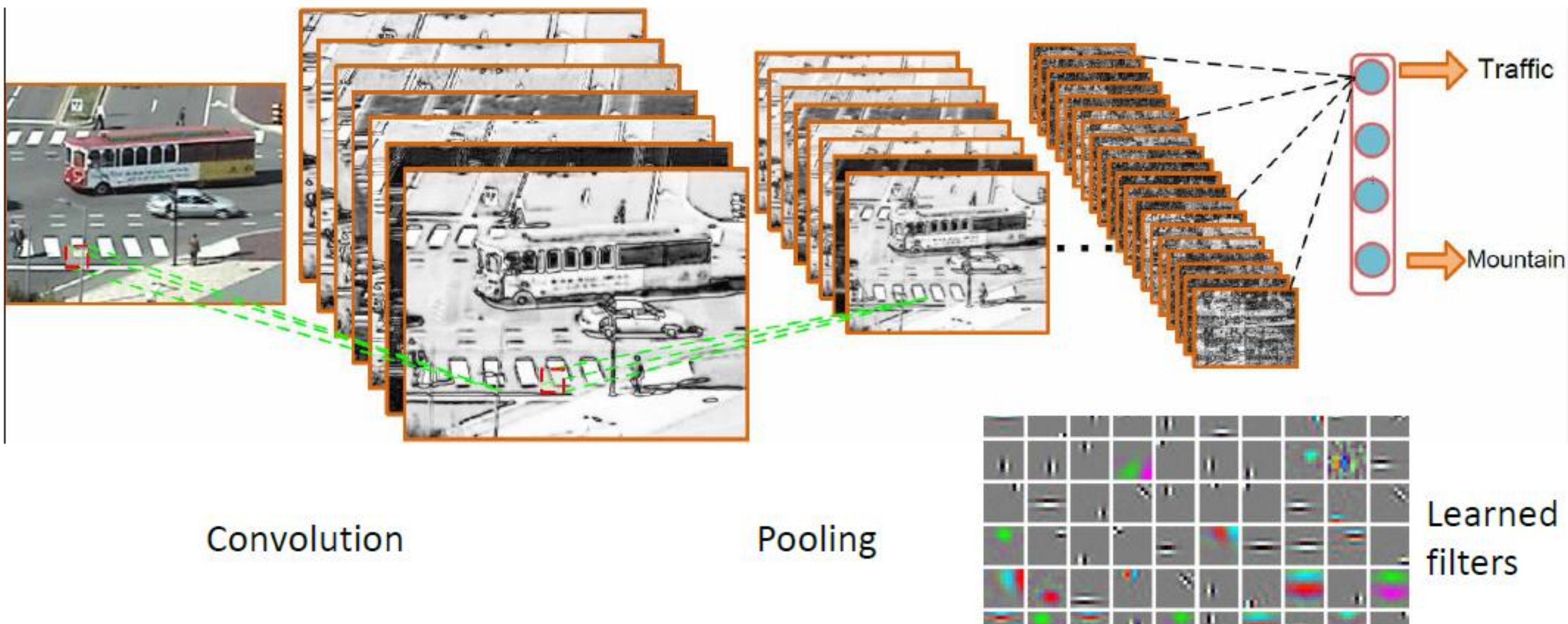
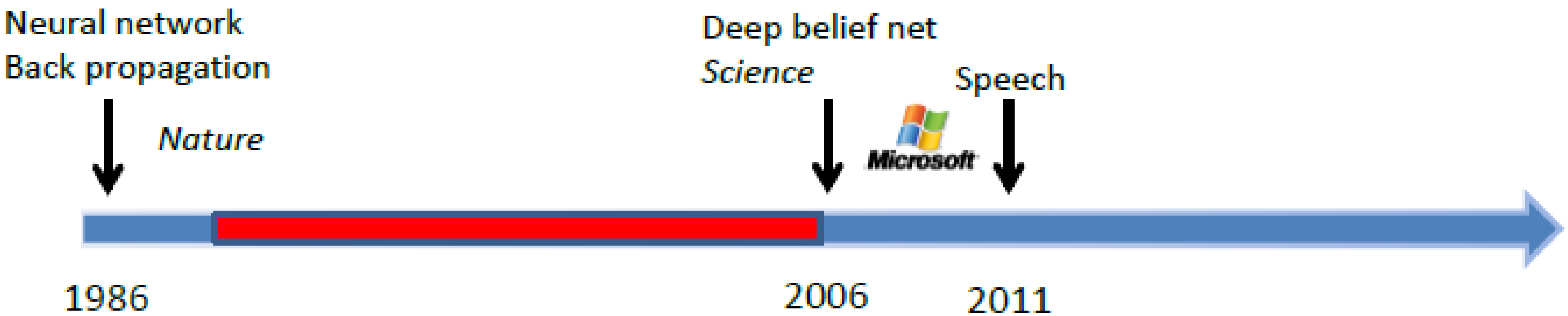
Classical Deep Models

- Convolutional Neural Networks (CNN)
  - First proposed by Fukushima in 1980
  - Improved by LeCun, Bottou, Bengio and Haffner in 1998



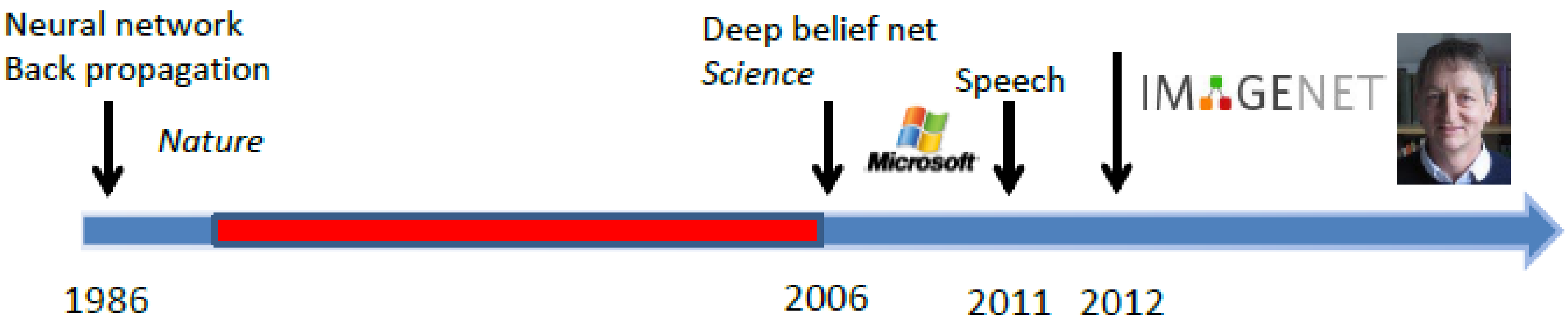


deep learning results

task	hours of training data	DNN-HMM	GMM-HMM with same data
Switchboard (test set 1)	309	18.5	27.4
Switchboard (test set 2)	309	16.1	23.6
English Broadcast News	50	17.5	18.8
Bing Voice Search (Sentence error rates)	24	30.4	36.2
Google Voice Input	5,870	12.3	
Youtube	1,400	47.6	52.3

Deep Networks Advance State of Art in Speech

Deep Learning leads to breakthrough in speech recognition at MSR.



Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

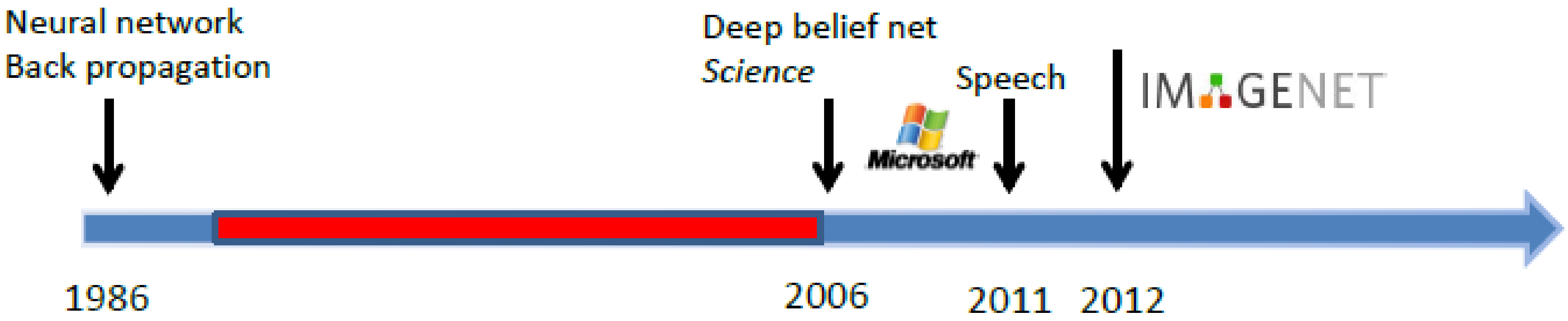
A. Krizhevsky, L. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," NIPS, 2012.

Examples from ImageNet

1000 object classes that we recognize

poster created by Fengjun Lv using VIPBase

Images courtesy of ImageNet (http://www.image-net.org/challenges/LSVRC/2010/index)



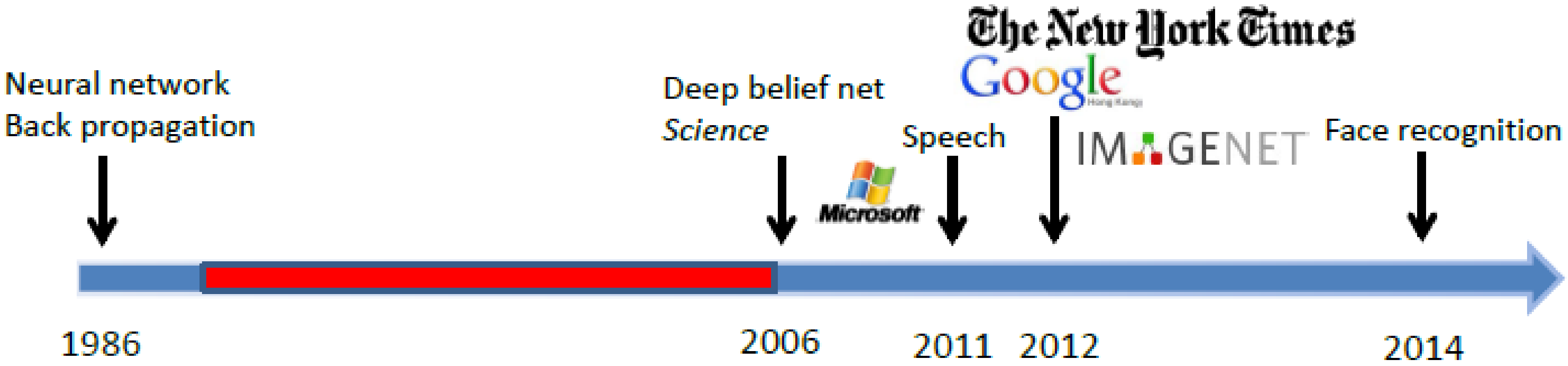
- ImageNet 2013 – image classification challenge

