Scene Segmentation and Interpretation

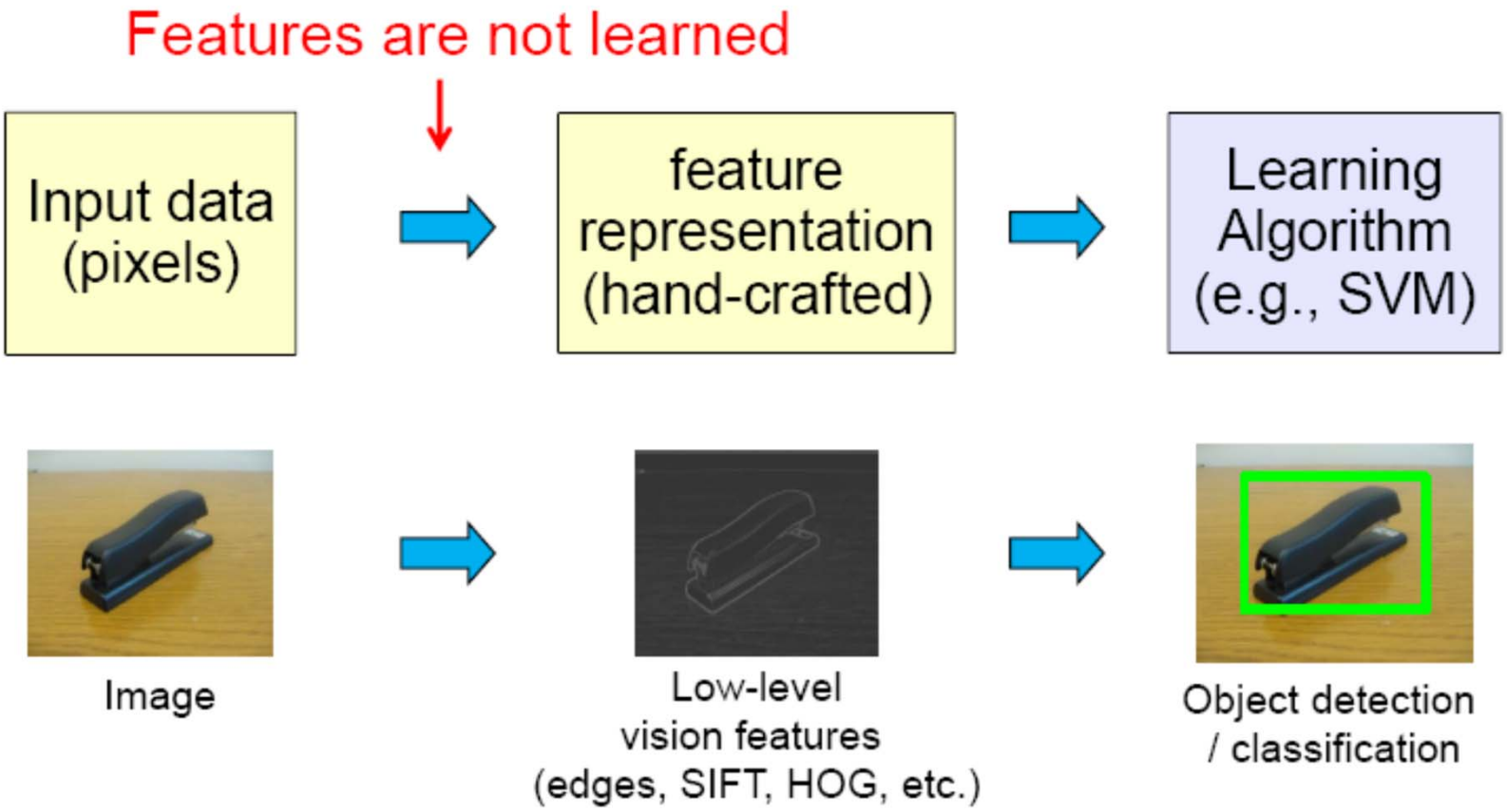
Xavier Lladó (llado@eia.udg.edu)
Arnau Oliver (aoliver@eia.udg.edu)

VICOROB – Universitat de Girona

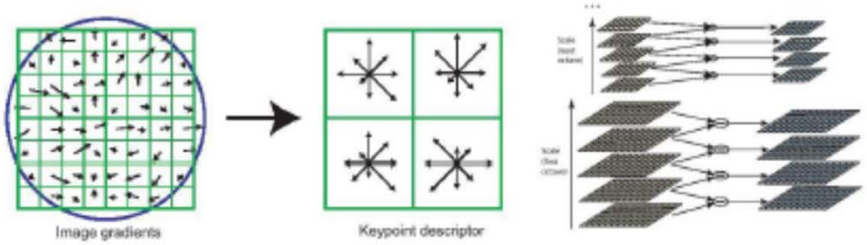
Deep Learning

Convolutional Neural Networks (CNNs)

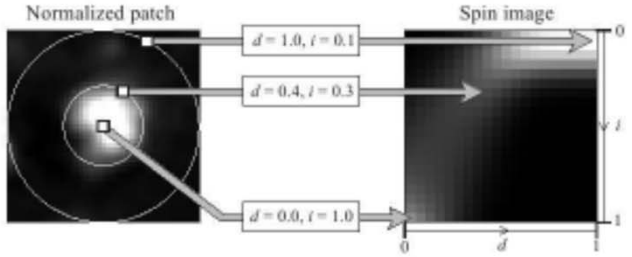
Traditional Recognition Approach



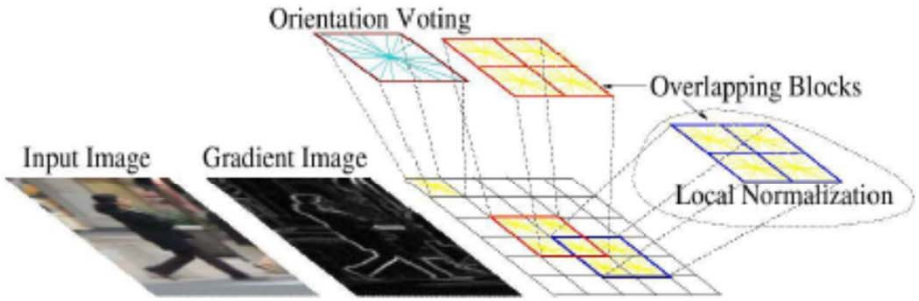
Computer vision features



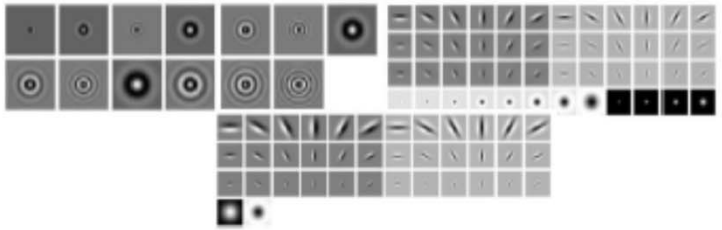
SIFT



Spin image



HoG



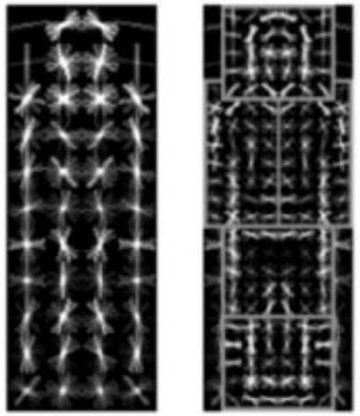
Textons

and many others:

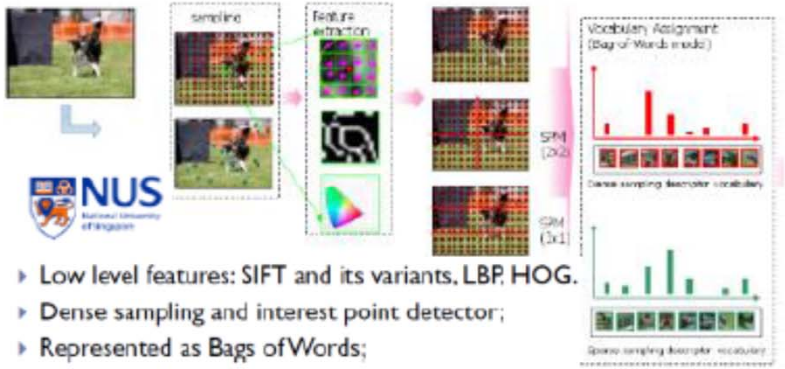
SURF, MSER, LBP, Color-SIFT, Color histogram, GLOH, …..

Motivation

- Features are key to recent progress in recognition
- Multitude of hand-designed features currently in use
- Where next? Better classifiers? building better features?



Felzenszwalb, Girshick,
McAllester and Ramanan, PAMI 2007

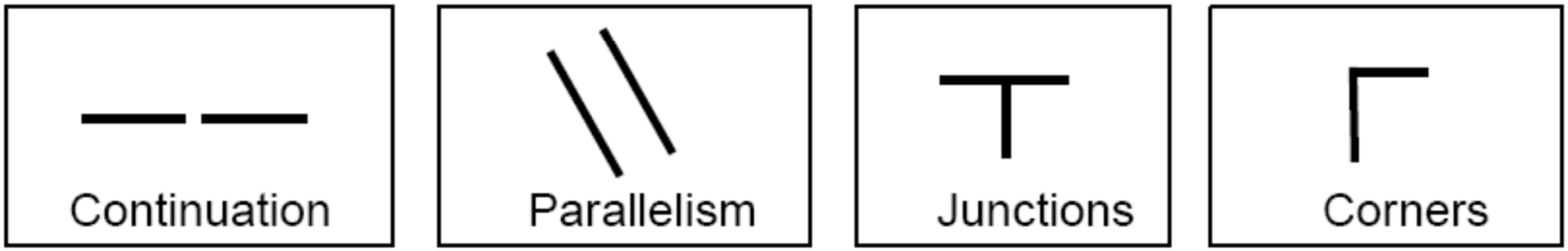


Yan & Huang
(Winner of PASCAL 2010 classification competition)

Slide: R. Fergus

Mid-Level Representations

- Mid-level cues



"Tokens" from Vision by D.Marr:

- Object parts:

- Difficult to hand-engineer → What about learning them?

Slide: R. Fergus

Learning Feature Hierarchy

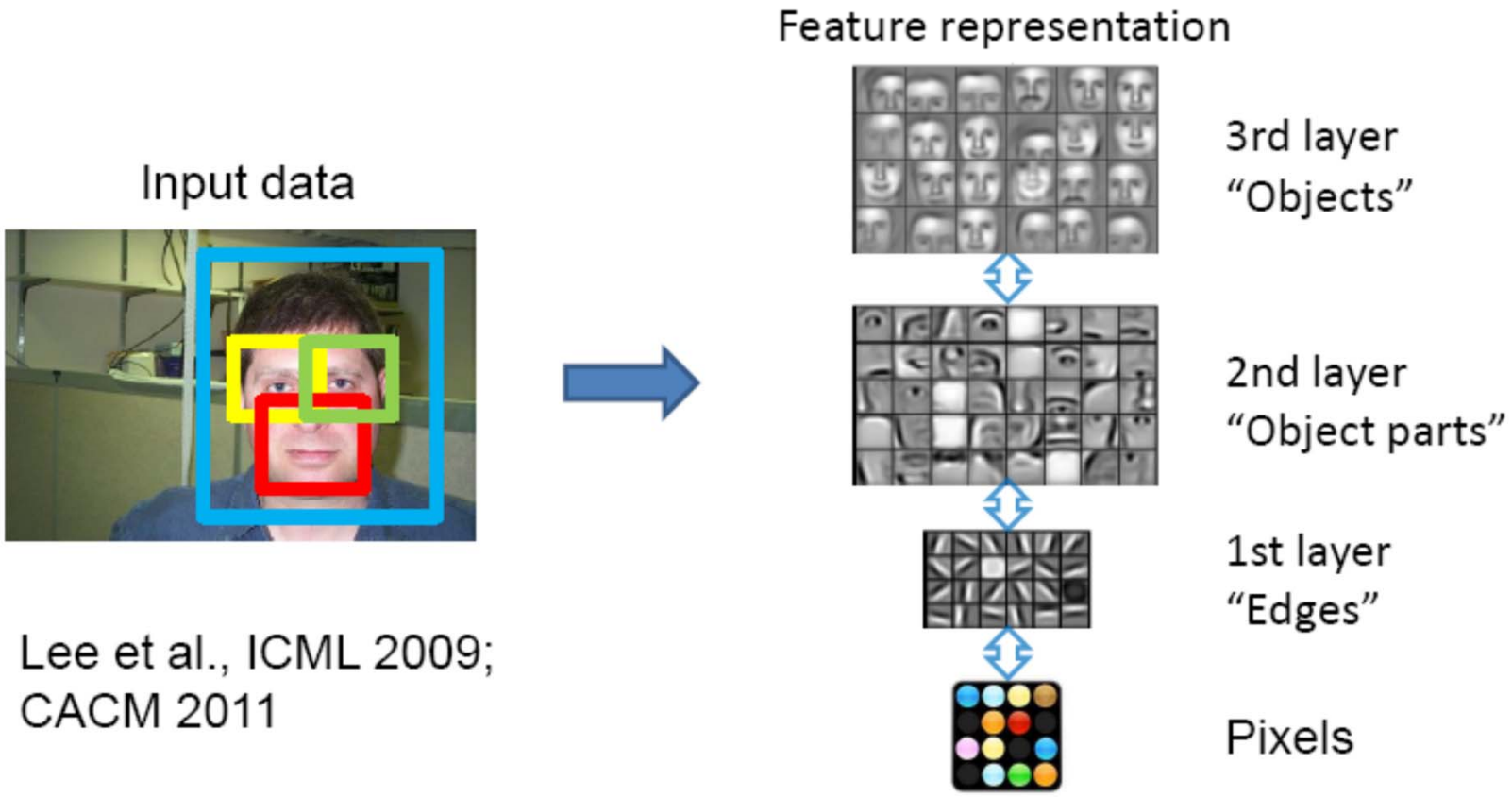
- Learn hierarchy
- All the way from pixels → classifier
- One layer extracts features from output of previous layer

