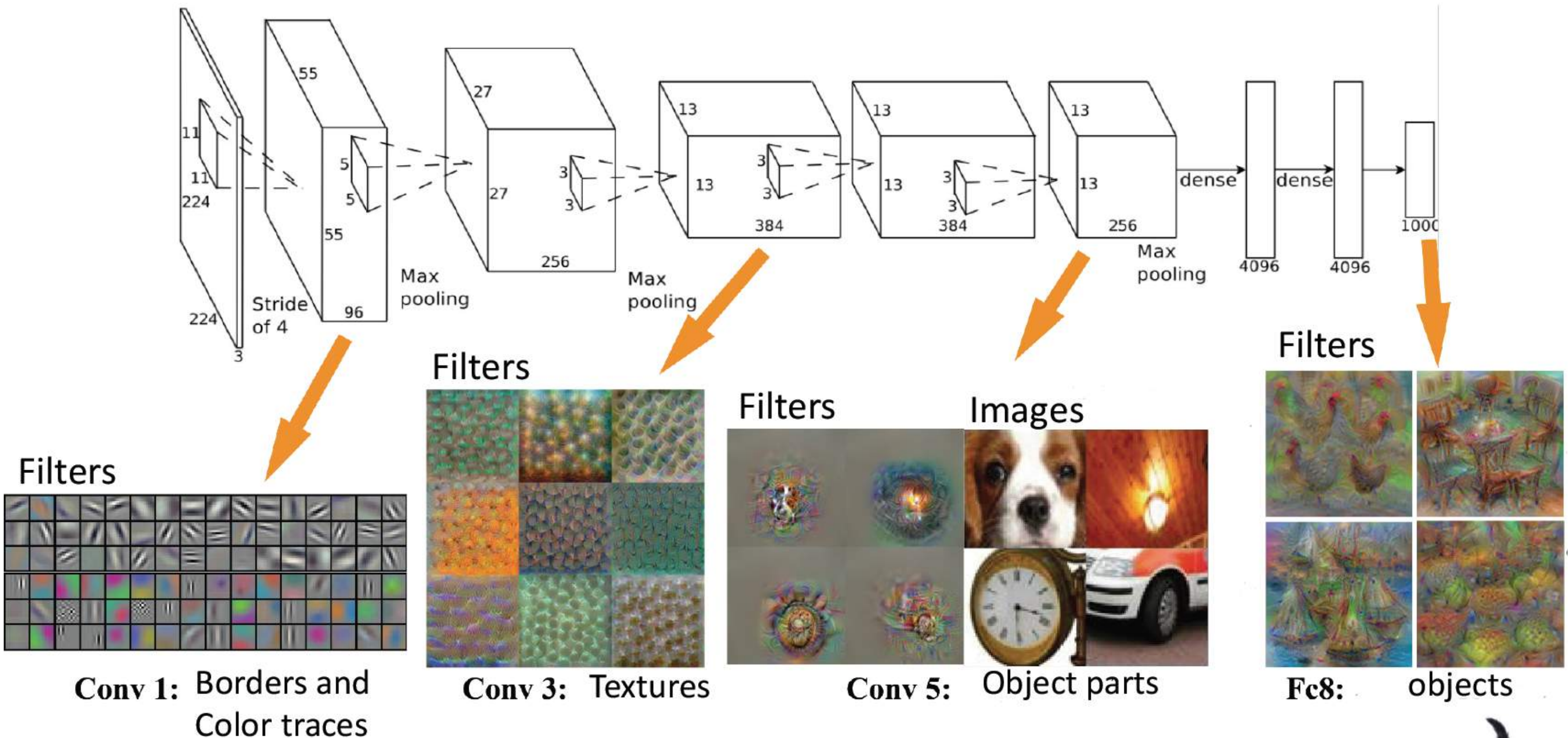
Next best (non-convnet) – **26.2% error**