Rank	Name	Error rate	Description
1	NYU	0.11197	Deep learning
2	NUS	0.12535	Deep learning
3	Oxford	0.13555	Deep learning

MSRA, IBM, Adobe, NEC, Clarifai, Berkley, U. Tokyo, UCLA, UIUC, Toronto .... Top 20 groups all used deep learning

- ImageNet 2013 – object detection challenge

Rank	Name	Mean Average Precision	Description
1	UvA-Euvision	0.22581	Hand-crafted features
2	NEC-MU	0.20895	Hand-crafted features
3	NYU	0.19400	Deep learning

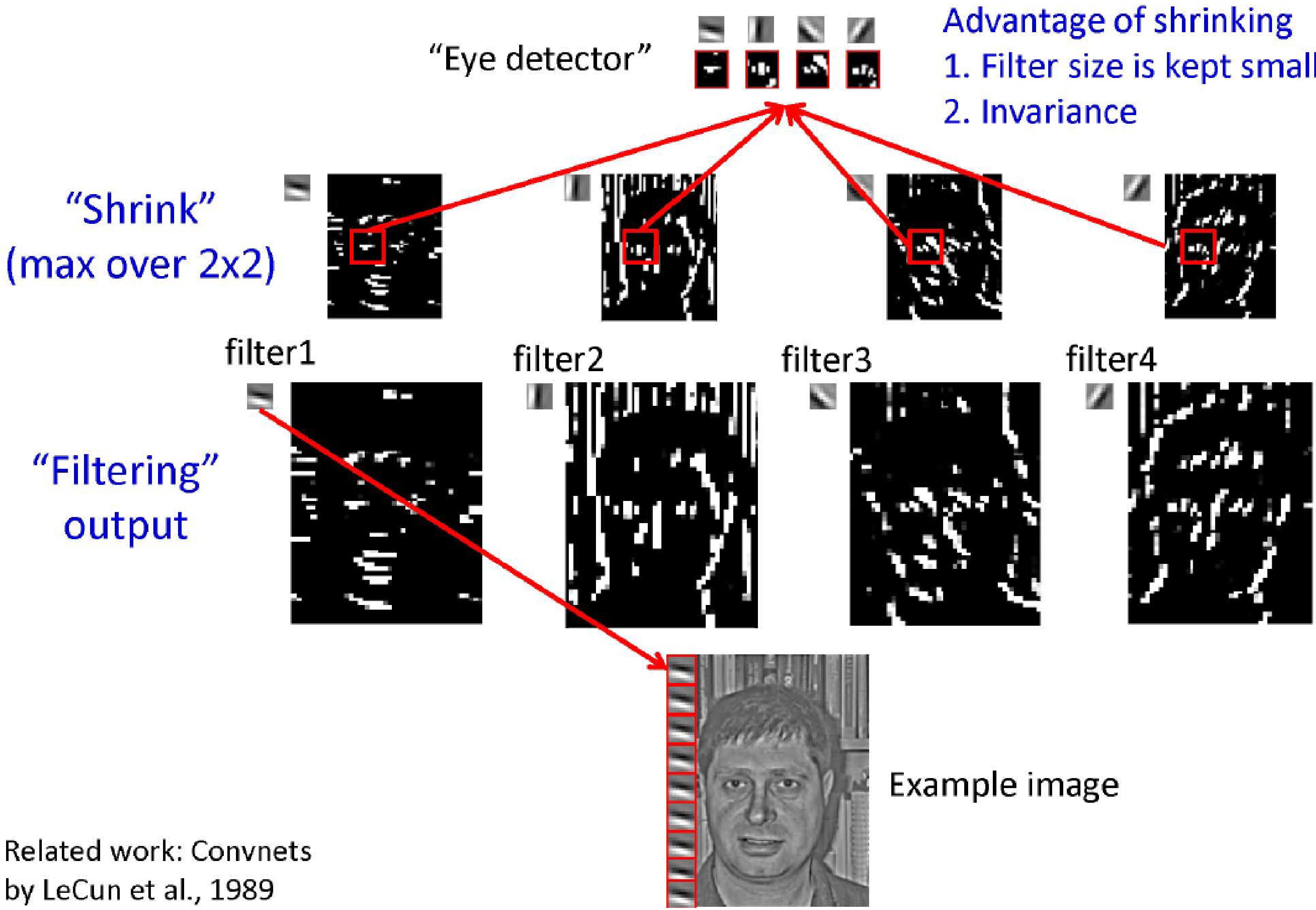


- Deep learning achieves 99.53% face verification accuracy on Labeled Faces in the Wild (LFW), higher than human performance

Y. Sun, X. Wang, and X. Tang. Deep Learning Face Representation by Joint Identification-Verification. NIPS, 2014.

Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. CVPR, 2015.

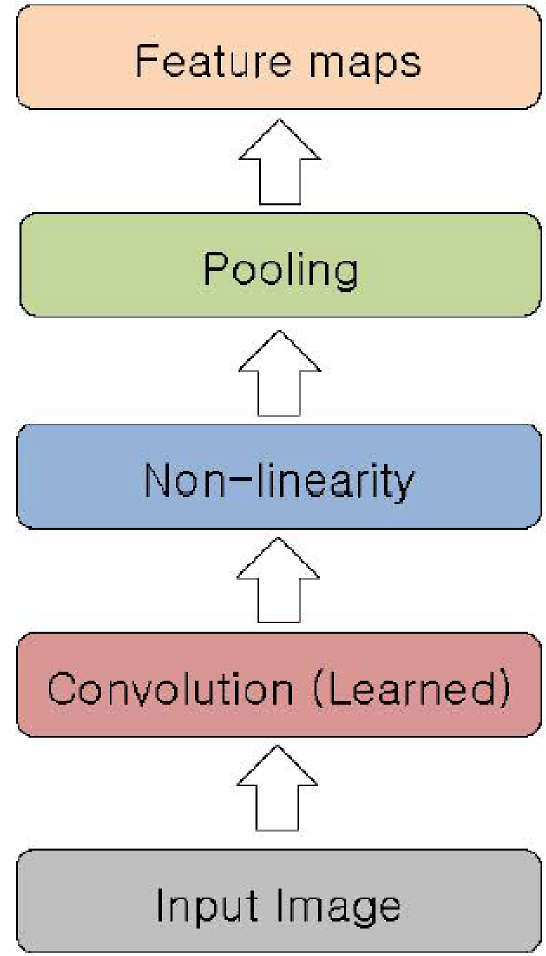
Illustration: Learning an "eye" detector



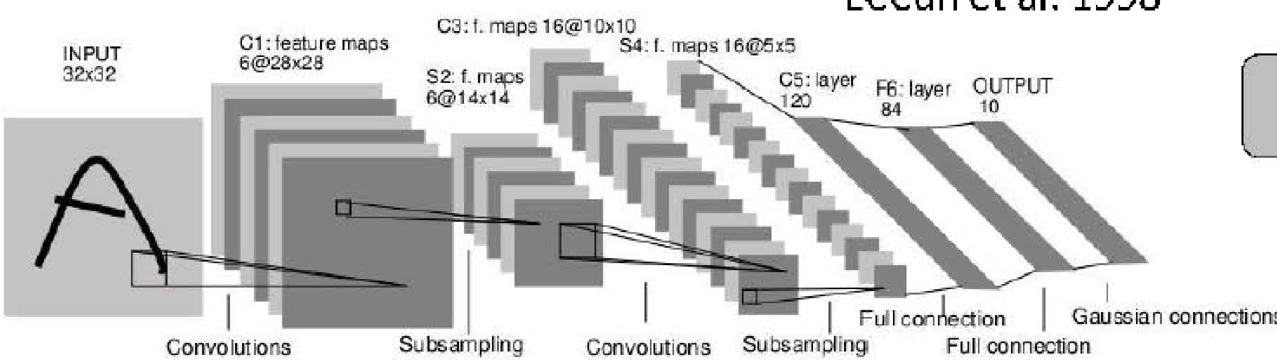
Related work: Convnets
by LeCun et al., 1989

Convolutional Neural Networks

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



LeCun et al. 1998

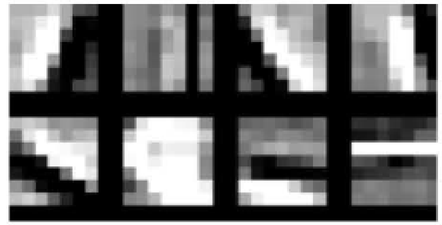


Slide: R. Fergus

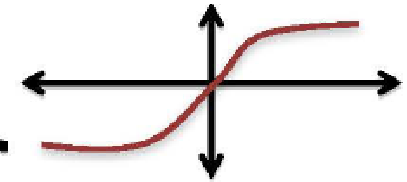
Components of Each Layer

Pixels / Features

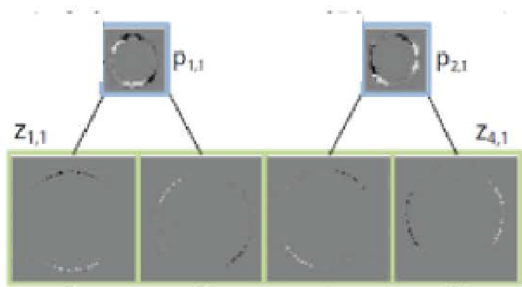
Filter with Dictionary
(convolutional or tiled)



+ Non-linearity



Spatial/Feature
(Sum or Max)



[Optional]