- Train all layers jointly

Slide: R. Fergus

Learning Feature Hierarchy

1. Learn **useful higher-level features** from images



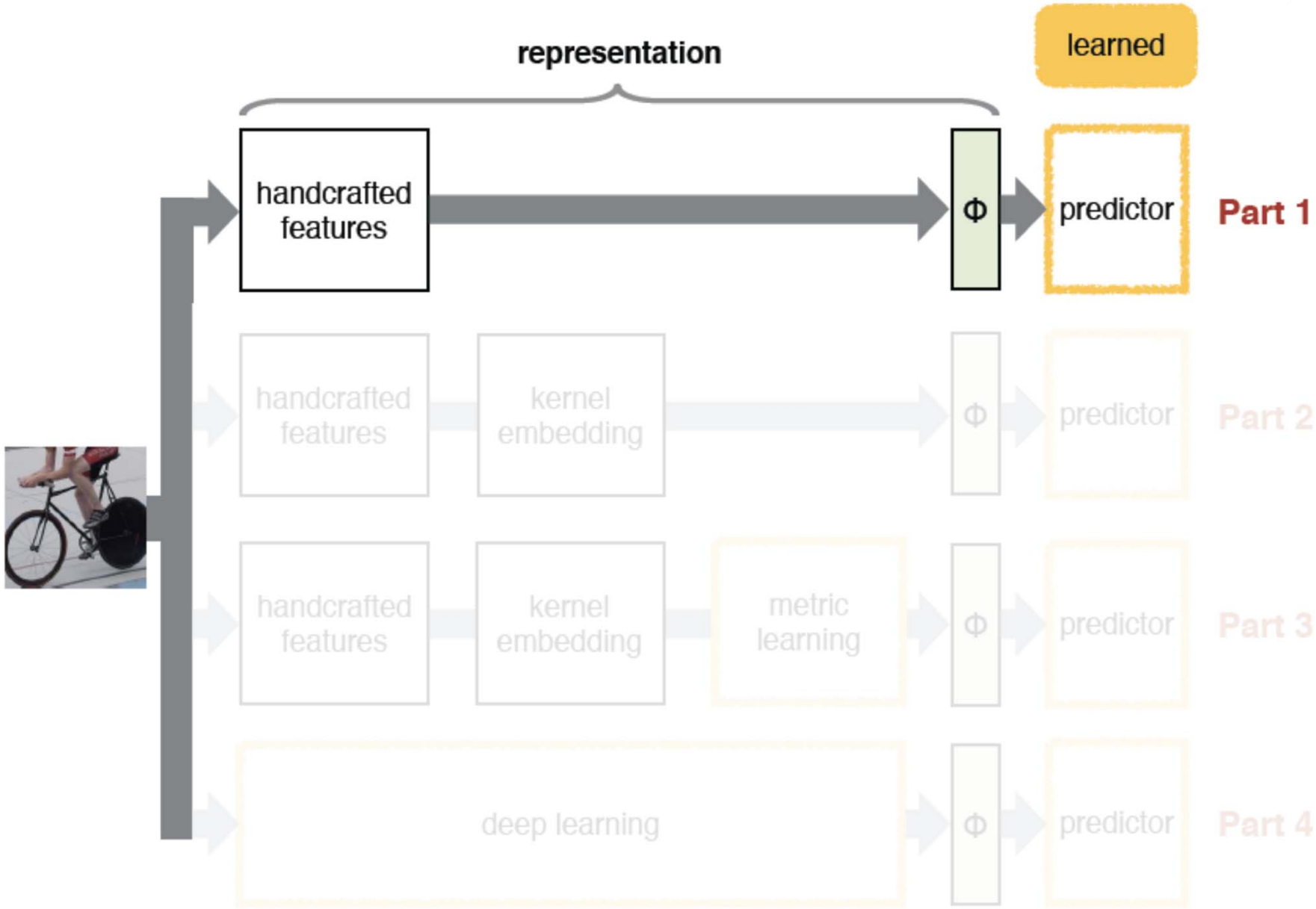
Lee et al., ICML 2009;
CACM 2011

2. **Fill in representation gap** in recognition

Approaches to learning features

- Supervised Learning
  - End-to-end learning of deep architectures (e.g., deep neural networks) with back-propagation
  - Works well when the amounts of labels is large
  - Structure of the model is important (e.g. convolutional structure)

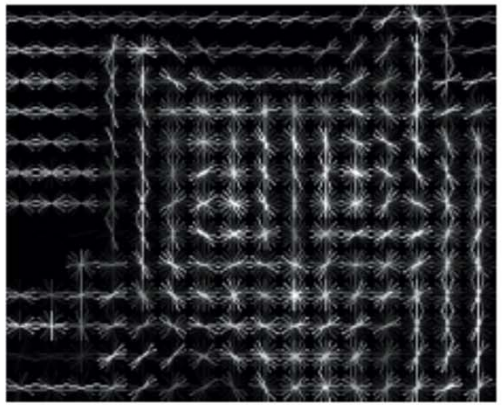
- Unsupervised Learning
  - Learn statistical structure or dependencies of the data from unlabeled data
  - Layer-wise training
  - Useful when the amount of labels is not large

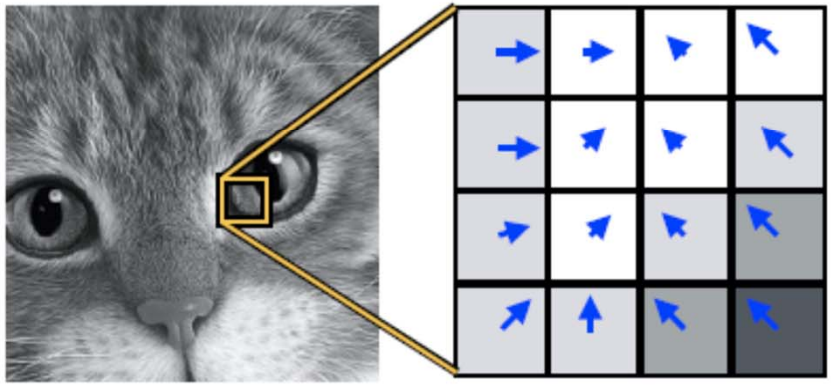

representation
learned
handcrafted features
Φ
predictor
Part 1
handcrafted features
kernel embedding
Φ
predictor
Part 2
handcrafted features
kernel embedding
metric learning
Φ
predictor
Part 3
deep learning
Φ
predictor
Part 4

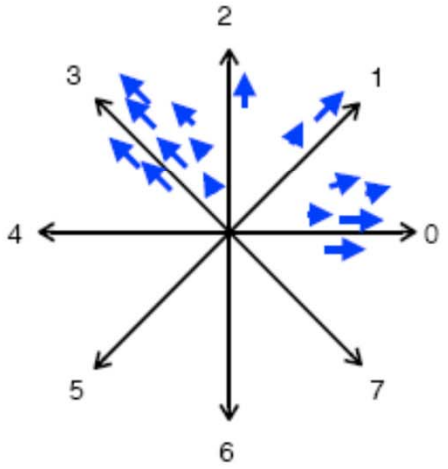
Histogram of Oriented Gradients

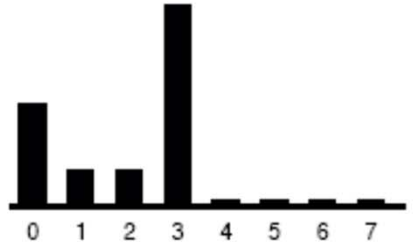
[Lowe 1999, Dalal & Triggs 2005]

HOG captures the local gradient (edge) orientations in the image





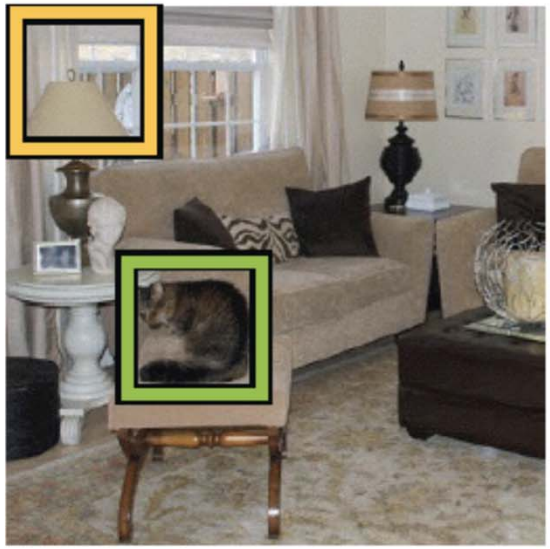


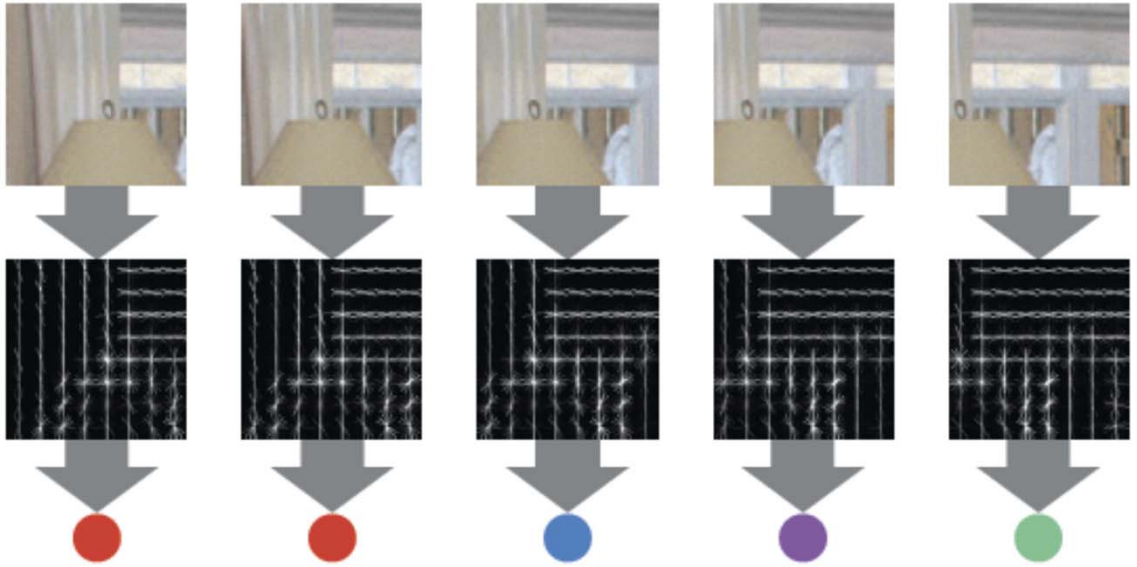


+ block l^2 normalisation

Bag of visual words

[Sivic & Zisserman 2003, Csurka *et al.* 2004, Nowak *et al.* 2006]

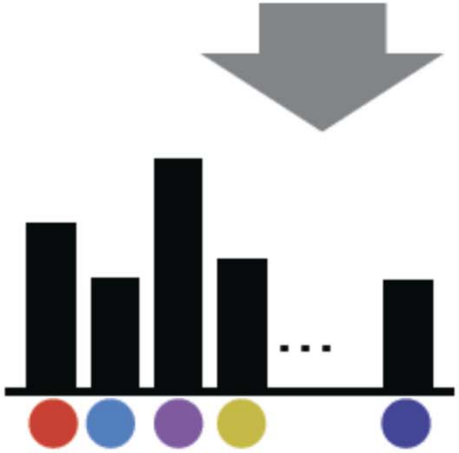




BoVW construction

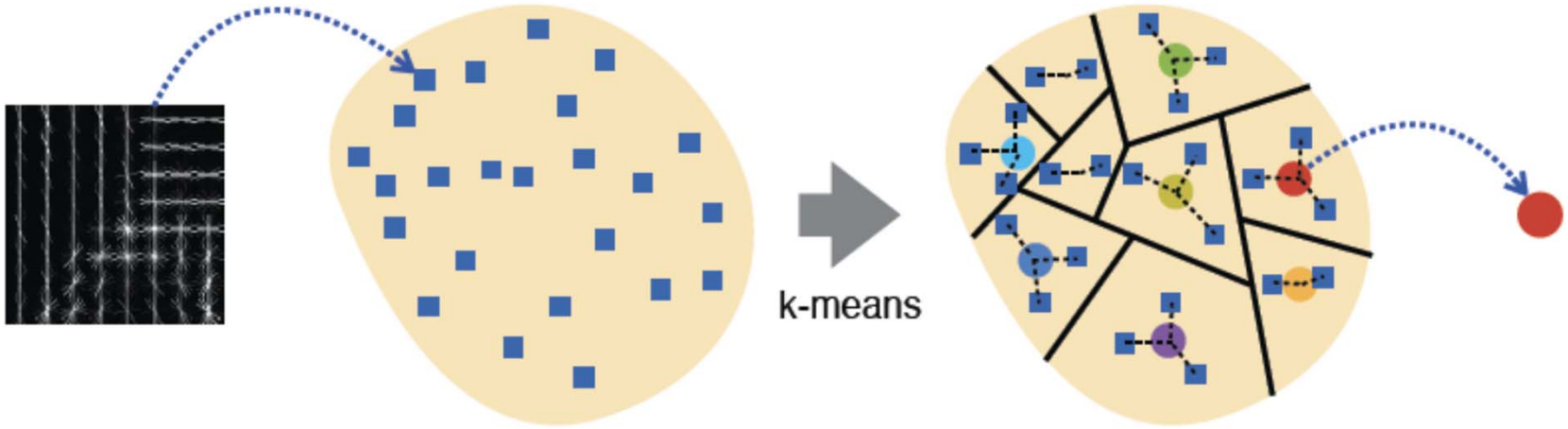
1. Extract local descriptor densely
2. Quantise descriptors
3. Form histogram