Revolution of Depth



A Hierarchy of Features

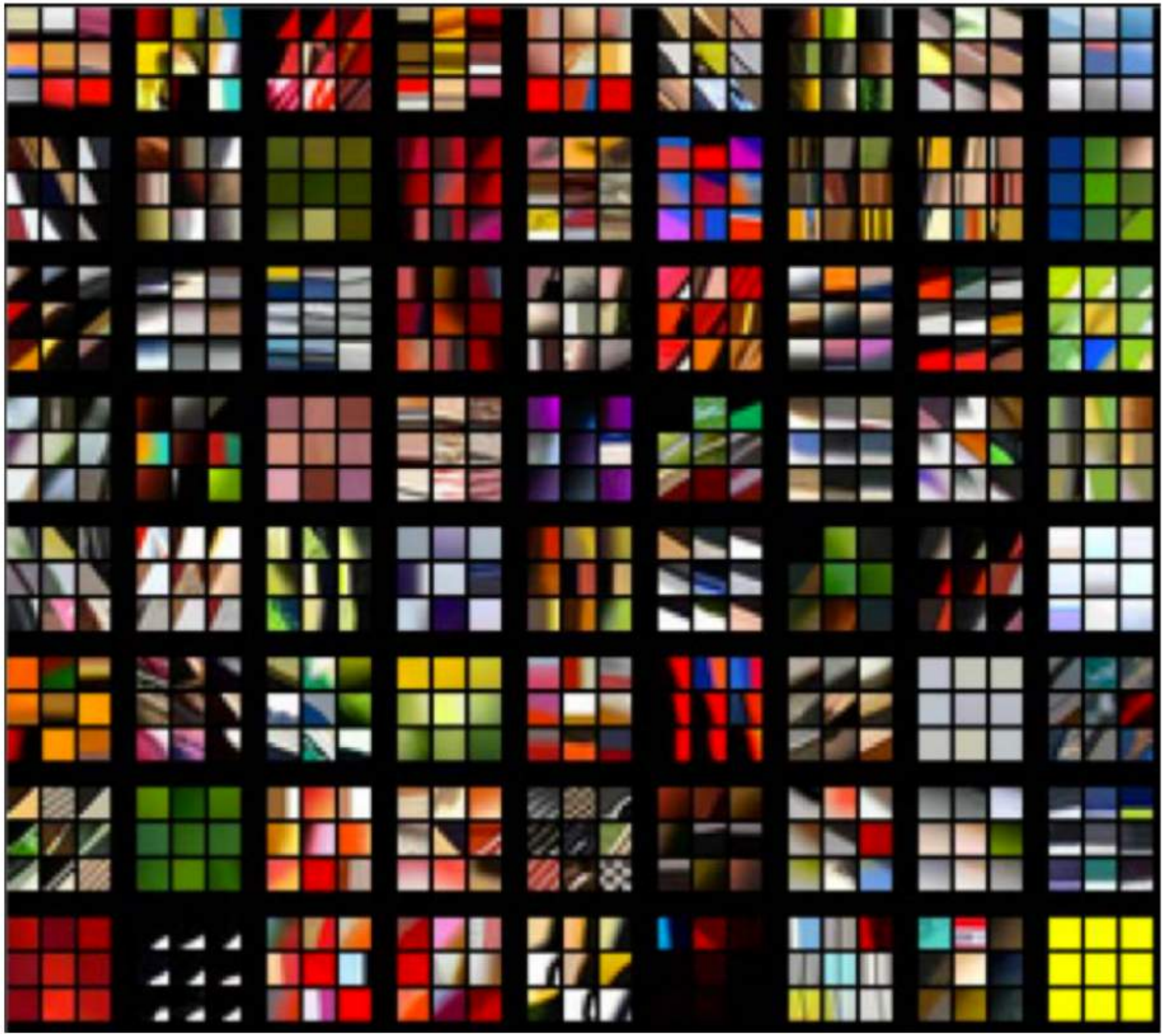


The deep network gradually learns more complex and abstract notions