Normalization between feature responses

Output Features

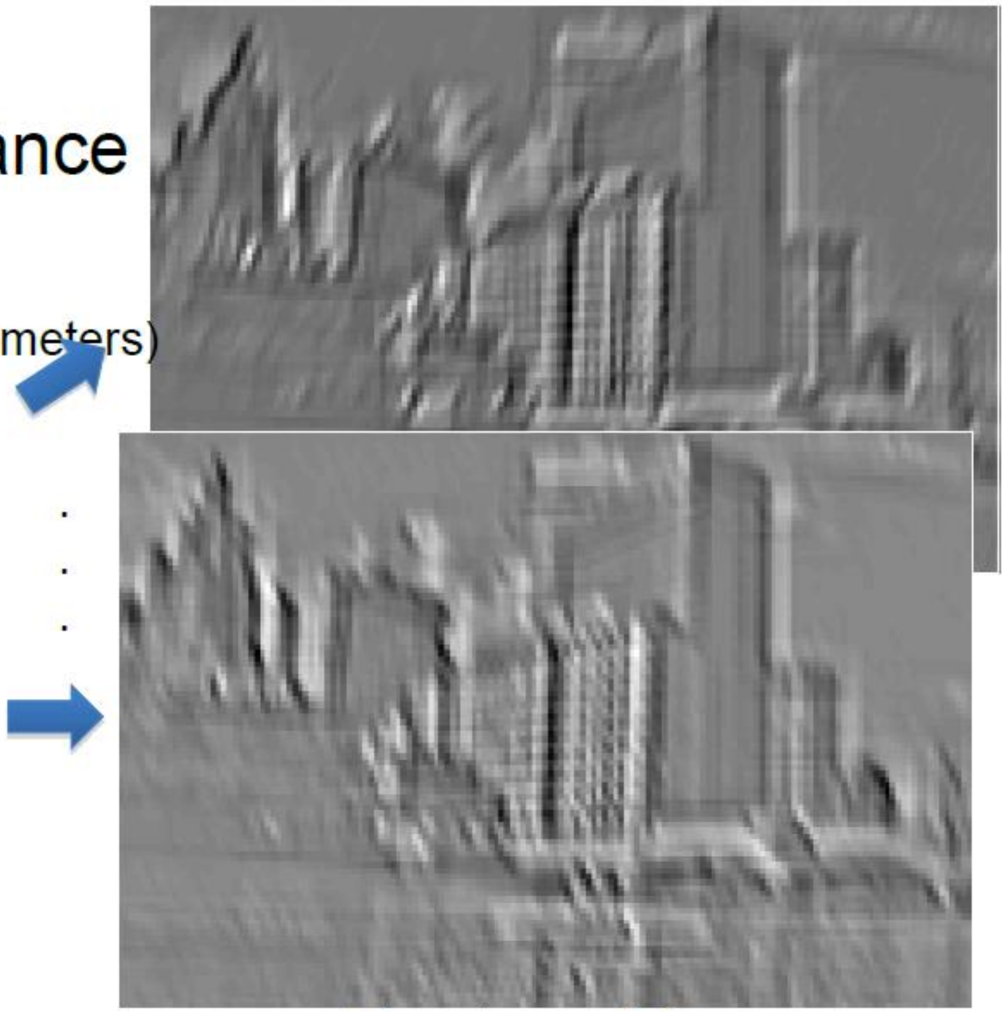
Slide: R. Fergus

Filtering

- Convolutional
  - Translation equivariance
  - Tied filter weights

  (same at each position → few parameters)

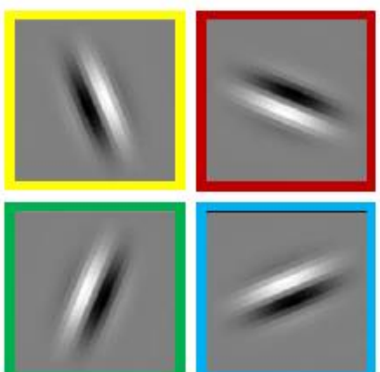
Input



Feature Map

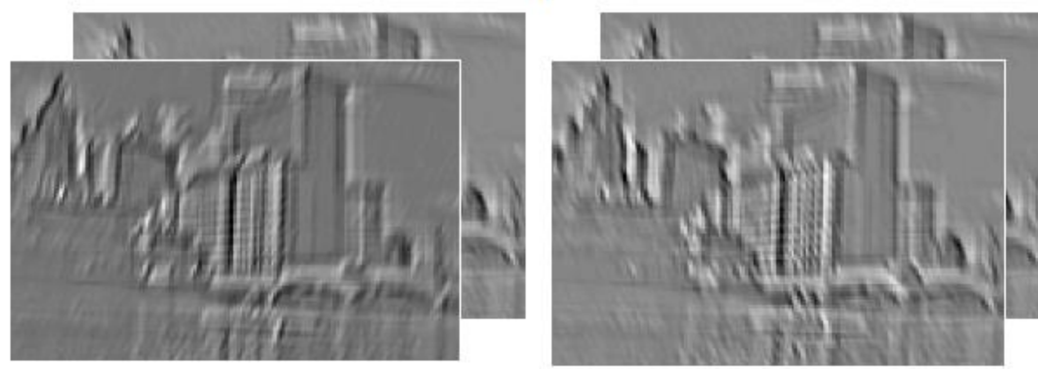
Filtering

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



Filters

Input

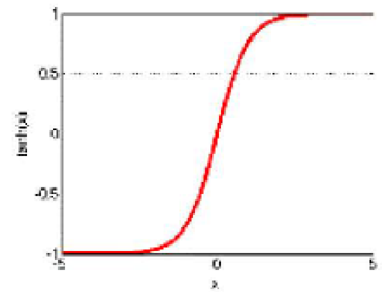


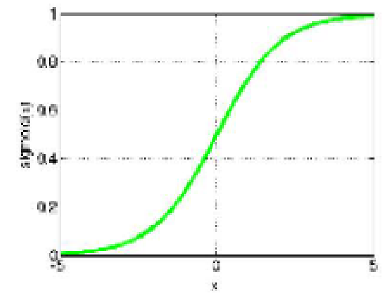
Feature maps

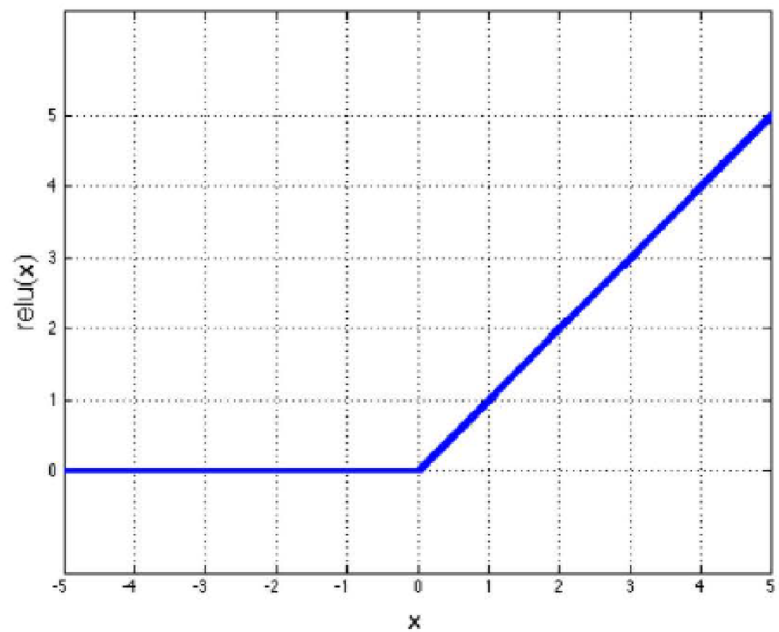
Non-Linearity

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

    → Preferred option



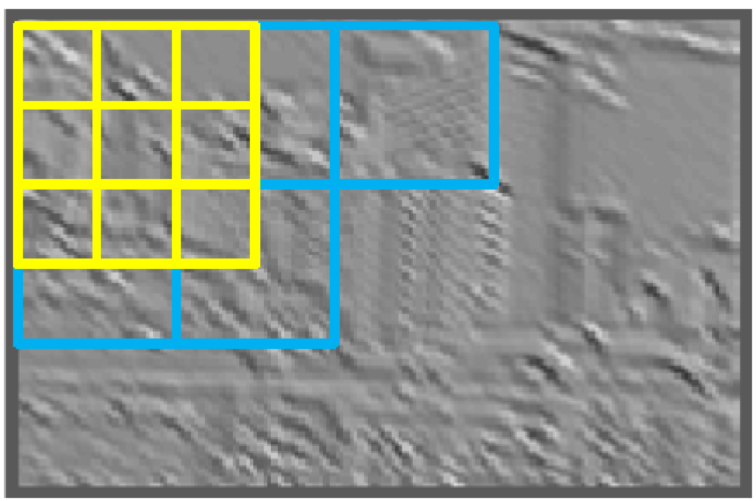




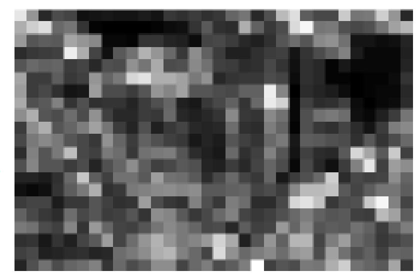
Slide: R. Fergus

Pooling

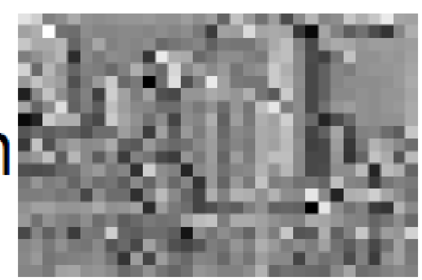
- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis



Max



Sum



2D Convolution

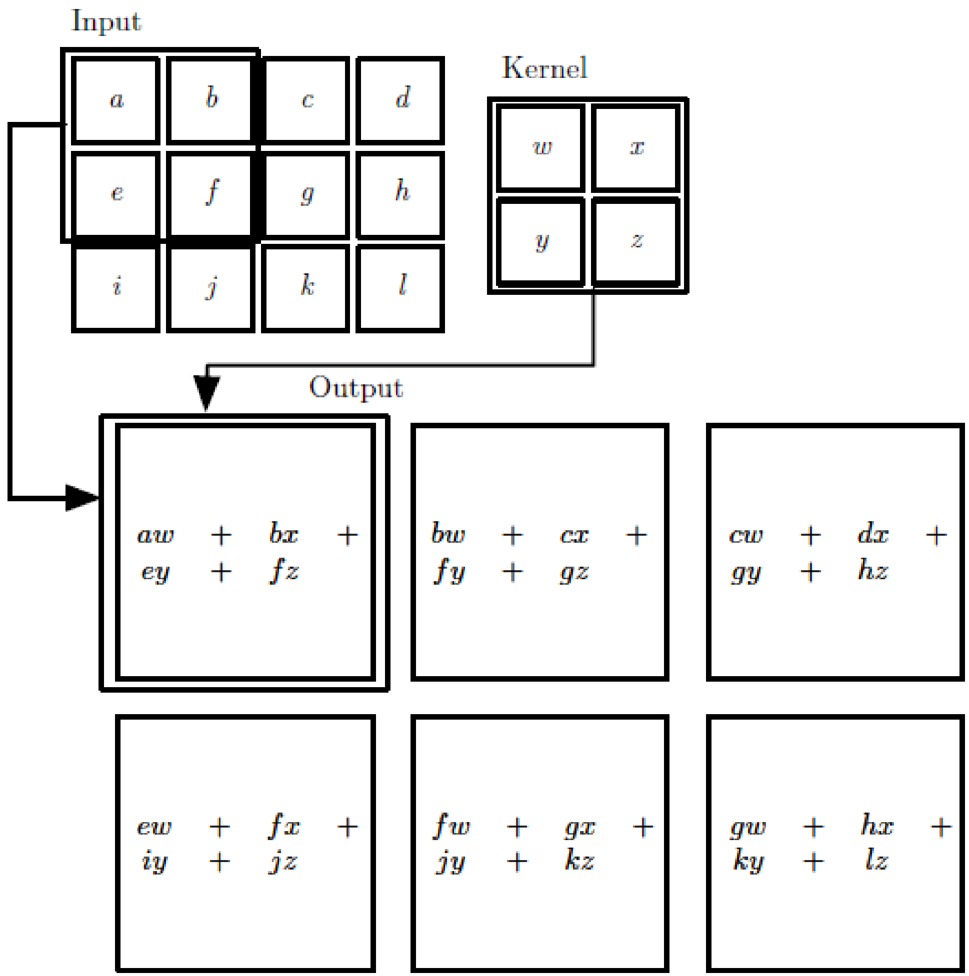


Figure 9.1

(Goodfellow 2016)

Three Operations

- Convolution: like matrix multiplication
  - Take an input, produce an output (hidden layer)
- "Deconvolution": like multiplication by transpose of a matrix
  - Used to back-propagate error from output to input
  - Reconstruction in autoencoder / RBM
- Weight gradient computation
  - Used to backpropagate error from output to weights
  - Accounts for the parameter sharing

Sparse Connectivity

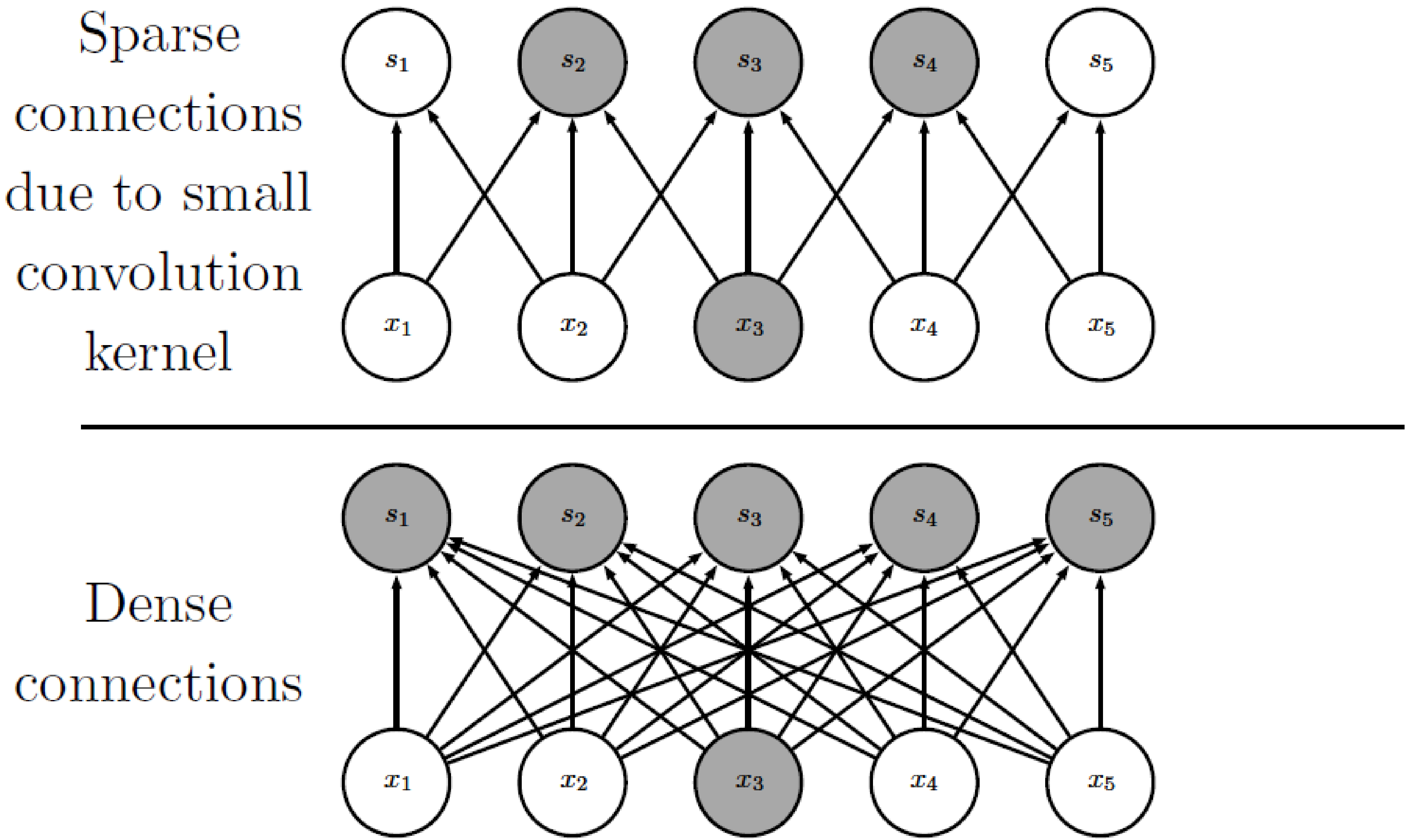


Figure 9.2

(Goodfellow 2016)

Sparse Connectivity

Sparse connections due to small convolution kernel

Dense connections

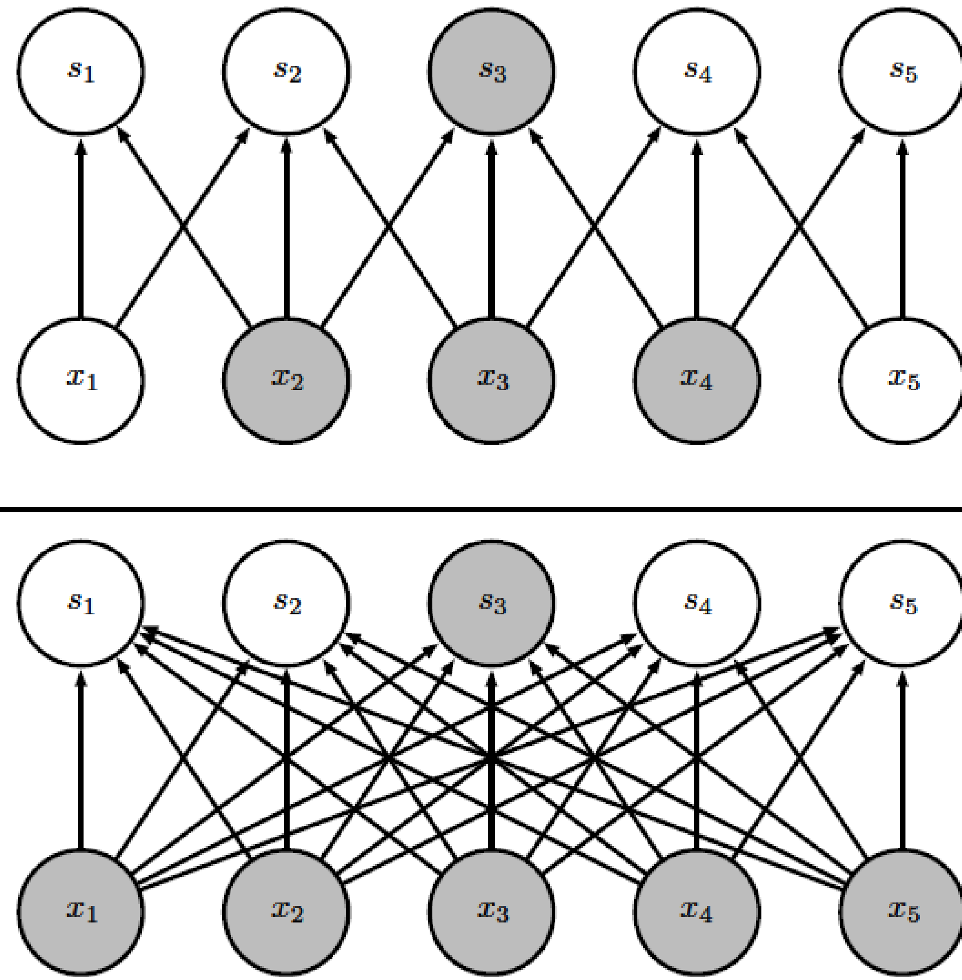


Figure 9.3

(Goodfellow 2016)