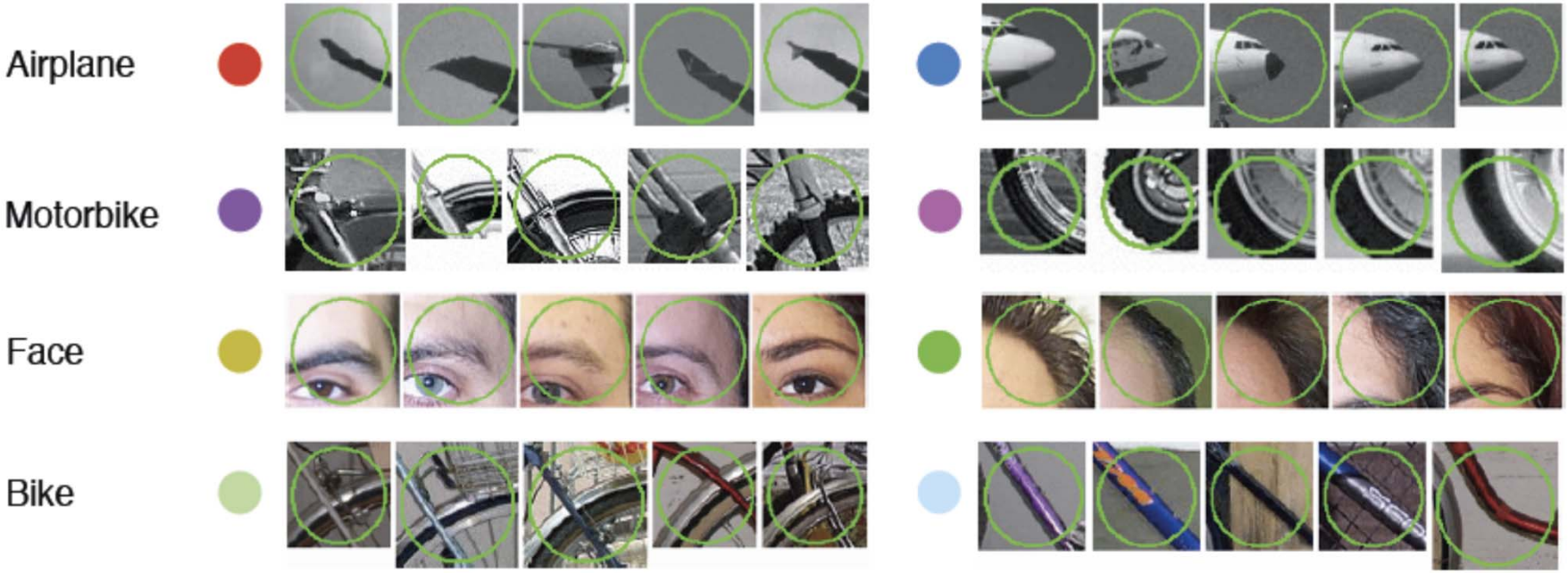
2. **Discards spatial information**



+ l^2 normalisation

Quantisation

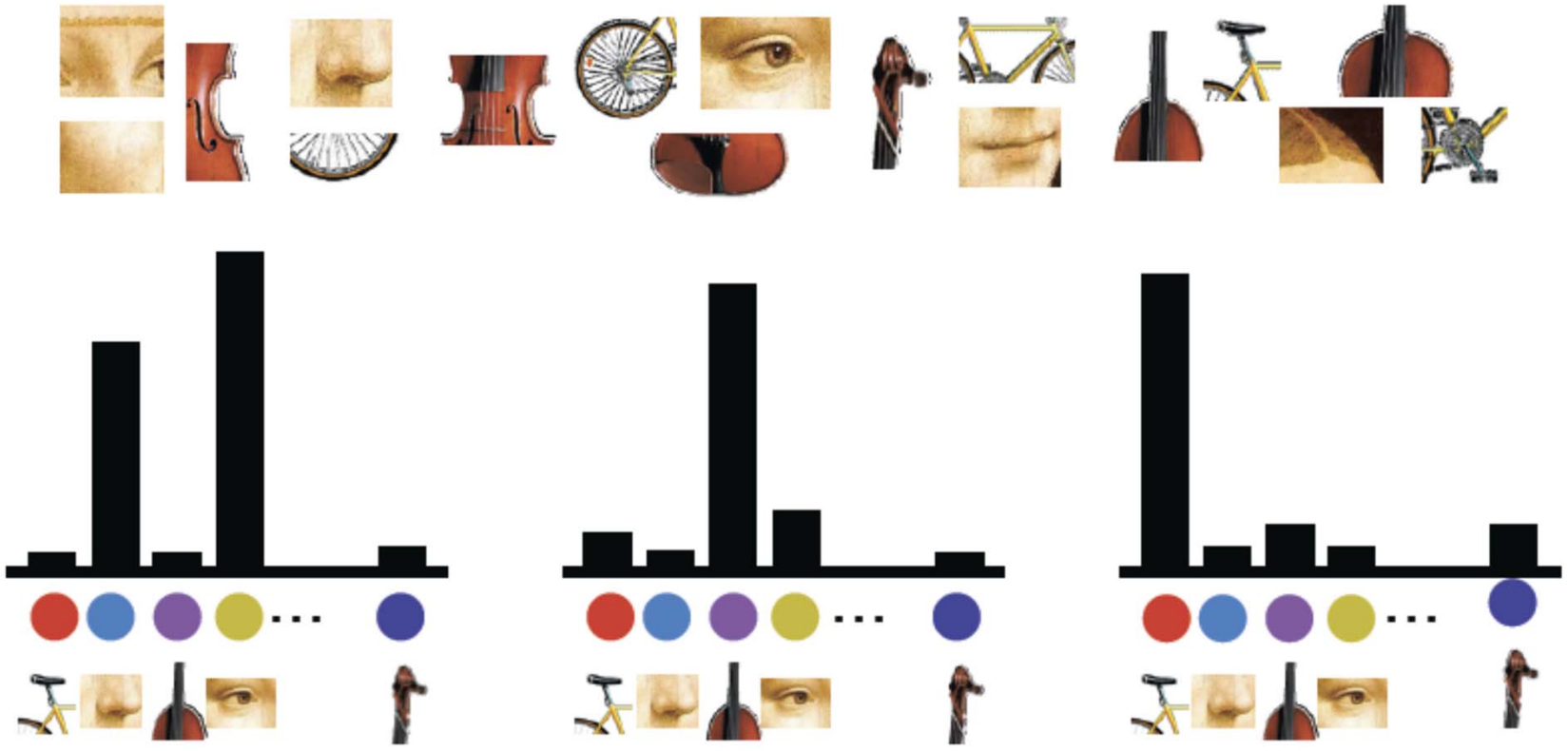




BoVW intuition

Discarding spatial information gives **lots of invariance**

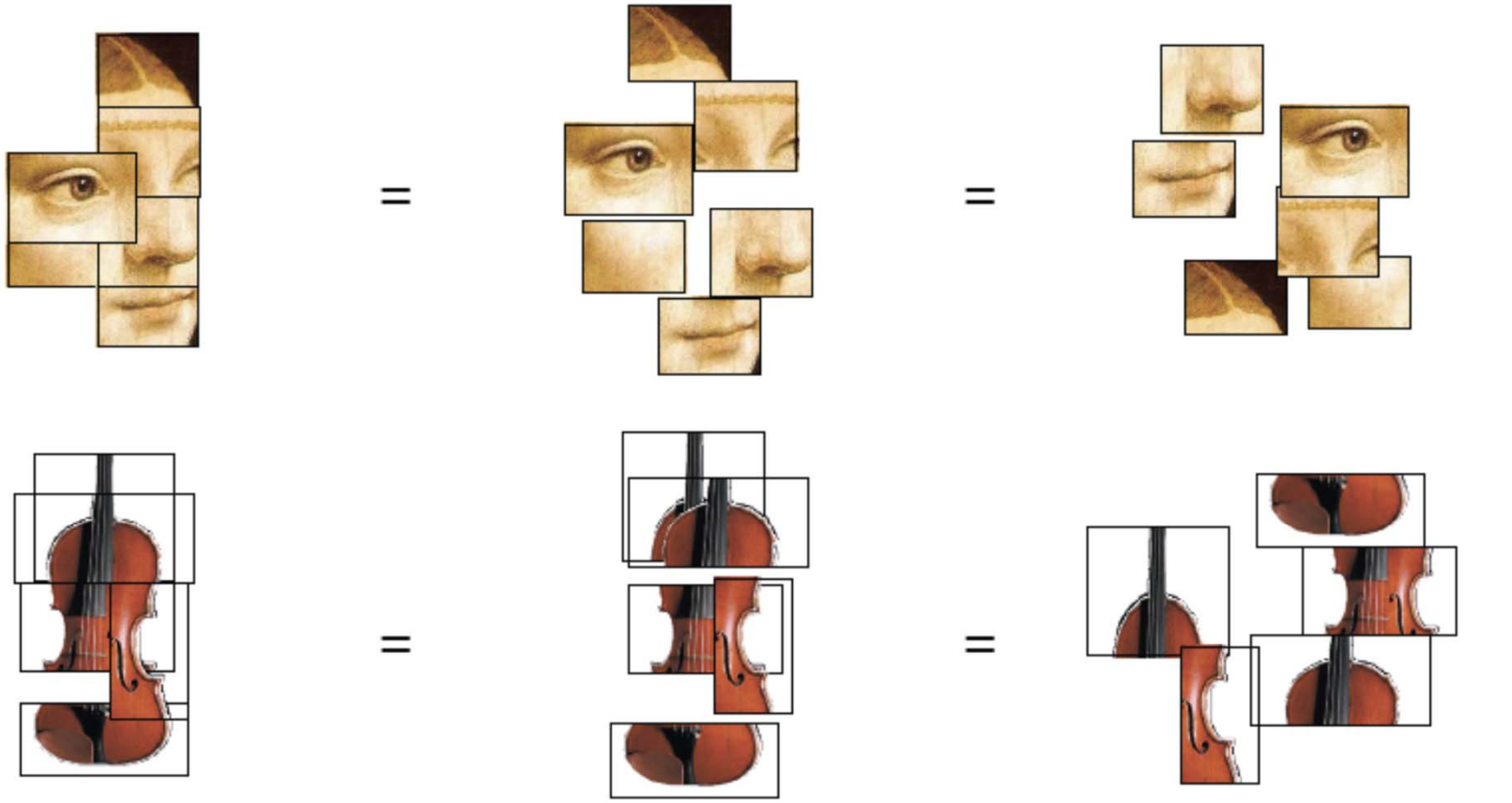
Visual words represent **"iconic" image fragments**



The loss of spatial information

Bag of features representation effectively forgets the relative location of the features

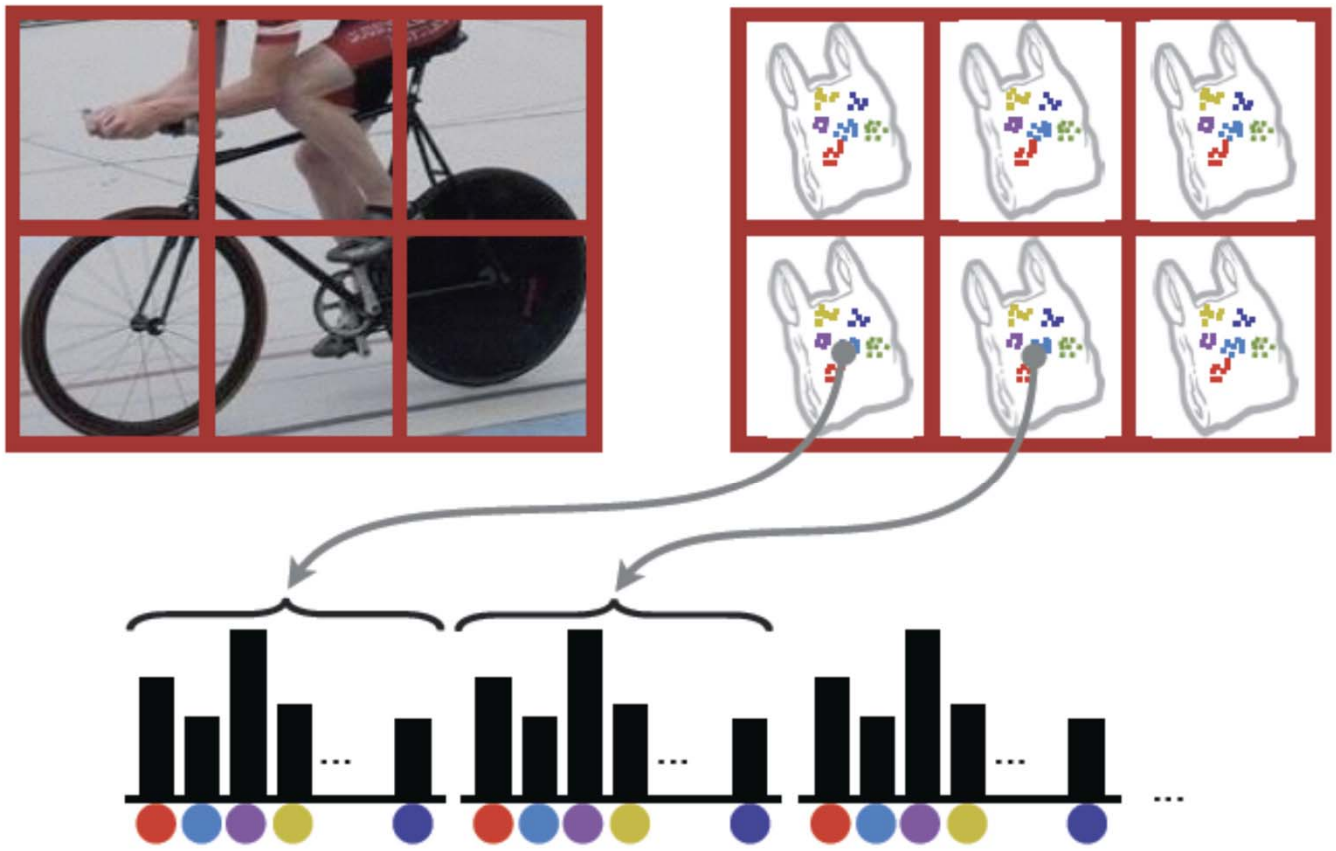
image | plausible deformation | implausible deformation



Spatial histograms

[Lazebnik *et al.* 2006]