From: B. Biggio

Layer 1

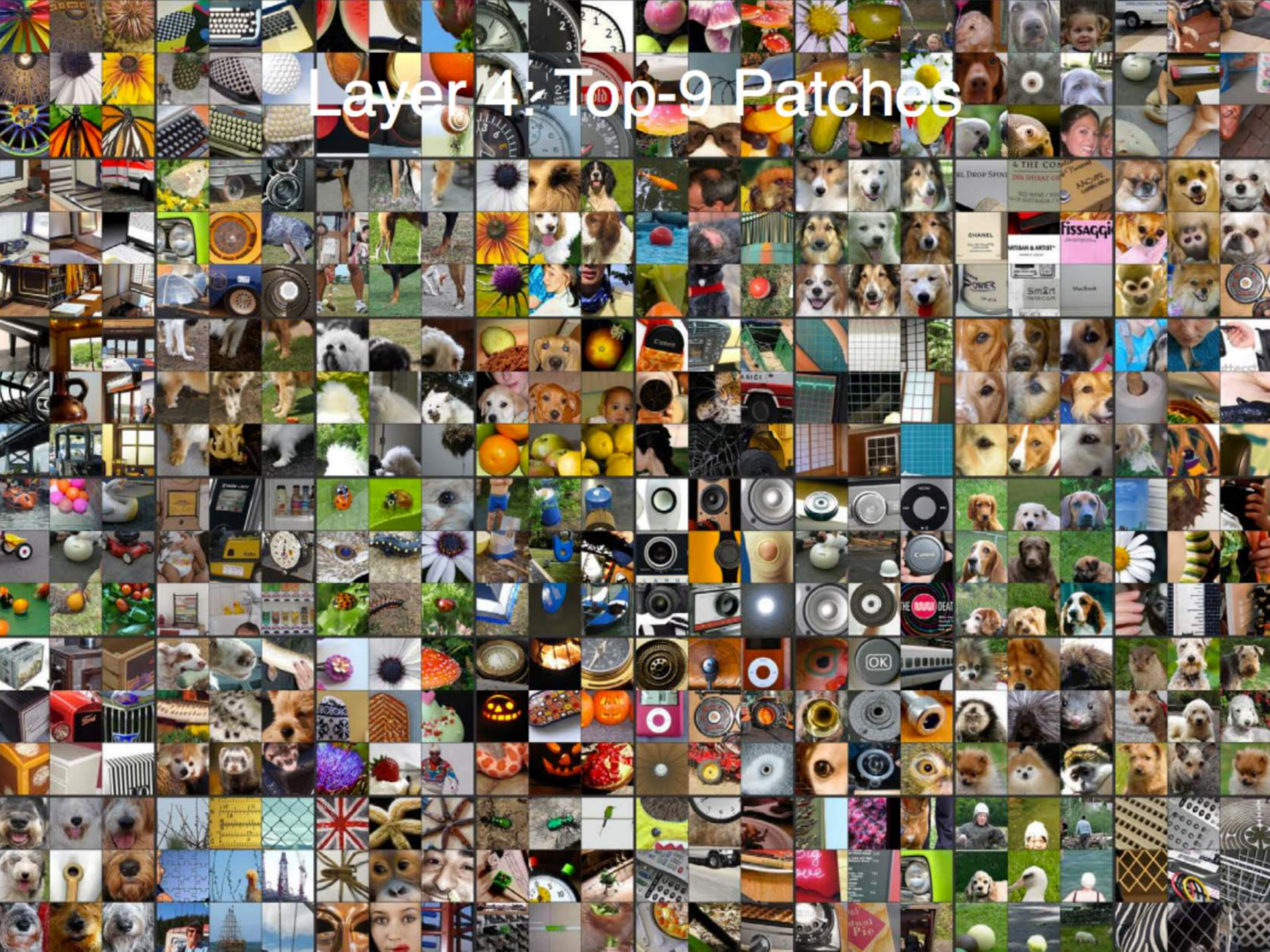
Each 3x3 block shows the top 9 patches for one filter