Growing Receptive Fields

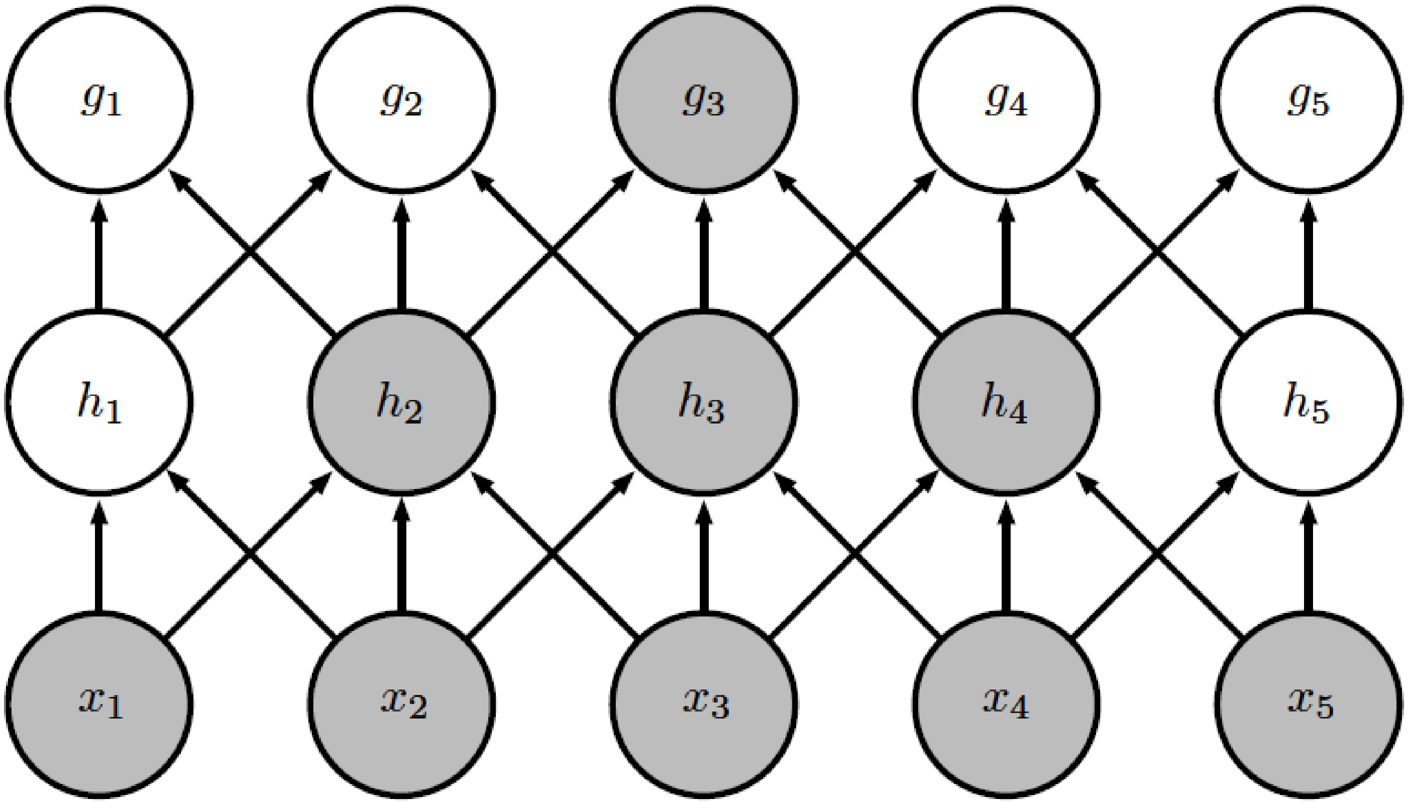


Figure 9.4

Parameter Sharing

Convolution shares the same parameters across all spatial locations

Traditional matrix multiplication does not share any parameters

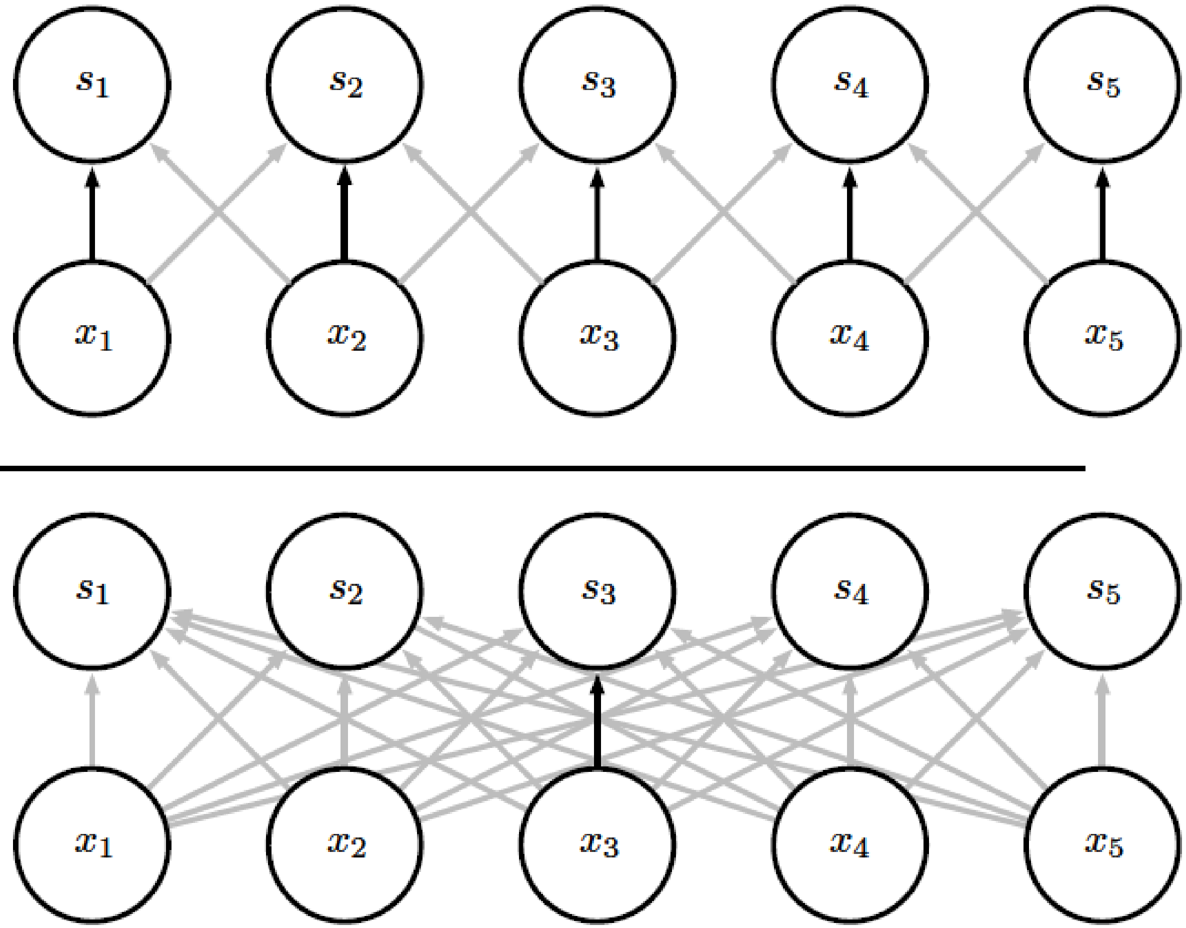


Figure 9.5

(Goodfellow 2016)

Edge Detection by Convolution

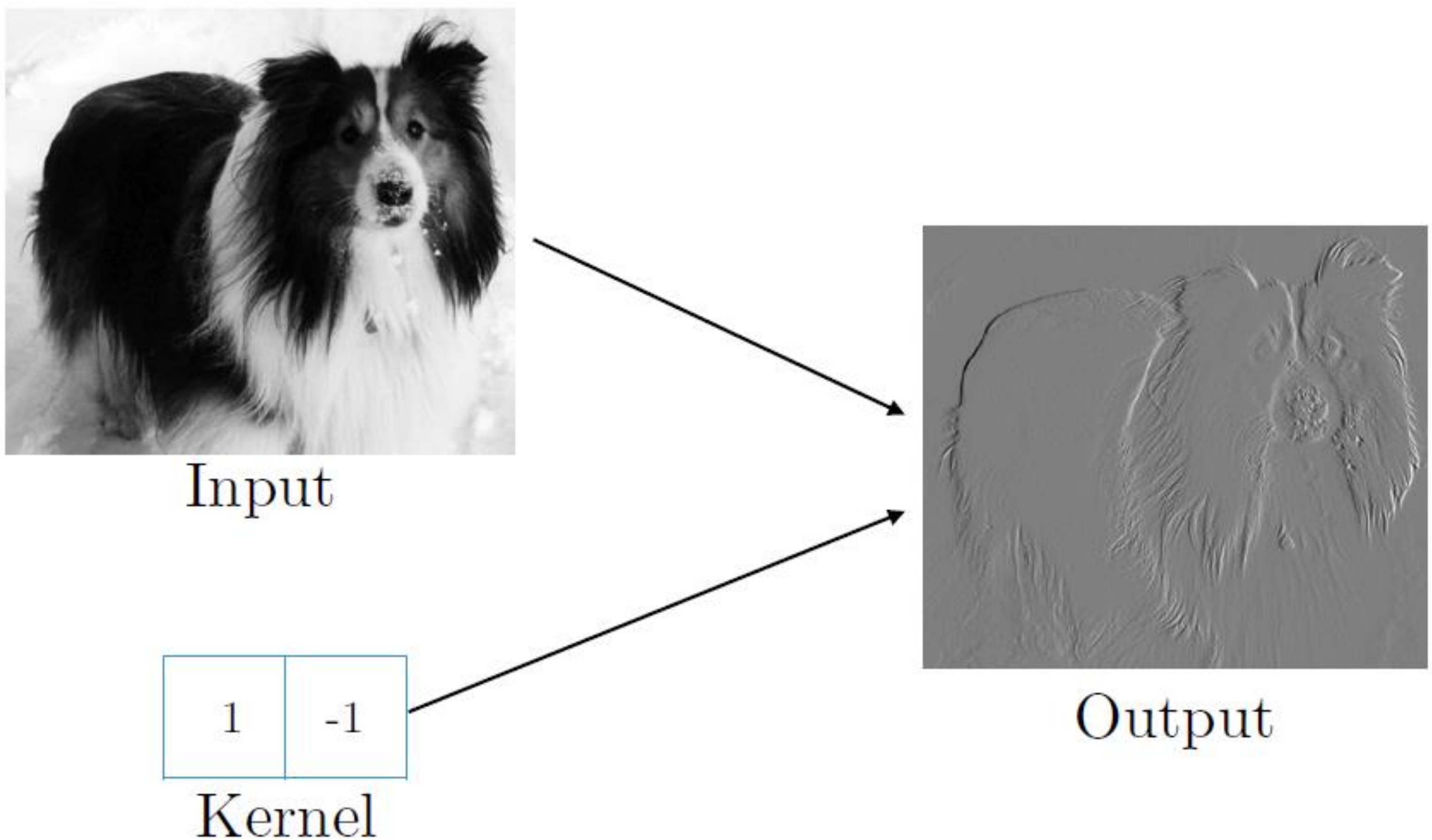


Figure 9.6

(Goodfellow 2016)

Efficiency of Convolution

Input size: 320 by 280

Kernel size: 2 by 1

Output size: 319 by 280

	Convolution	Dense matrix	Sparse matrix
Stored floats	2	319*280*320*280 > 8e9	2*319*280 = 178,640
Float muls or adds	319*280*3 = 267,960	> 16e9	Same as convolution (267,960)