Weak geometry: **pool spatial information locally**

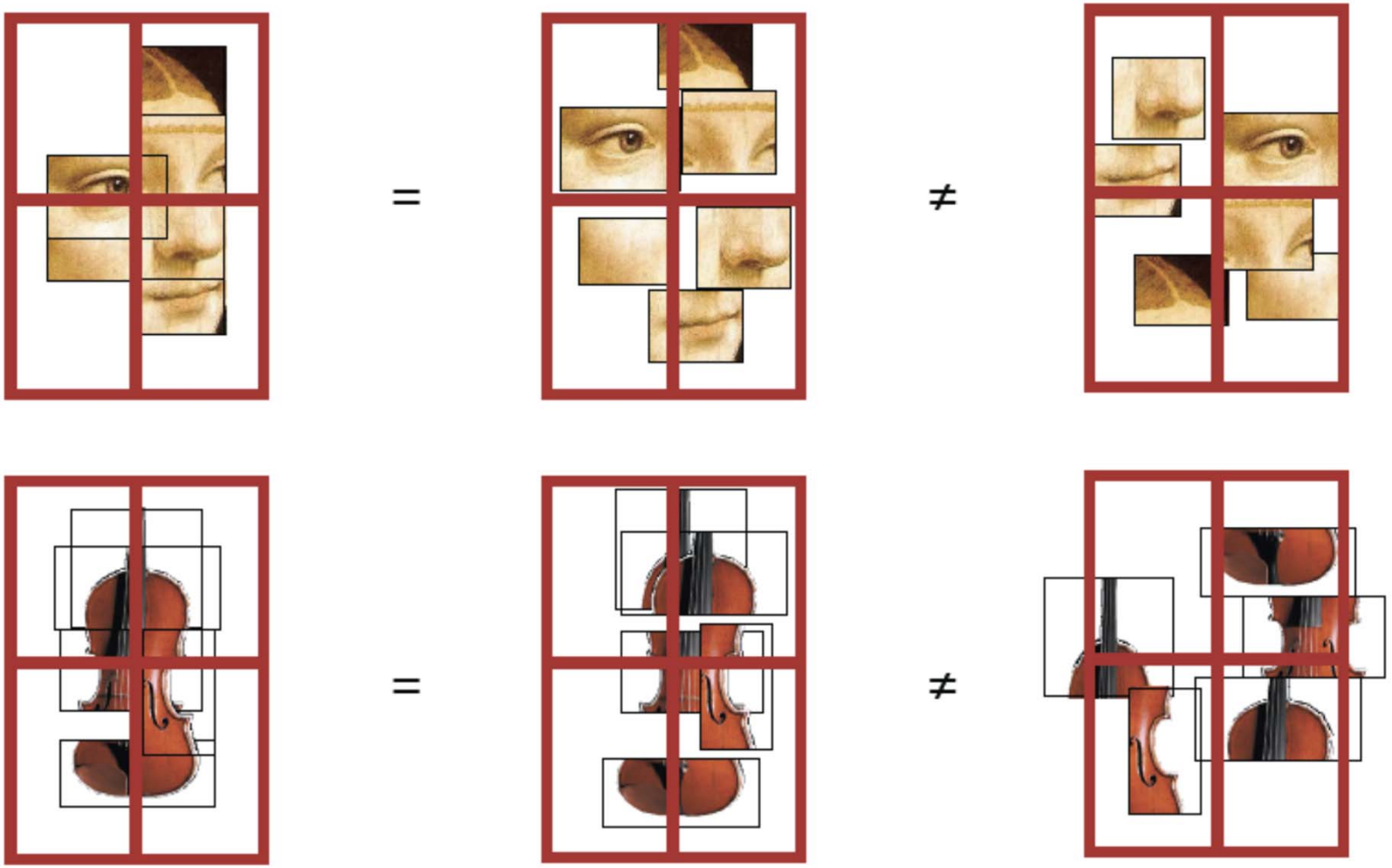


Spatial histograms capture weak geometry

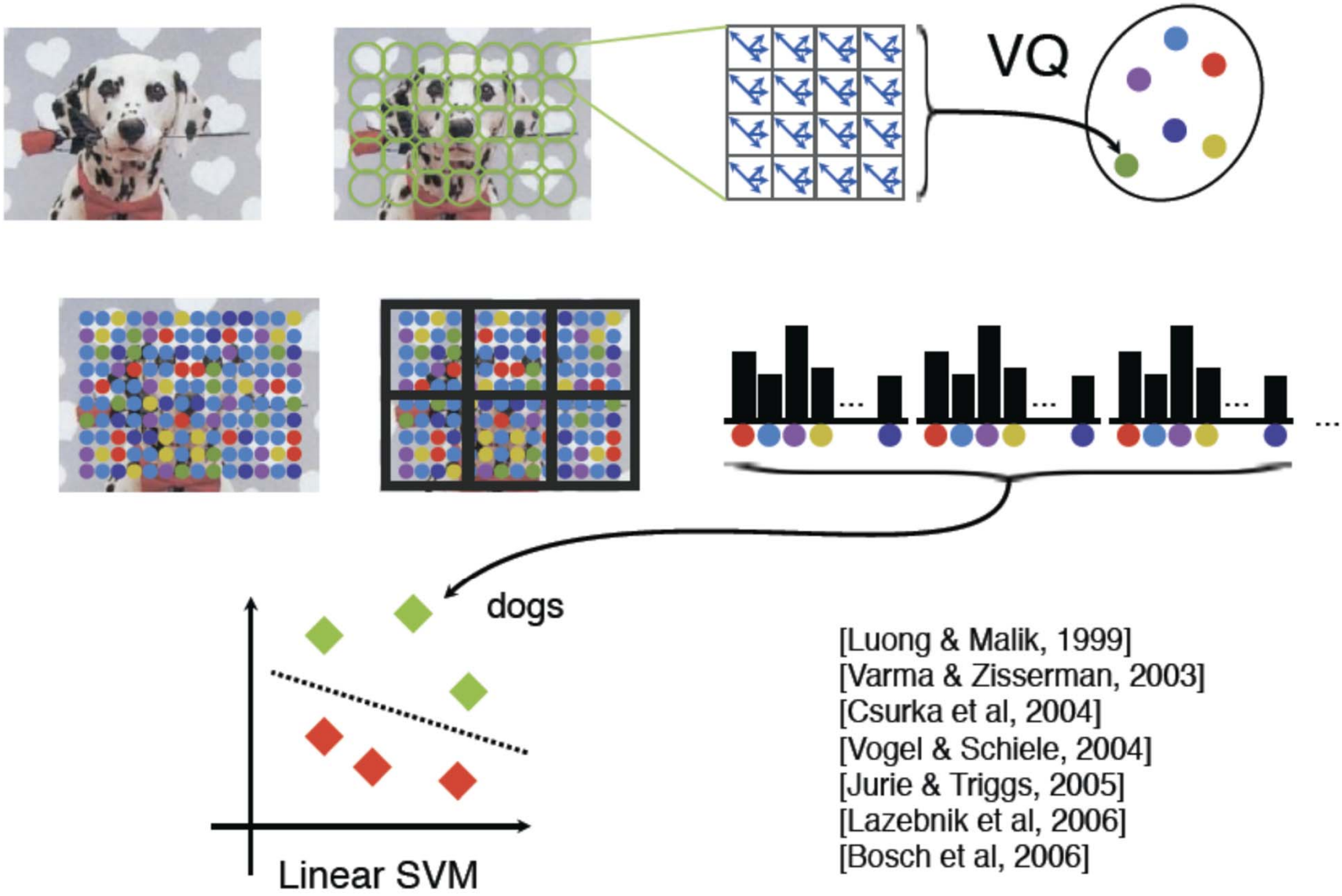
image

plausible deformation

implausible deformation



Summary so far



[Luong & Malik, 1999]
[Varma & Zisserman, 2003]
[Csurka et al, 2004]
[Vogel & Schiele, 2004]
[Jurie & Triggs, 2005]
[Lazebnik et al, 2006]
[Bosch et al, 2006]

Advanced encodings

Soft and **sparse** assignments, e.g.

- [Philbin et al CVPR 08, Gemert et al ECCV 08]
- Locality-constrained linear coding (LLC) – [Wang et al CVPR 10]

Representing SIFT distribution **mean** in Voronoi cell, e.g.

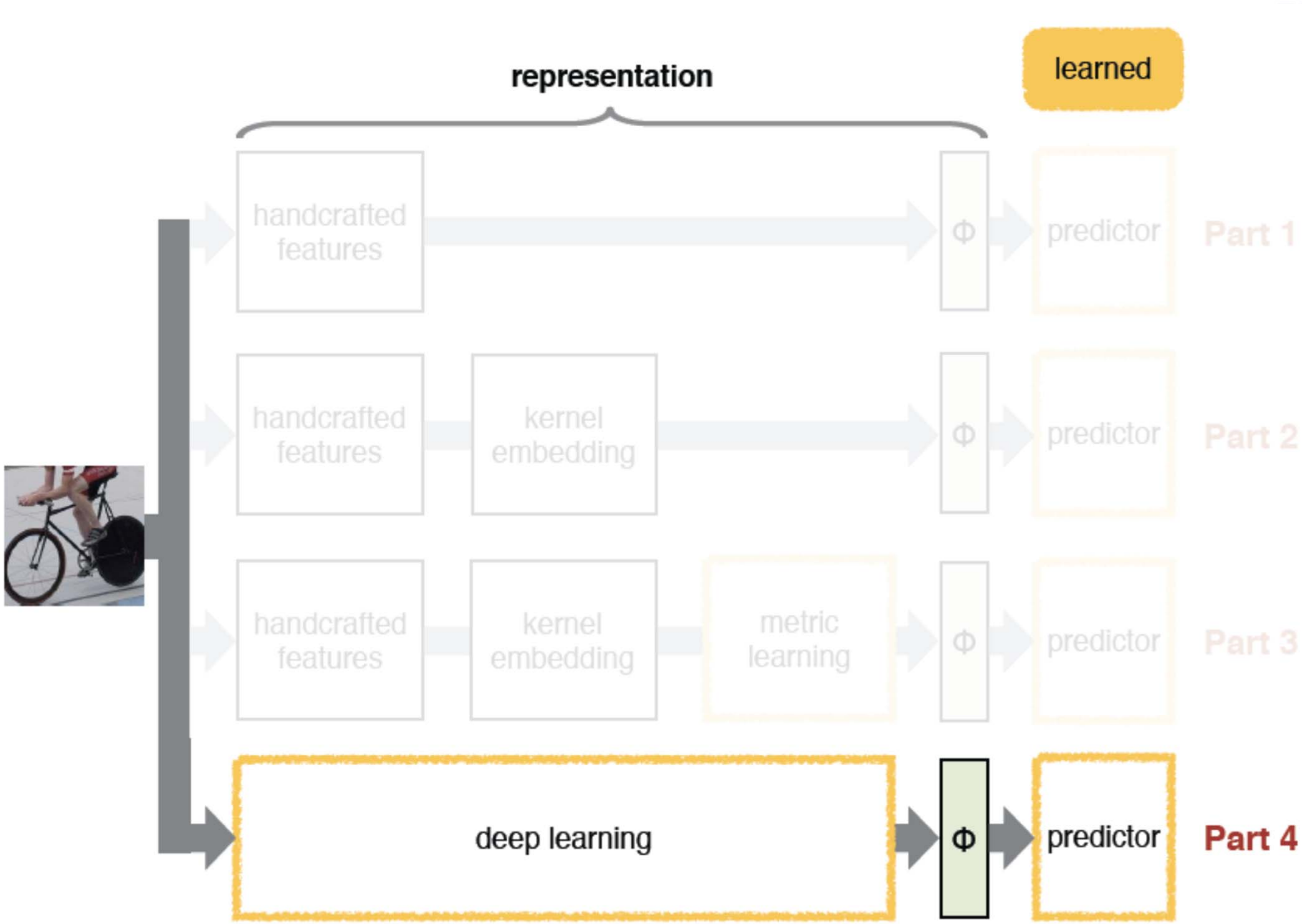
- Super-Vector Coding [Zhou et al ECCV 10]
- VLAD [Jegou et al CVPR 10]

Representing SIFT distribution **mean** and **covariance** in Voronoi cell, e.g.

- Fisher vector [Perronnin et al CVPR 07 & 10, ECCV 10]

Improvements to **normalization**, PCA, **whitening** for VLAD/FV

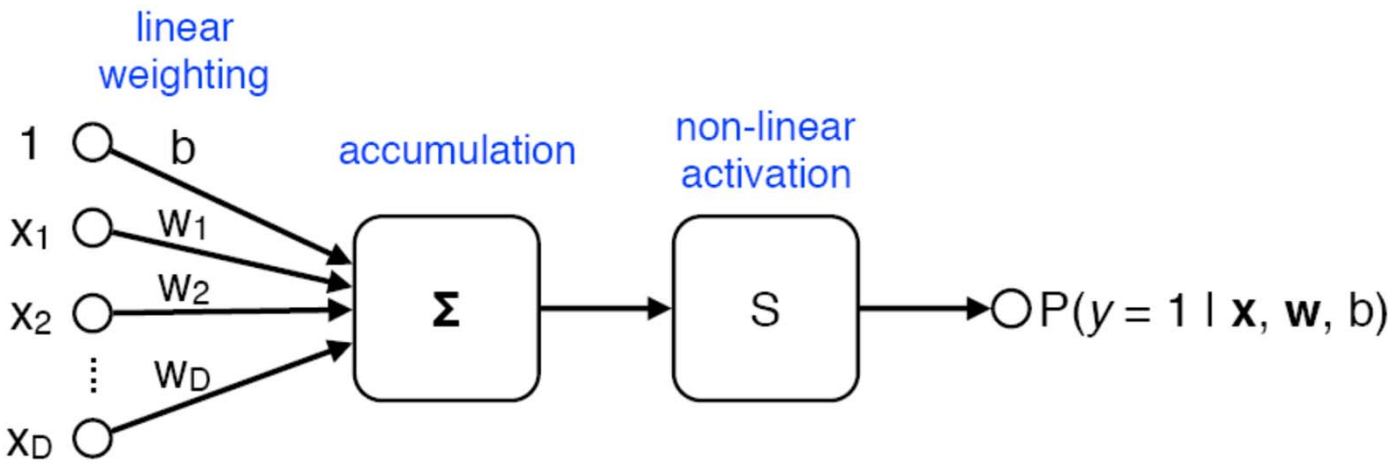
- Chen et al 2011 [Jegou & Chum ECCV 12]
- All about VLAD [Arandjelovic & Zisserman CVPR 13]


representation
learned
handcrafted features
Φ
predictor
Part 1
handcrafted features
kernel embedding
Φ
predictor
Part 2
handcrafted features
kernel embedding
metric learning
Φ
predictor
Part 3
deep learning
Φ
predictor
Part 4