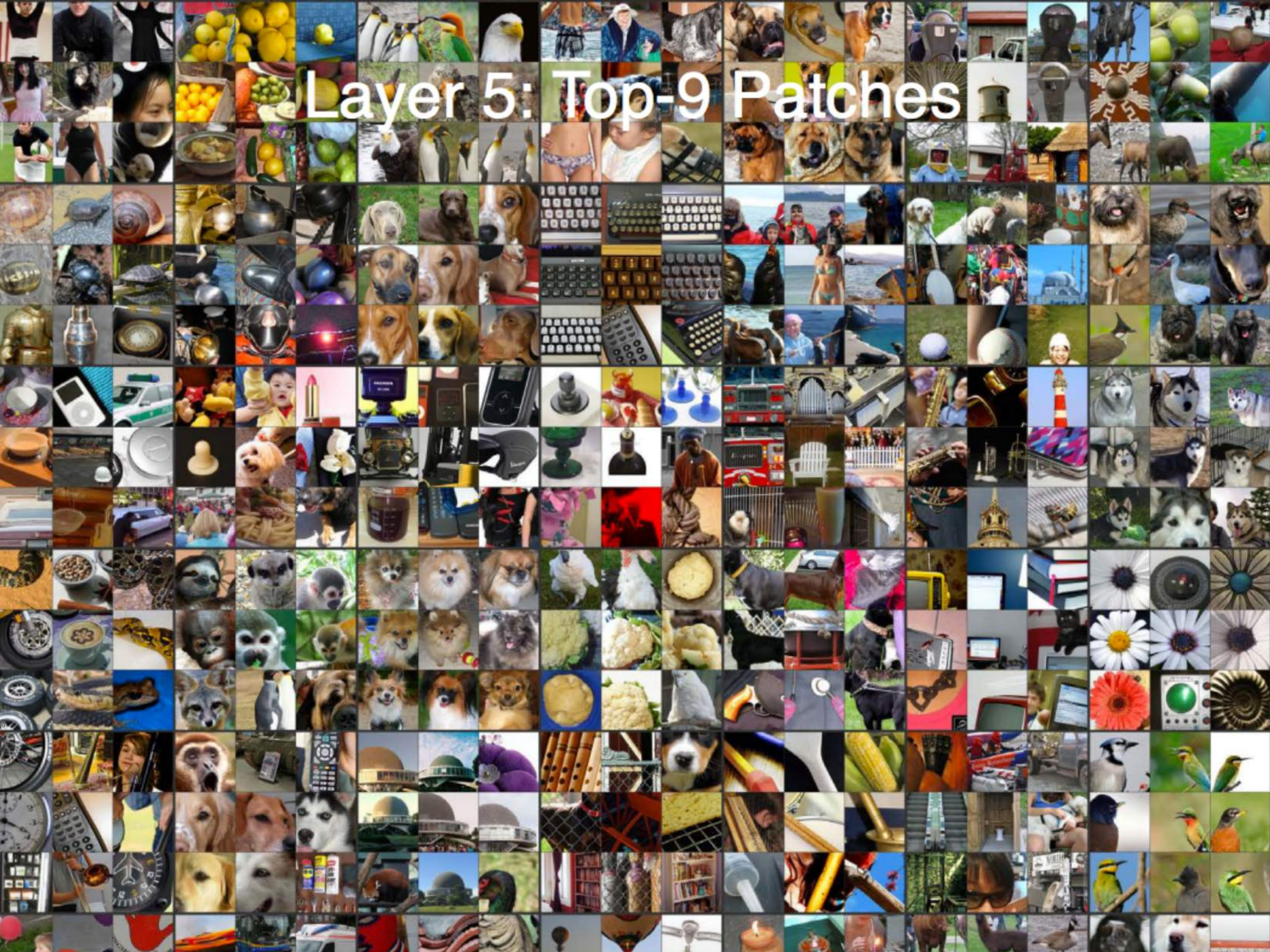
Layer 2

Layer 3: Top-9 Patches

Layer 4: Top-9 Patches



Layer 5: Top-9 Patches



Feature Analysis

- A well-trained ConvNet is an excellent **feature extractor**.

- Chop the network at desired layer and use the output as a feature representation to train an SVM on some other dataset (Zeiler-Fergus 2013):

	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)	$\mathbf{85.4 \pm 0.4}$	$\mathbf{72.6 \pm 0.1}$

- Improve further by taking a pre-trained ConvNet and re-training it on a different dataset (Fine tuning).

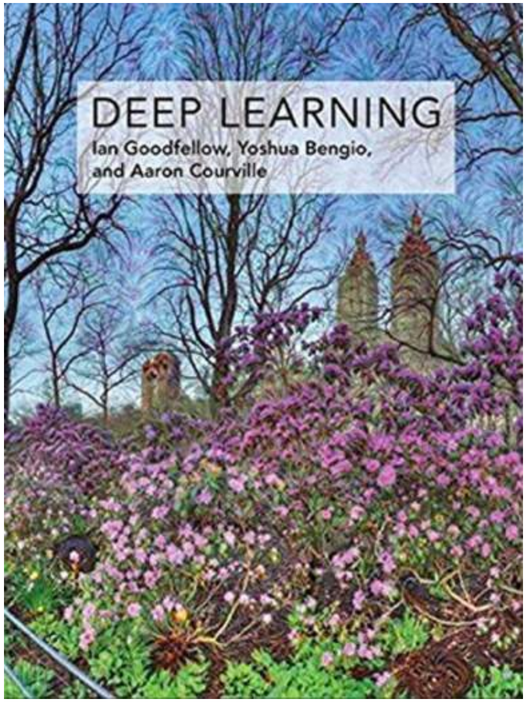
Other Success Stories of Deep Learning

Today deep learning, in its several manifestations, is being applied in a variety of different domains besides computer vision, such as:

- Speech recognition
- Optical character recognition
- Natural language processing
- Autonomous driving
- Game playing (e.g., Google's AlphaGo)
- …

References

- http://neuralnetworksanddeeplearning.com
- http://deeplearning.stanford.edu/tutorial/
- http://www.deeplearningbook.org/
- http://deeplearning.net/



Platforms:

- Theano
- PyTorch
- TensorFlow
- …