Convolutional Network Components

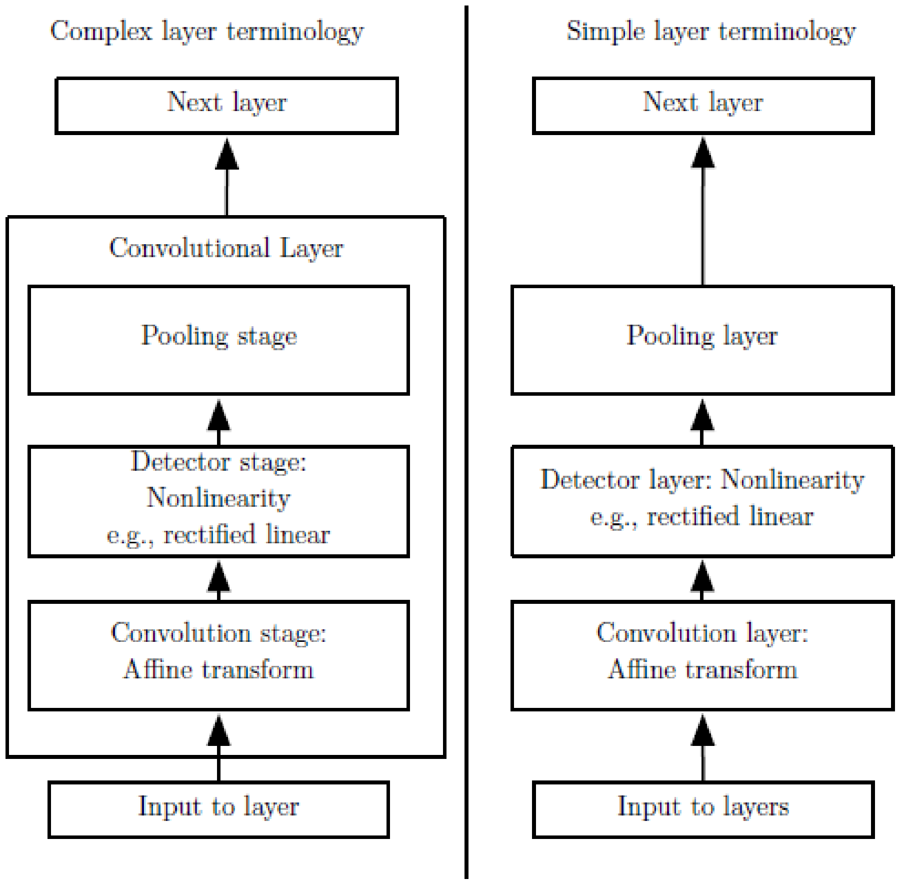


Figure 9.7

(Goodfellow 2016)

Max Pooling and Invariance to Translation

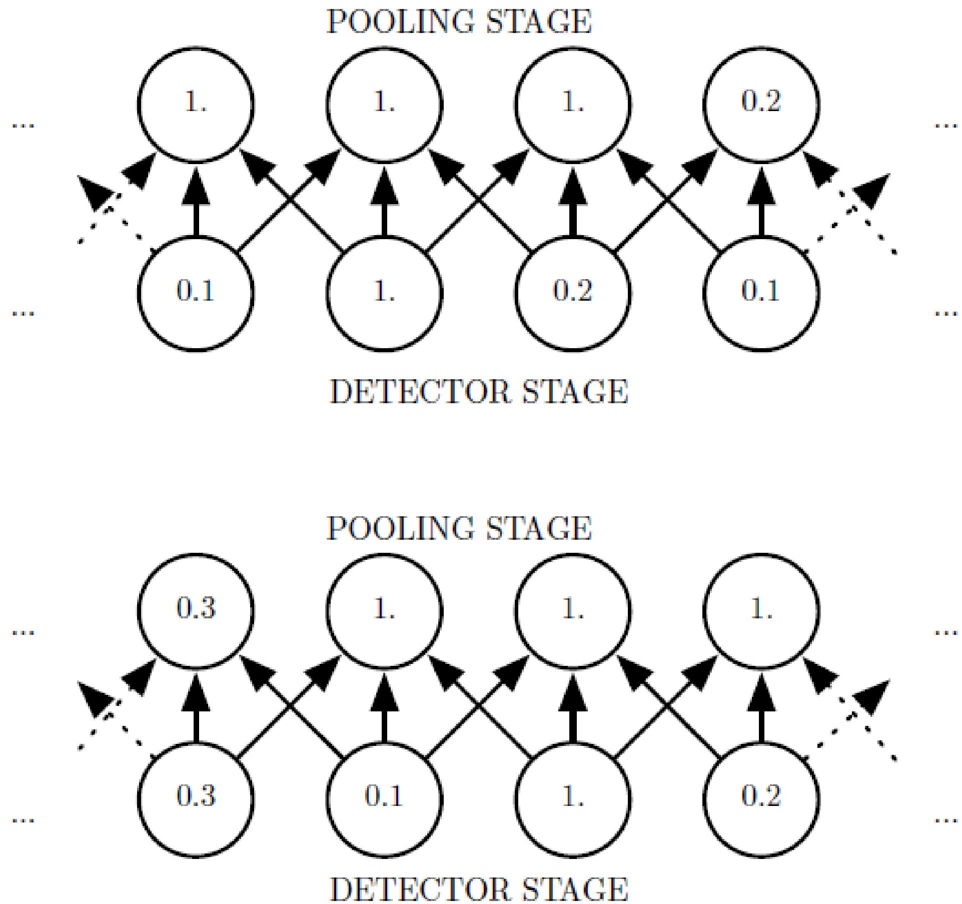


Figure 9.8

(Goodfellow 2016)

Cross-Channel Pooling and Invariance to Learned Transformations

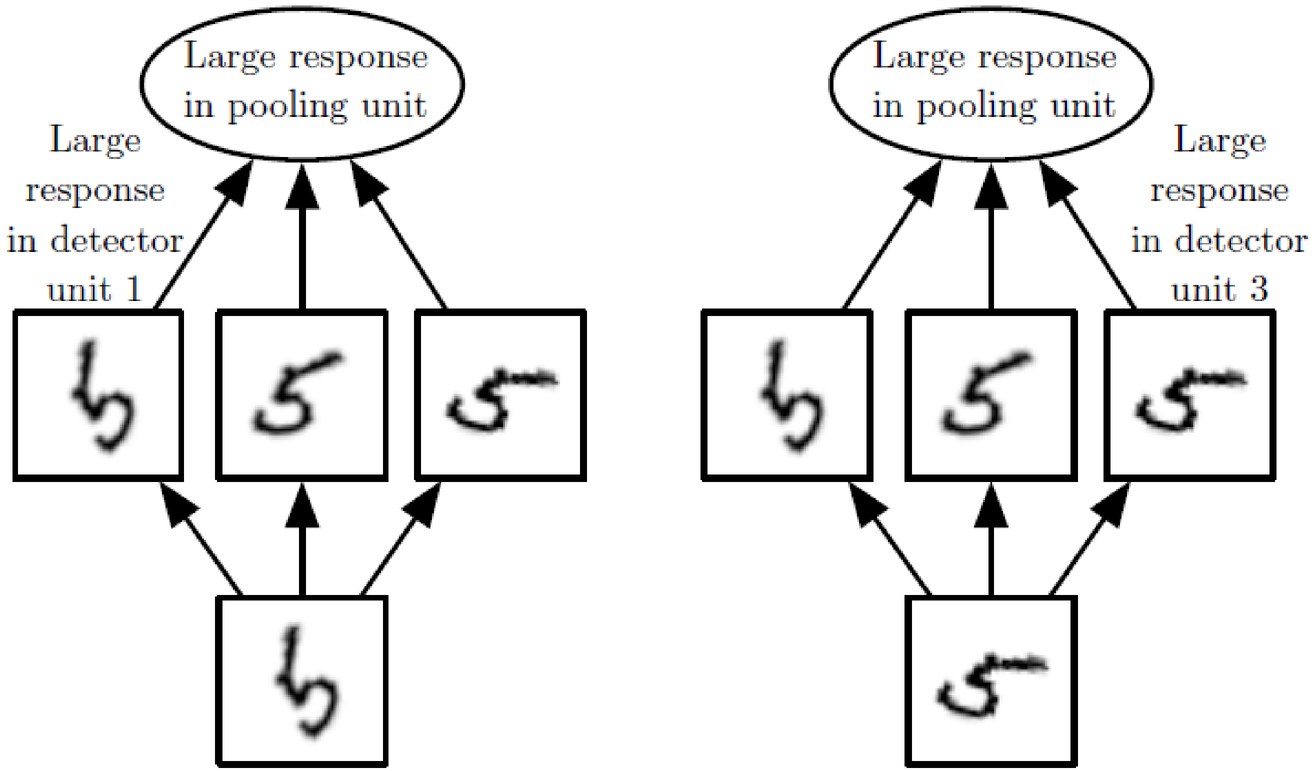


Figure 9.9

(Goodfellow 2016)

Pooling with Downsampling

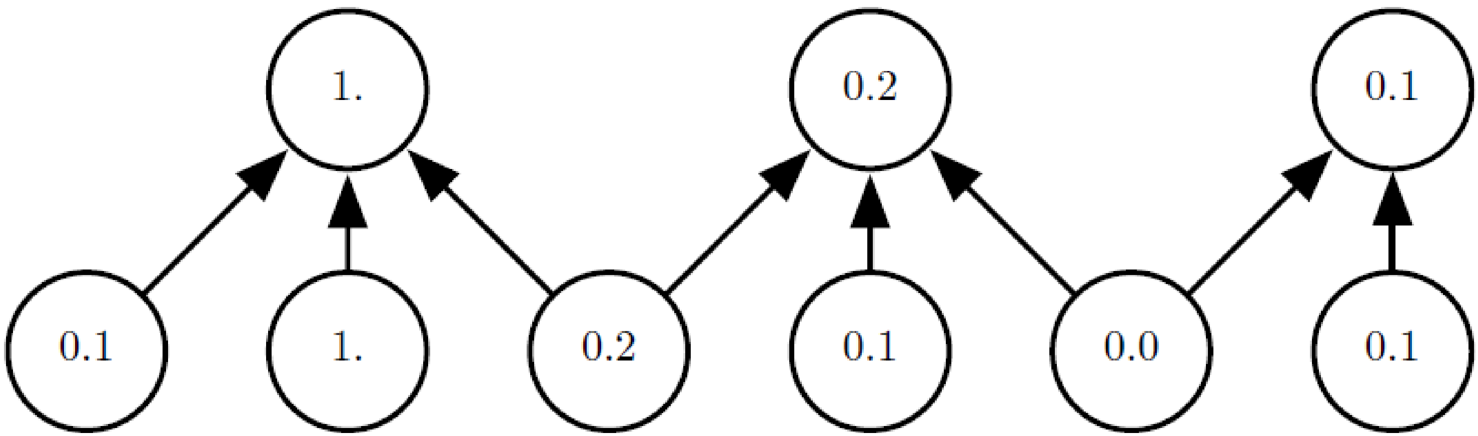


Figure 9.10

(Goodfellow 2016)