Perceptron

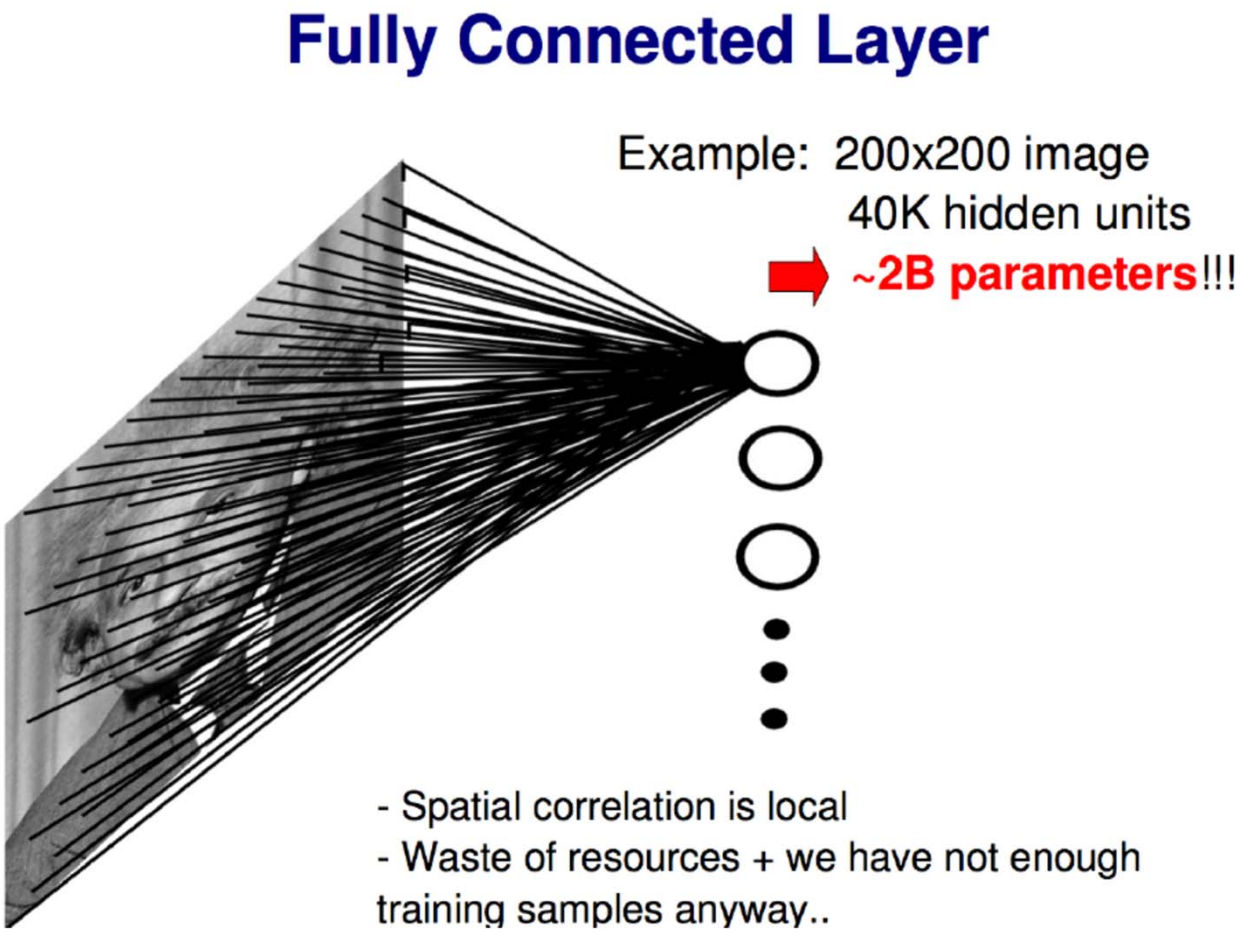
[Rosenblatt 57]

The goal is estimating the posterior probability of the binary label *y* of a vector **x**:

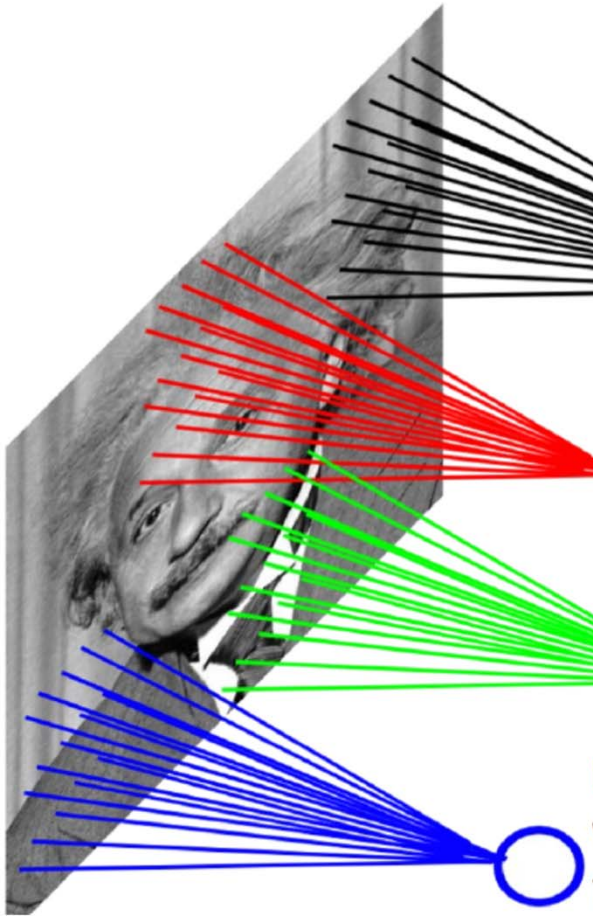


$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

Network Connectivity



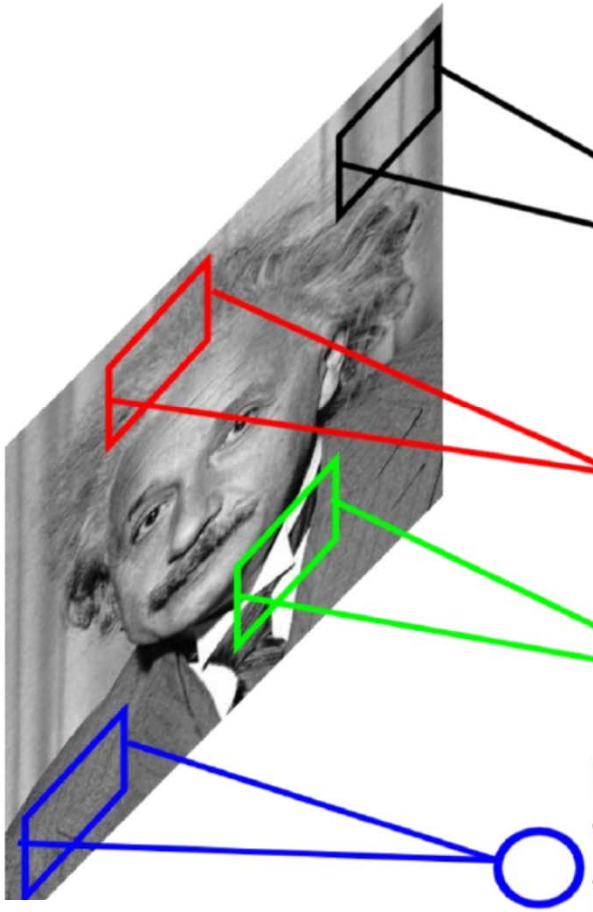
Locally Connected Layer



Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

Locally Connected Layer

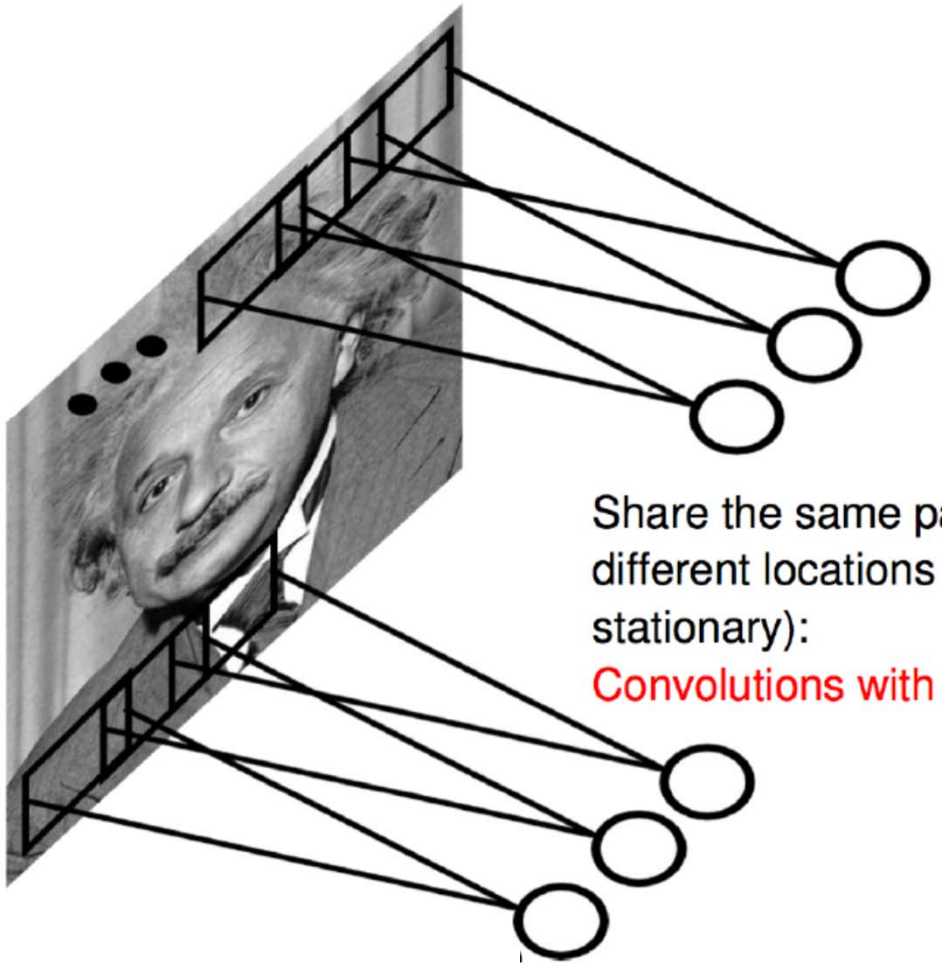


STATIONARITY? Statistics is similar at different locations

Example: 200x200 image
40K hidden units
Filter size: 10x10
4M parameters

Note: This parameterization is good when input image is registered (e.g., face recognition).

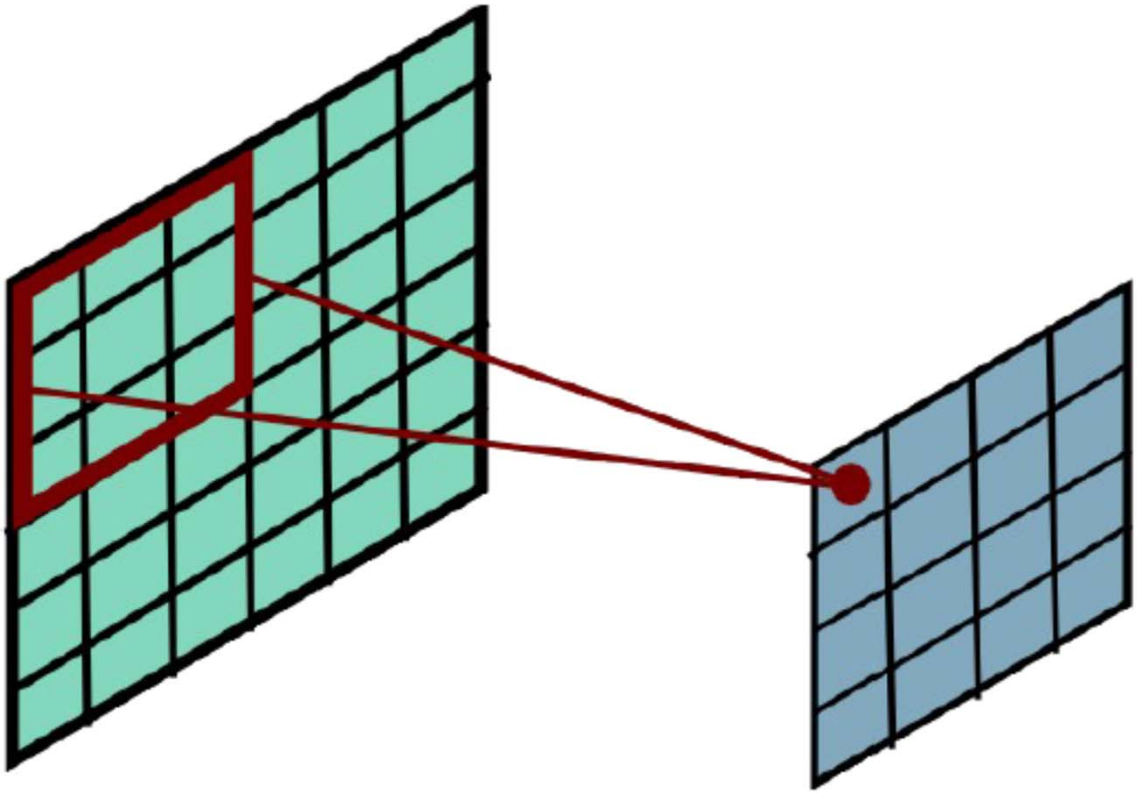
Convolutional Layer



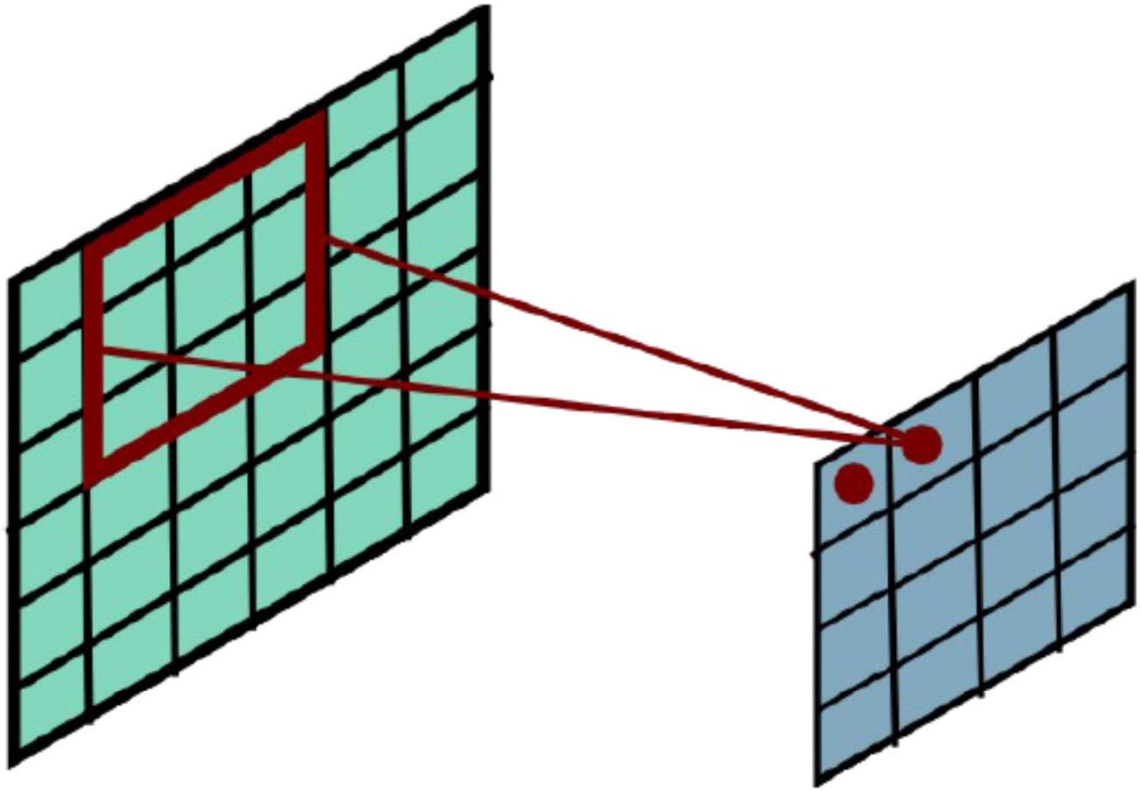
Share the same parameters across different locations (assuming input is stationary):

Convolutions with learned kernels

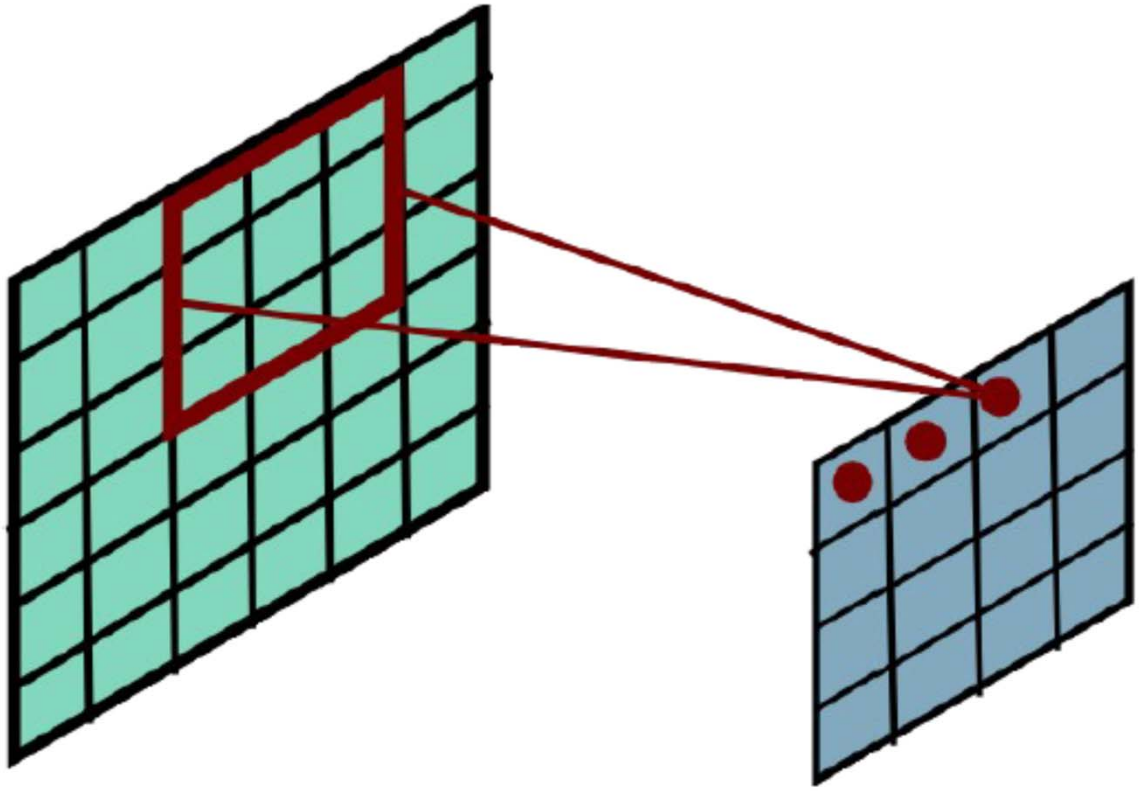
Convolutional Layer



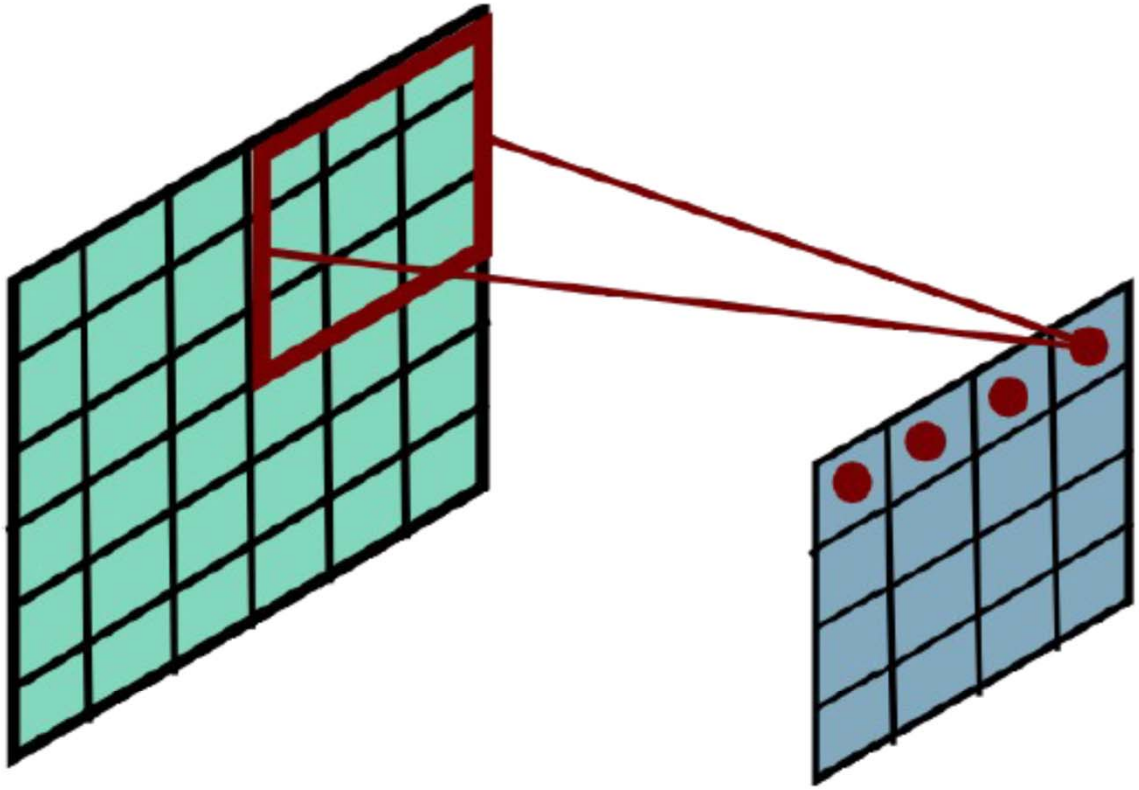
Convolutional Layer



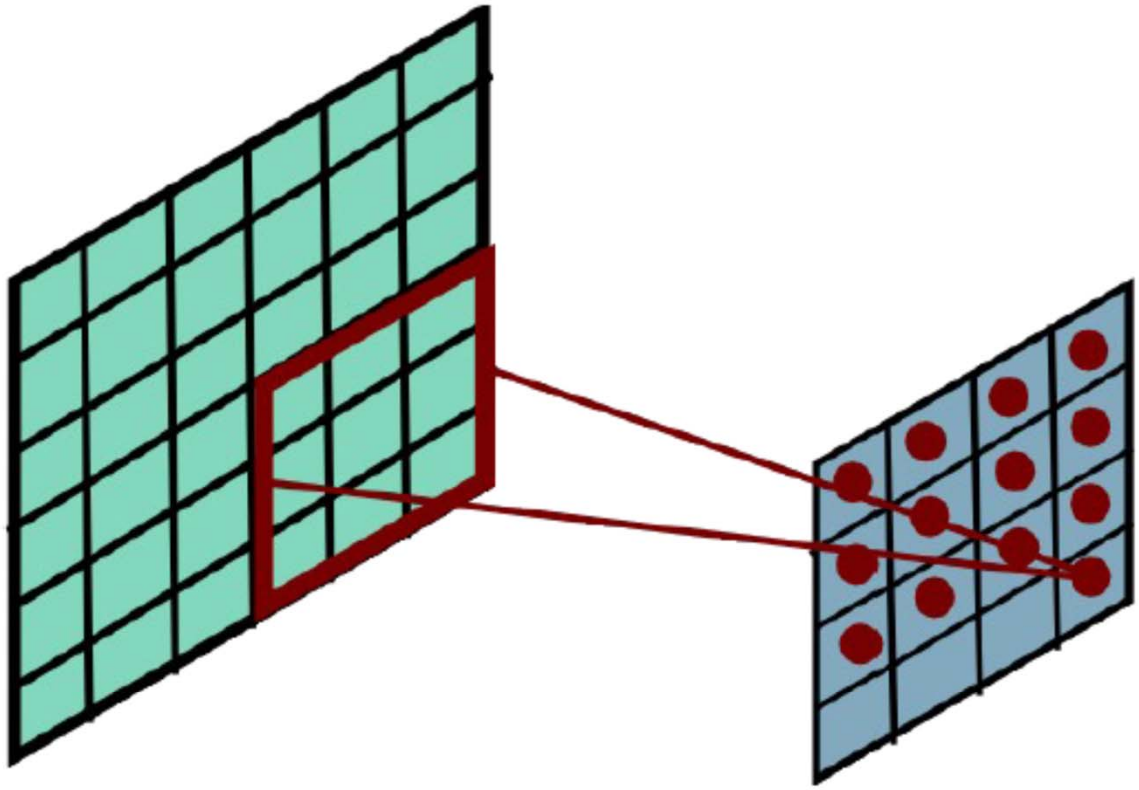
Convolutional Layer



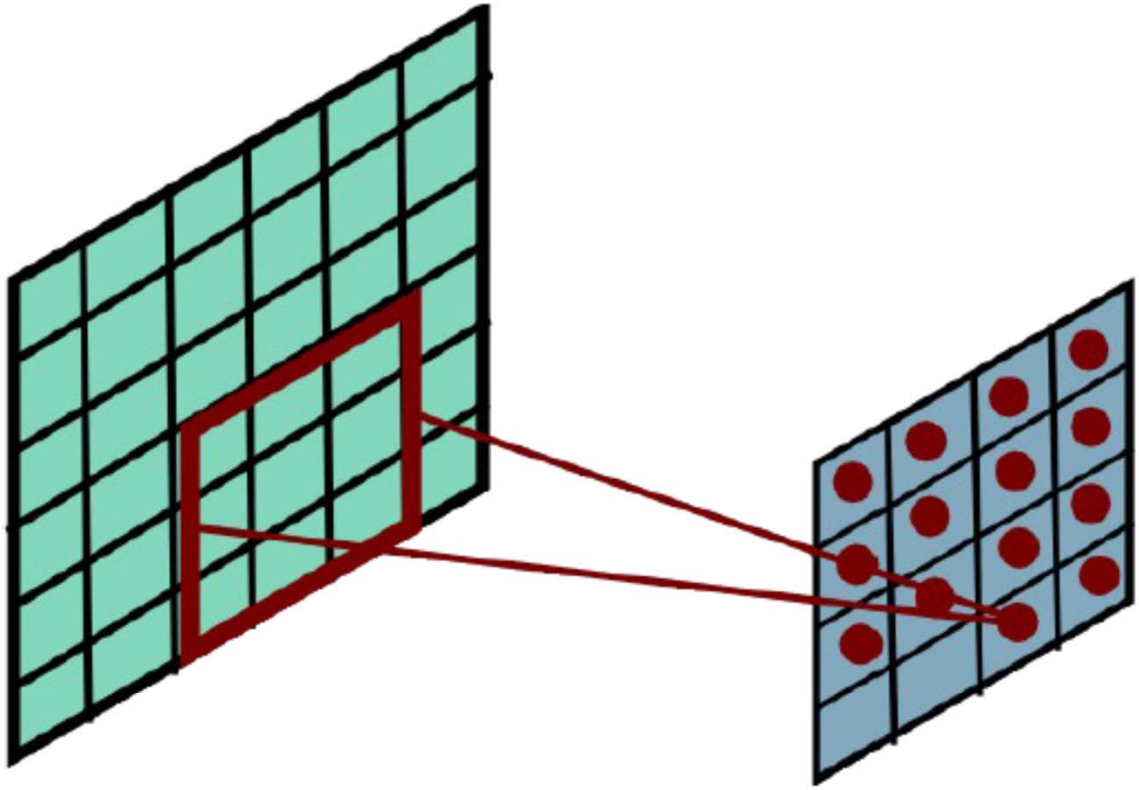
Convolutional Layer



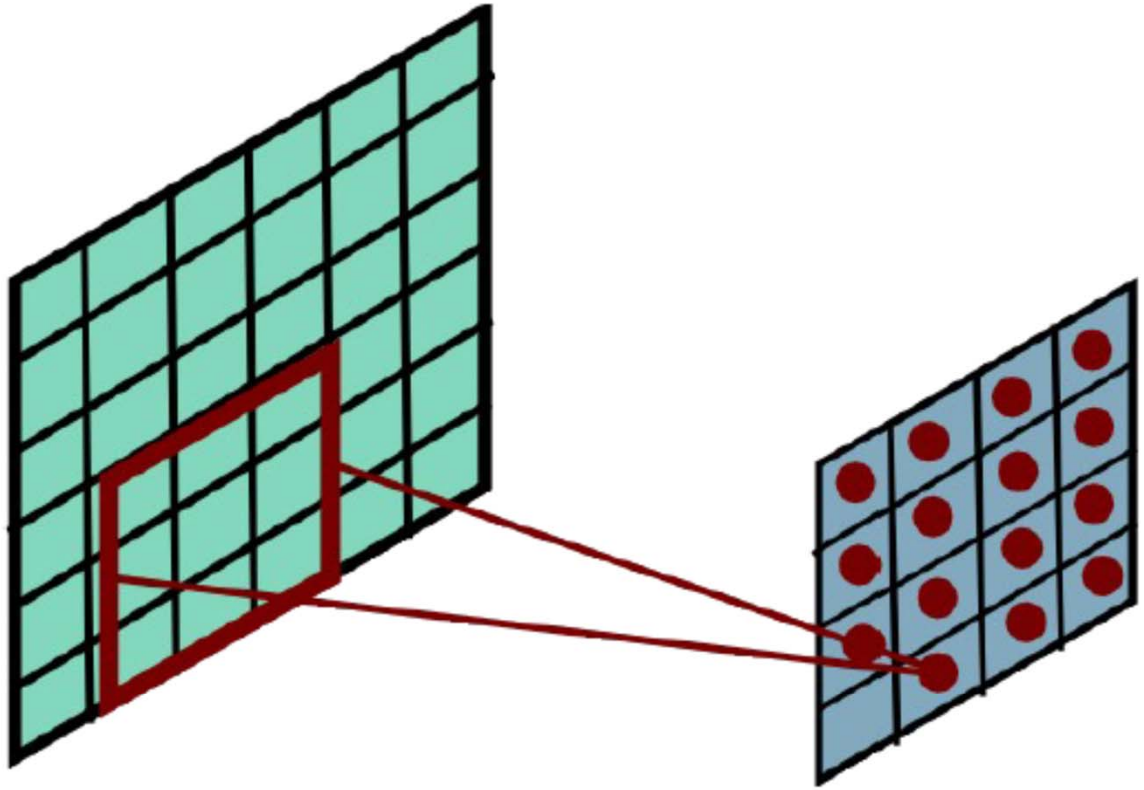
Convolutional Layer



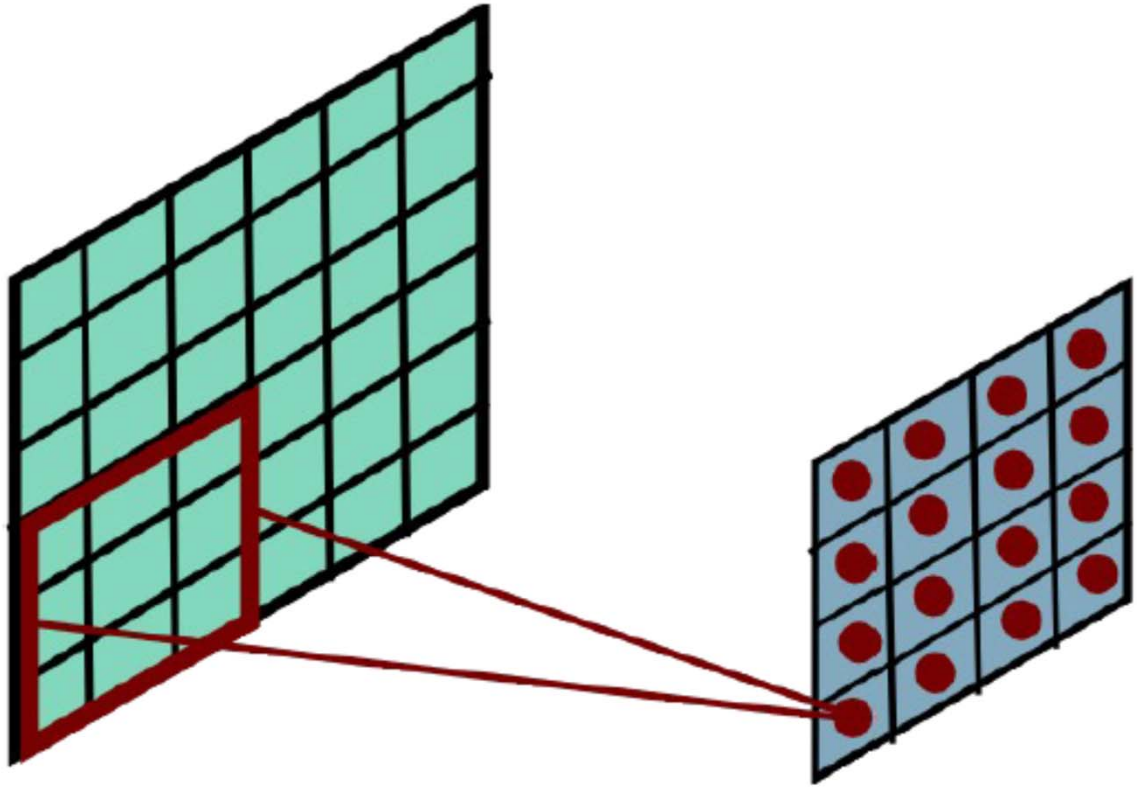
Convolutional Layer



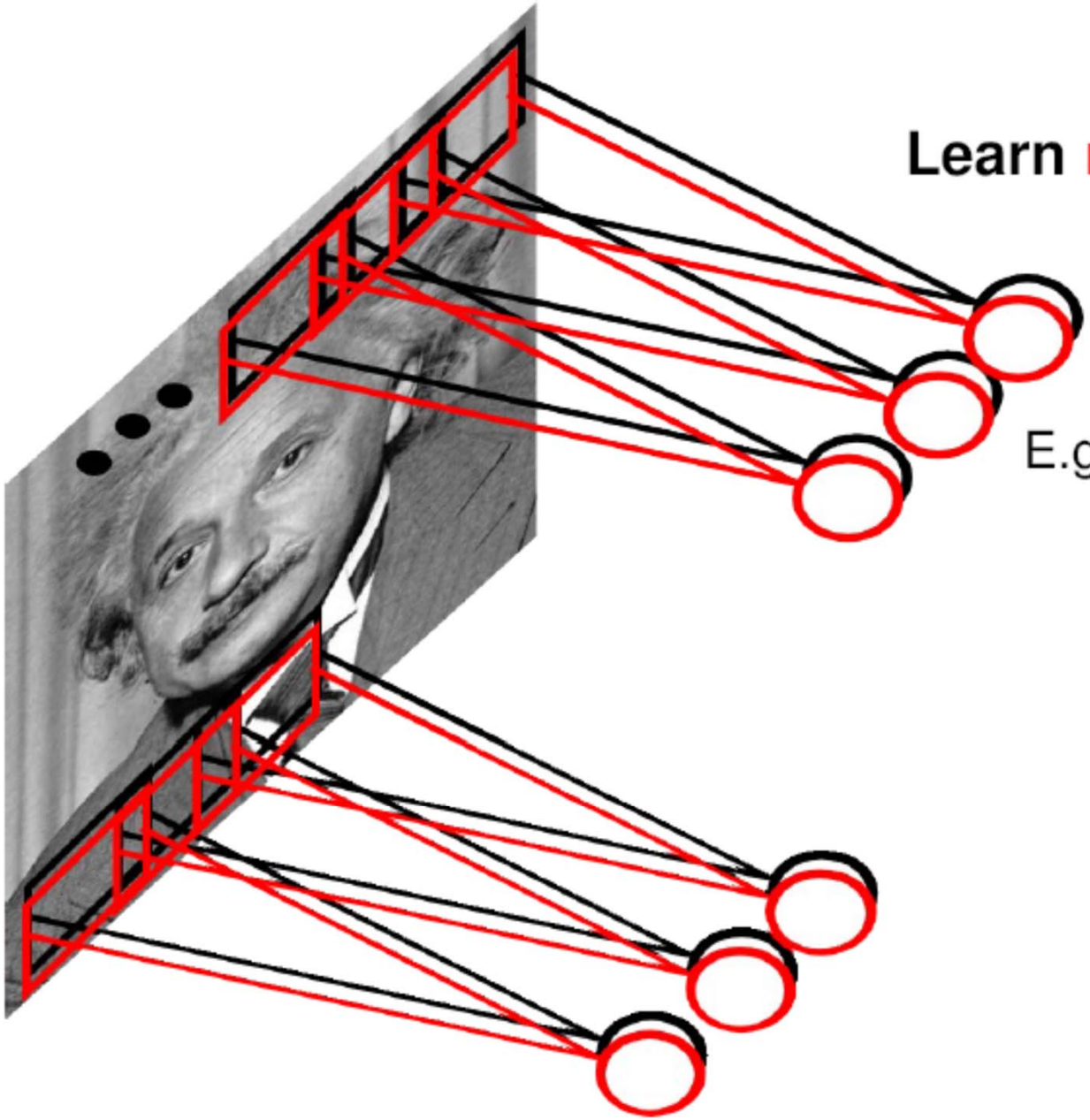
Convolutional Layer



Convolutional Layer



Convolutional Layer



Learn multiple filters.

E.g.: 200x200 image
100 Filters
Filter size: 10x10
10K parameters

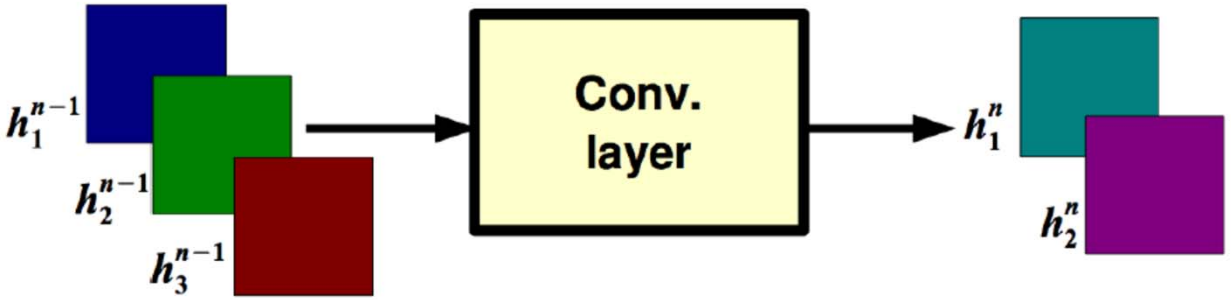