Example Classification Architectures

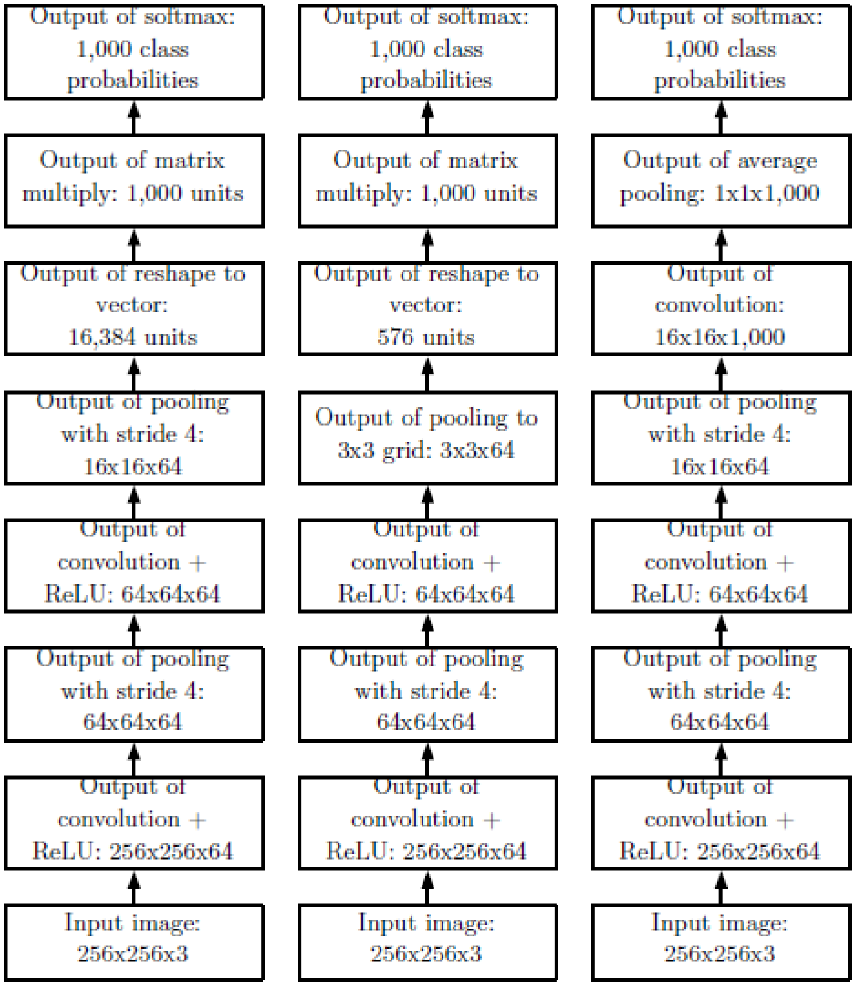


Figure 9.11

(Goodfellow 2016)

Convolution with Stride

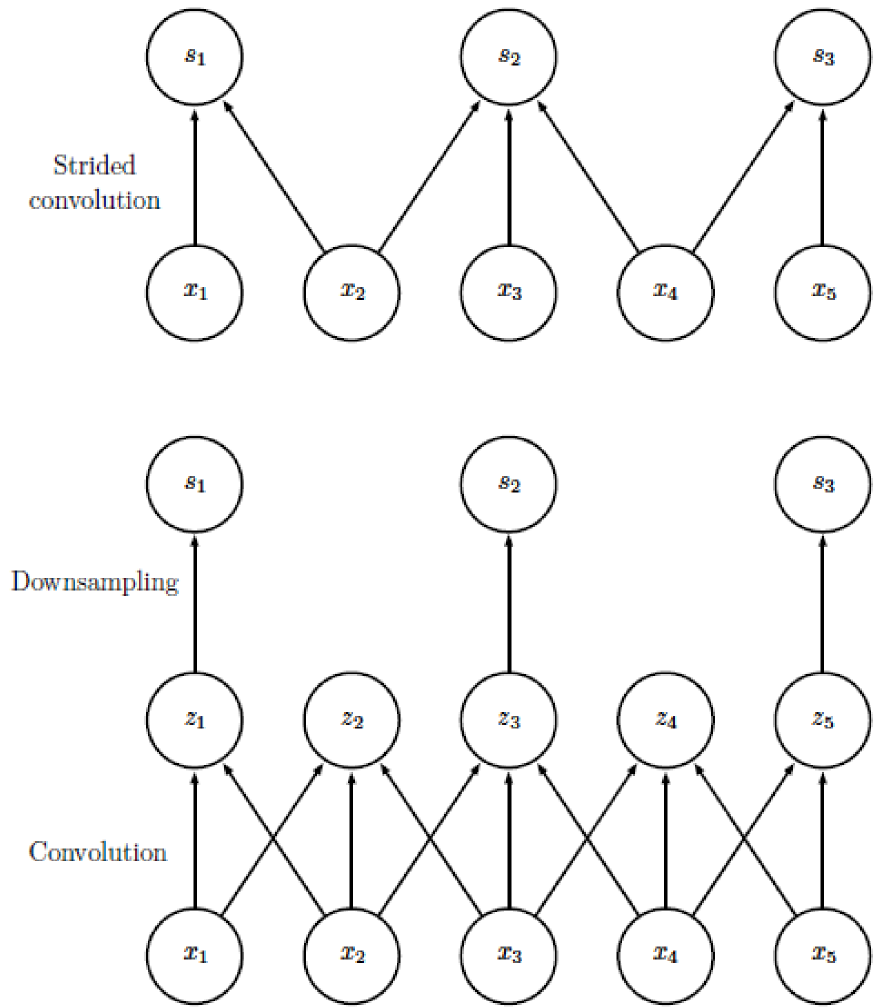


Figure 9.12

(Goodfellow 2016)

Zero Padding Controls Size

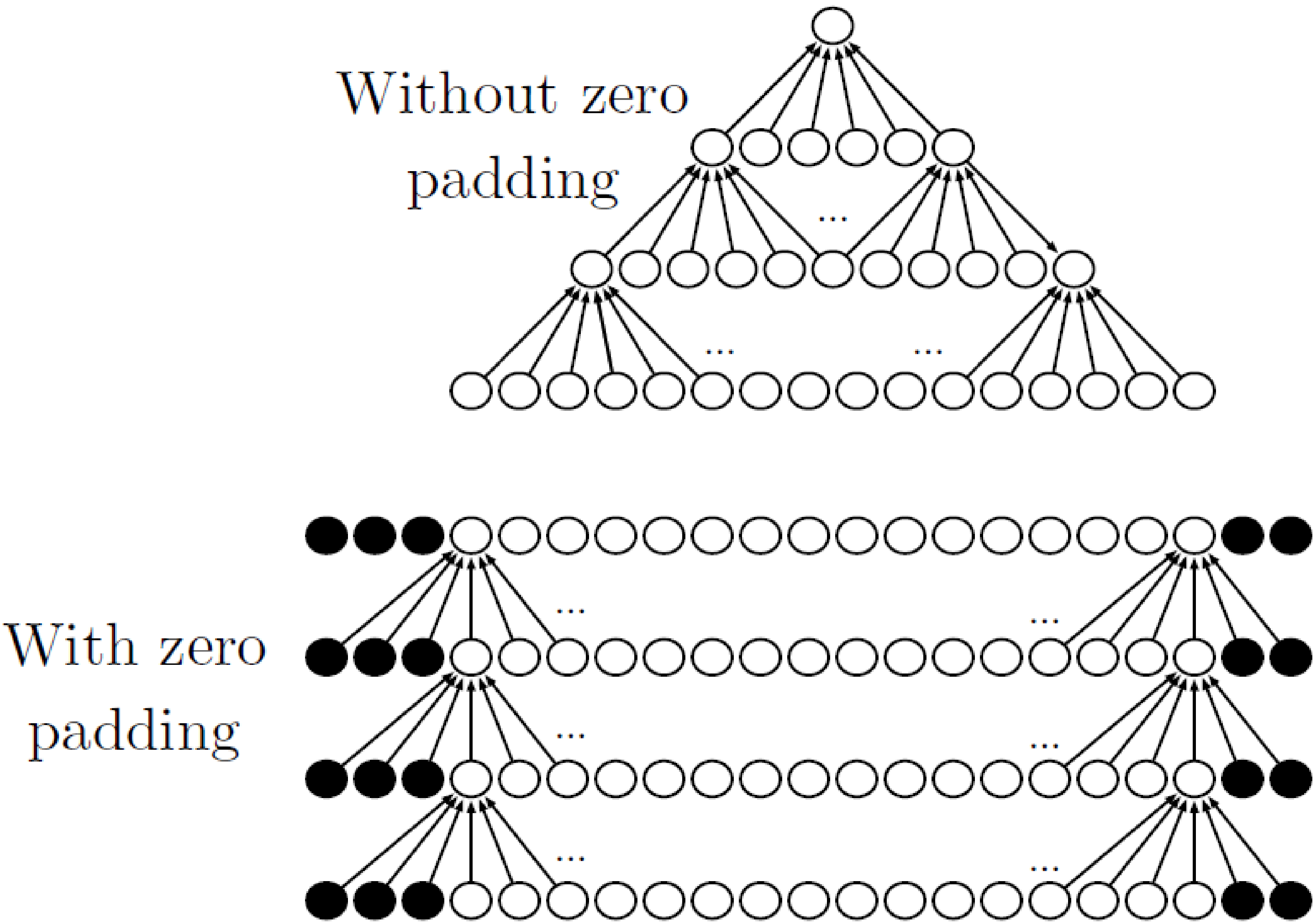
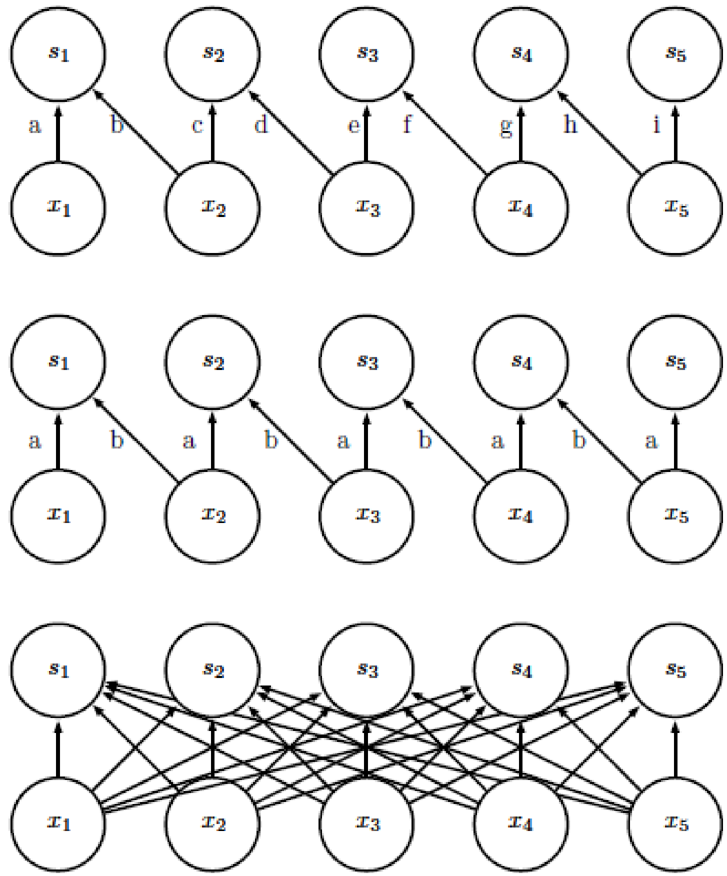