Convolutional Layer

$$h_j^n = max\left(0, \sum_{k=1}^{K} h_k^{n-1} * w_{kj}^n\right)$$

output feature map

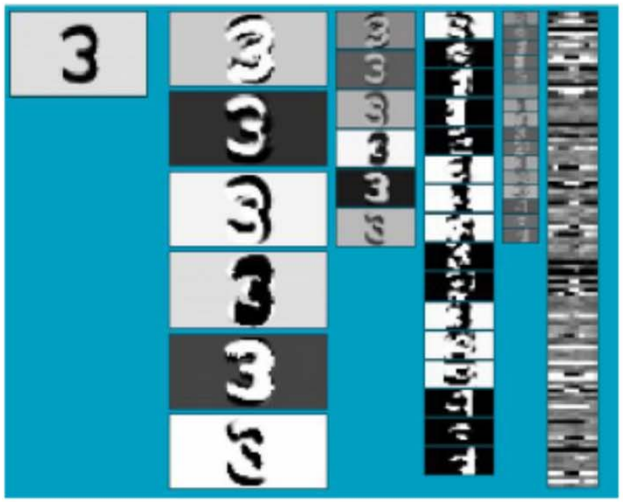
input feature map

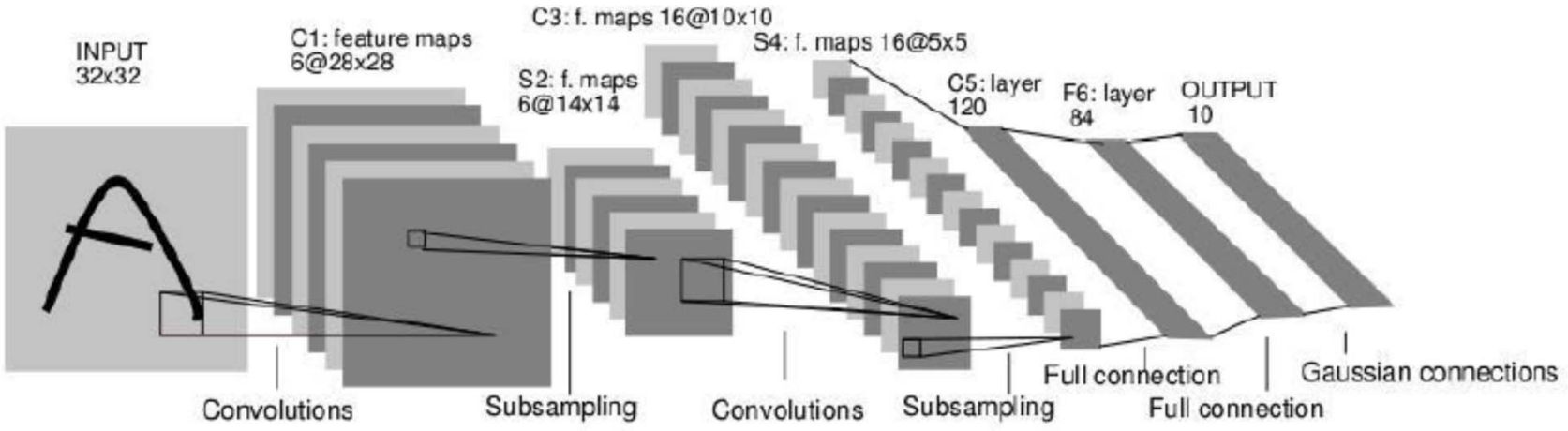
kernel



Example: Convolutional Neural Networks

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

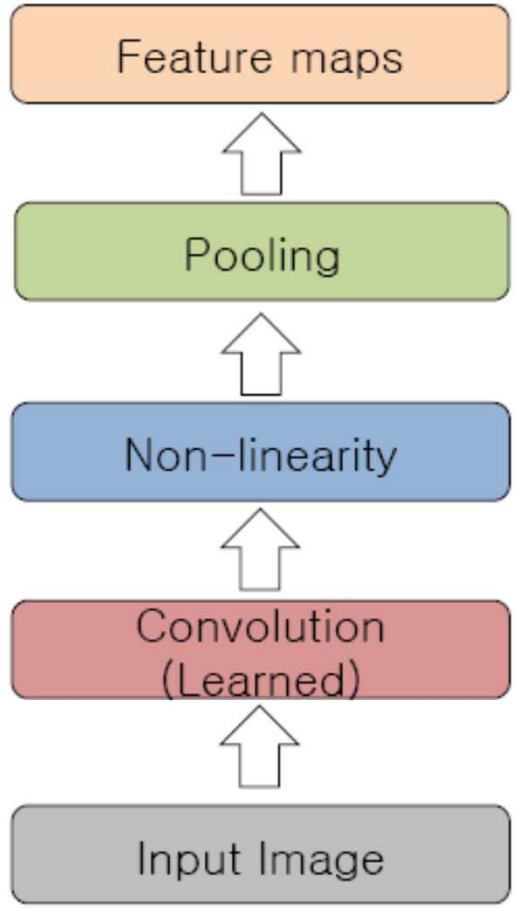




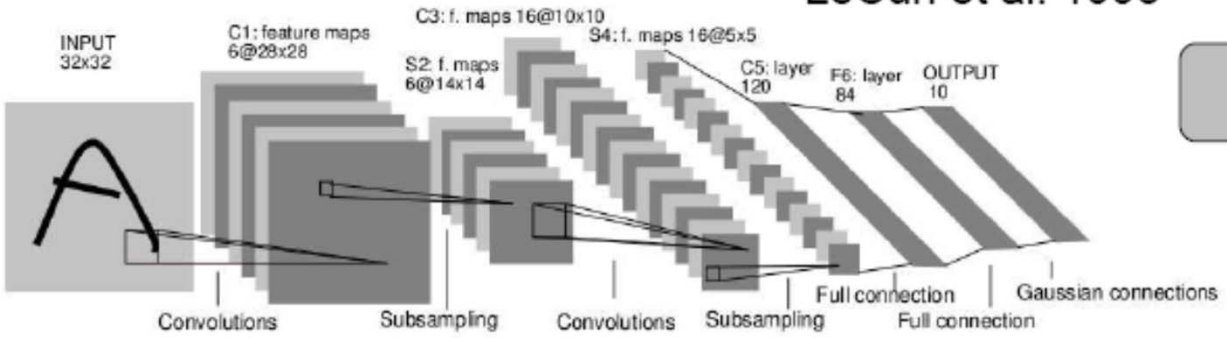
Slide: R. Fergus

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



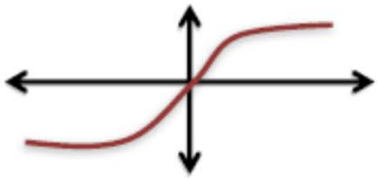
Slide: R. Fergus

Components of Each Layer

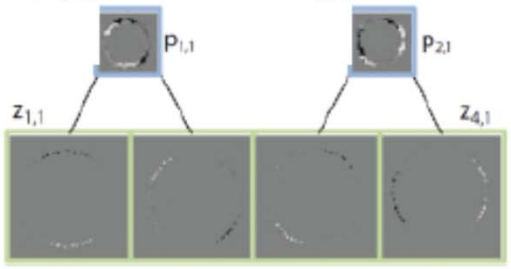
Pixels / Features

Filter with Dictionary (convolutional or tiled)