Figure 9.13

(Goodfellow 2016)

Kinds of Connectivity



Local connection: like convolution, but no sharing

Convolution

Fully connected

Figure 9.14

(Goodfellow 2016)

Partial Connectivity Between Channels

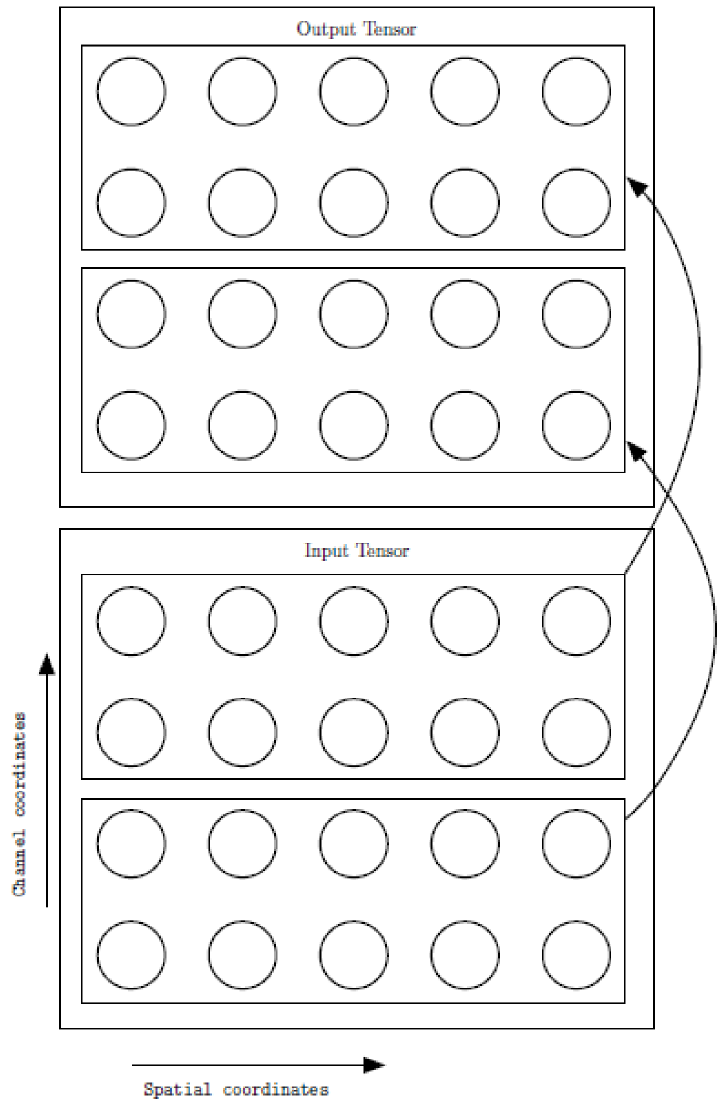
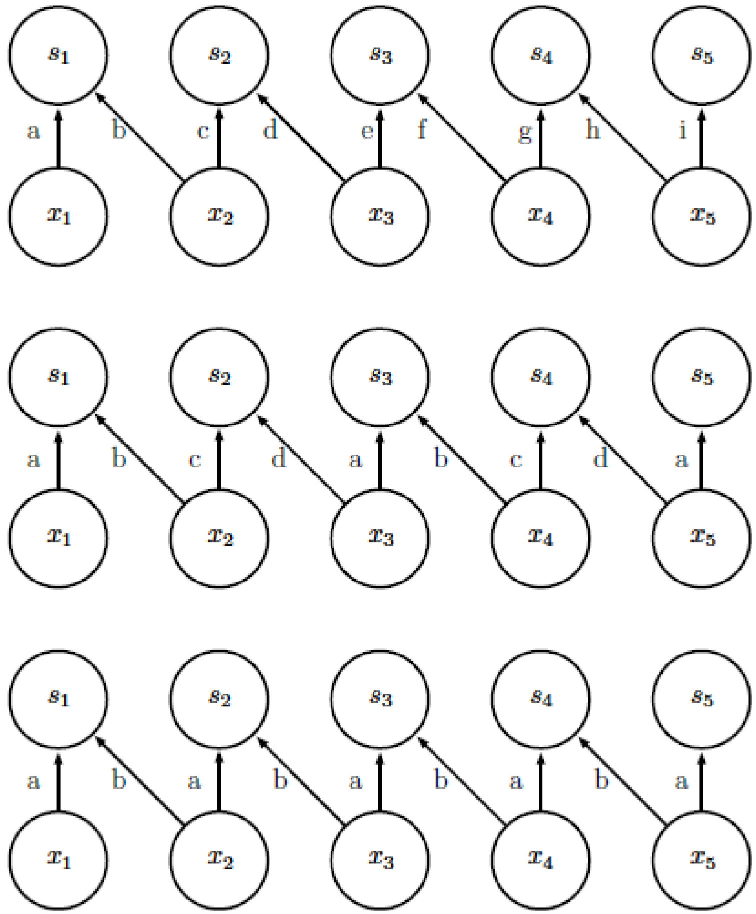


Figure 9.15

(Goodfellow 2016)

Tiled convolution



Local connection (no sharing)

Tiled convolution (cycle between groups of shared parameters)

Convolution (one group shared everywhere)

Figure 9.16

(Goodfellow 2016)

Local and Global Representations

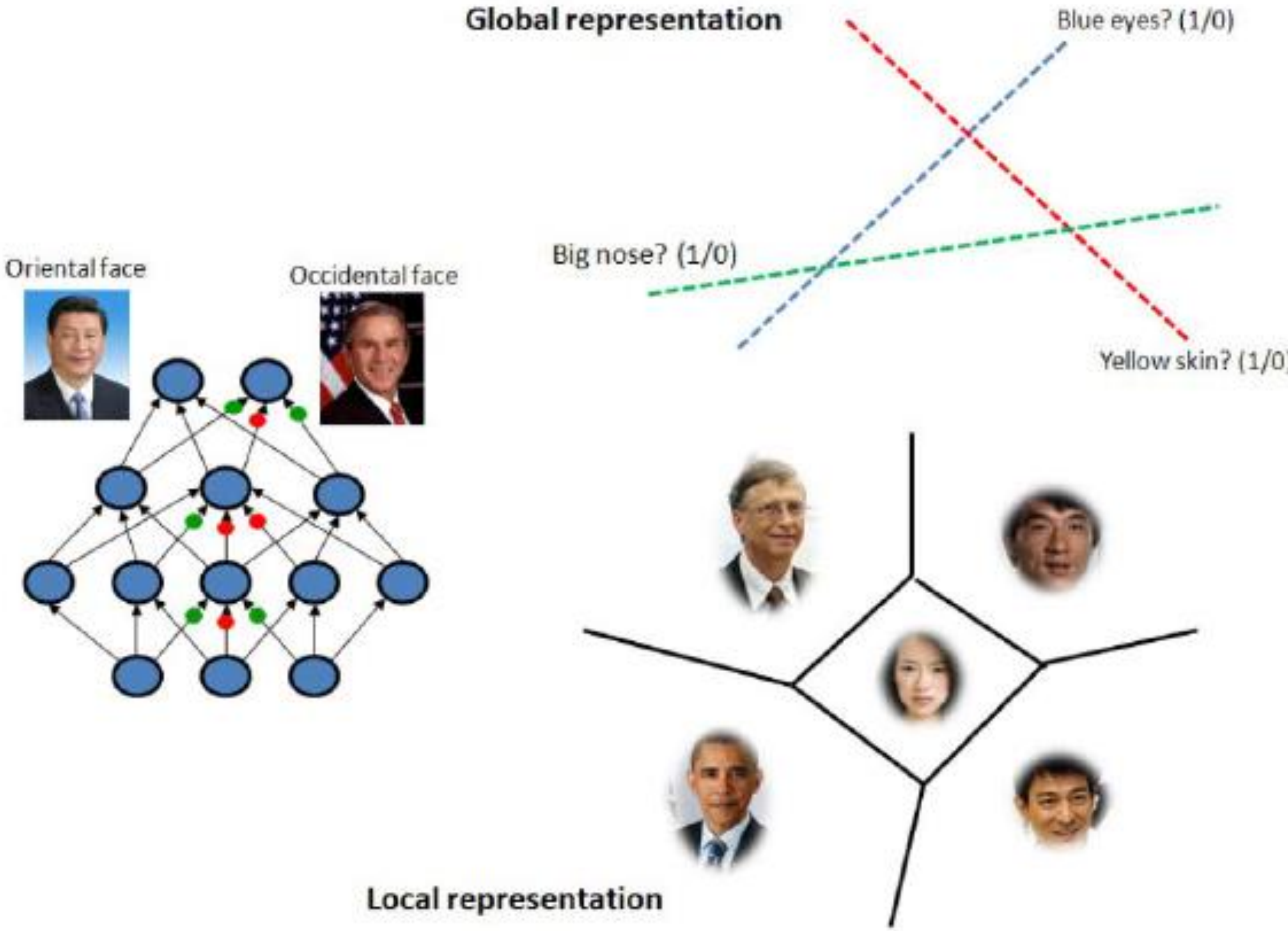


Image classification result