+ Non-linearity



Spatial/Feature (Sum or Max)



[Optional]

Normalization between feature responses

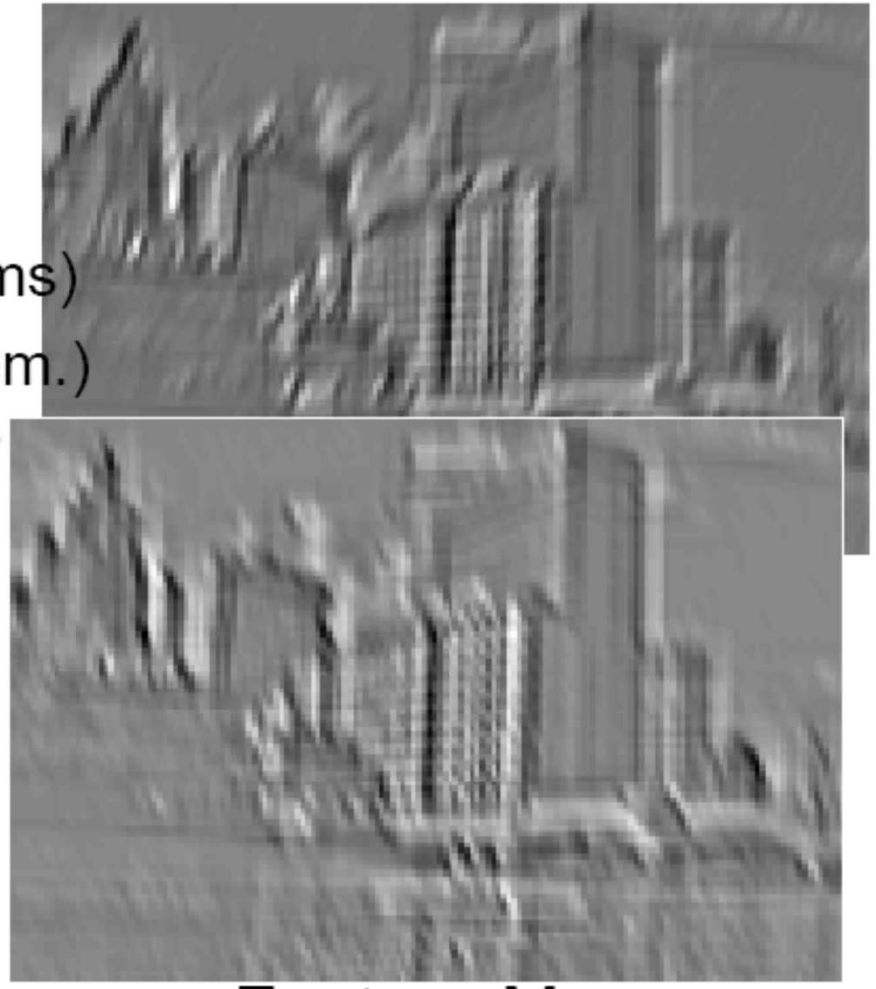
Output Features

Slide: R. Fergus

Filtering

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

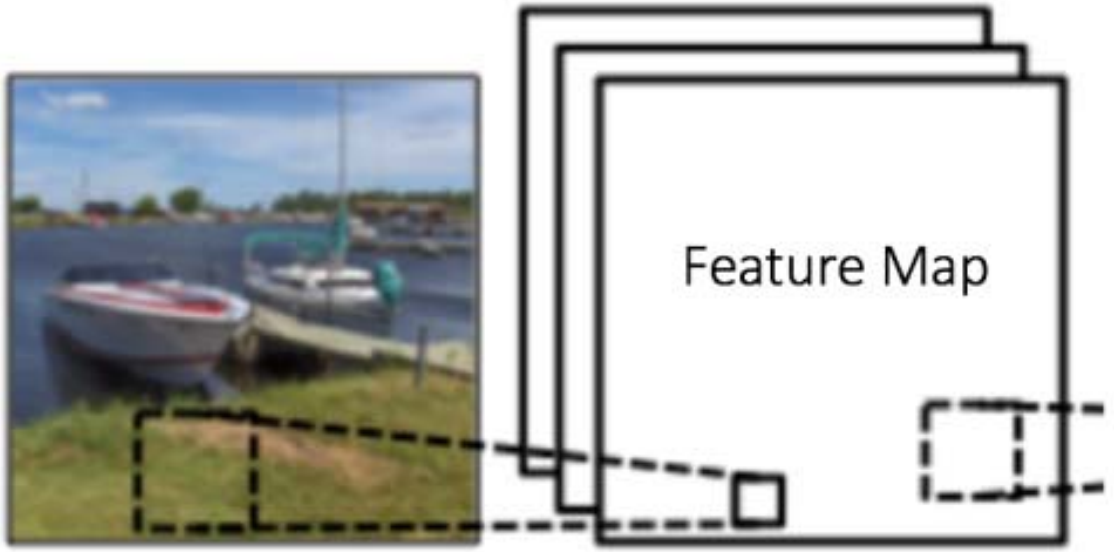
Input



Feature Map

Slide: R. Fergus

Feature map (Depth):



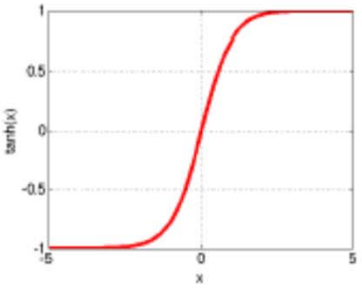
Feature Map having depth of 3 (since 3 filters have been used)

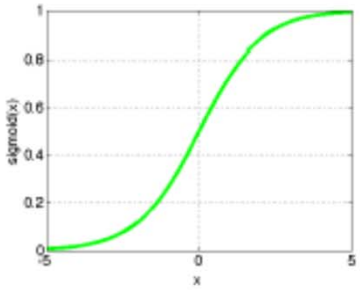
Convolution Operation

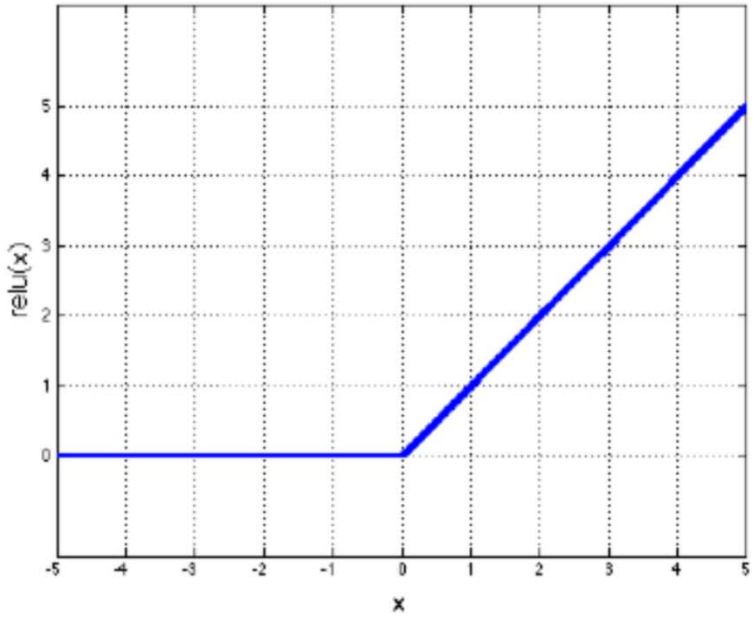
Non-Linearity

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

    → Preferred option



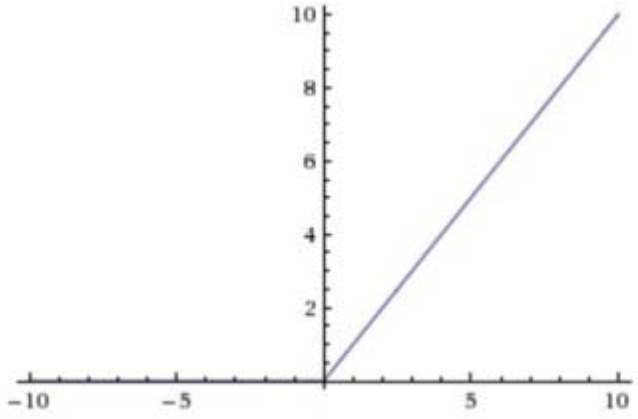


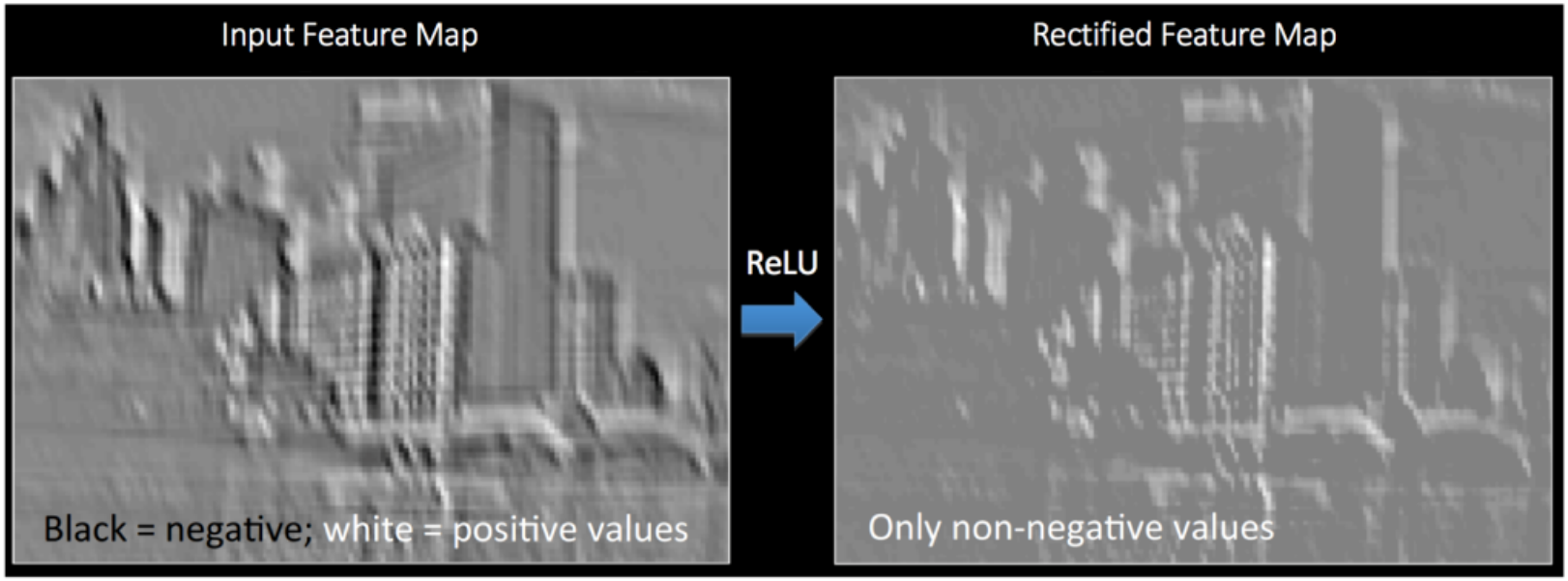


Slide: R. Fergus

Non Linearity RELU operation:

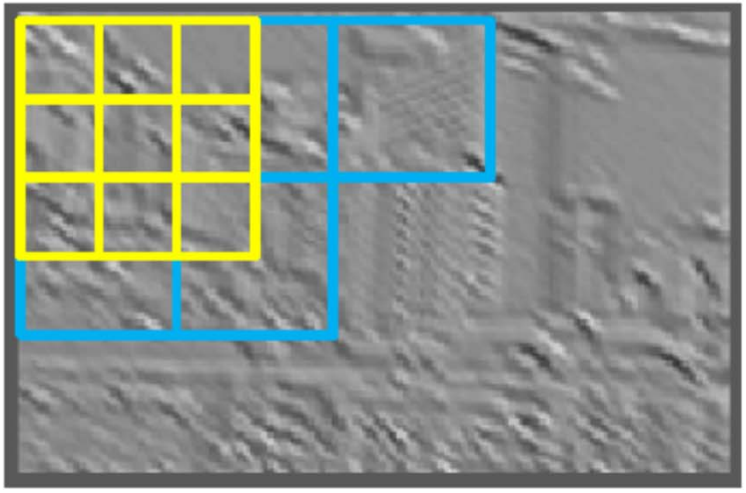
Output = Max(zero, Input)

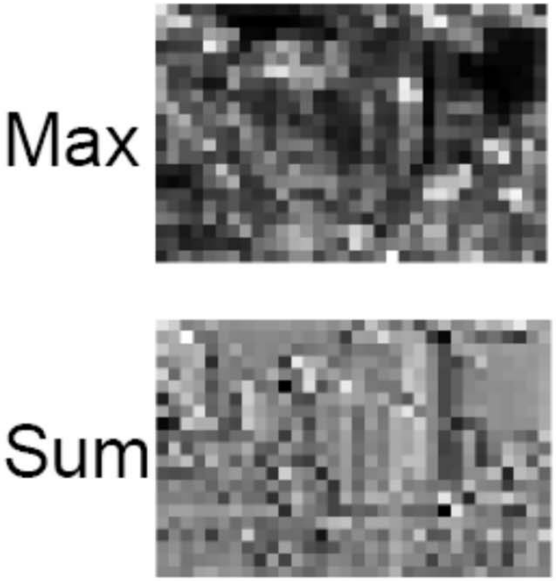




Pooling

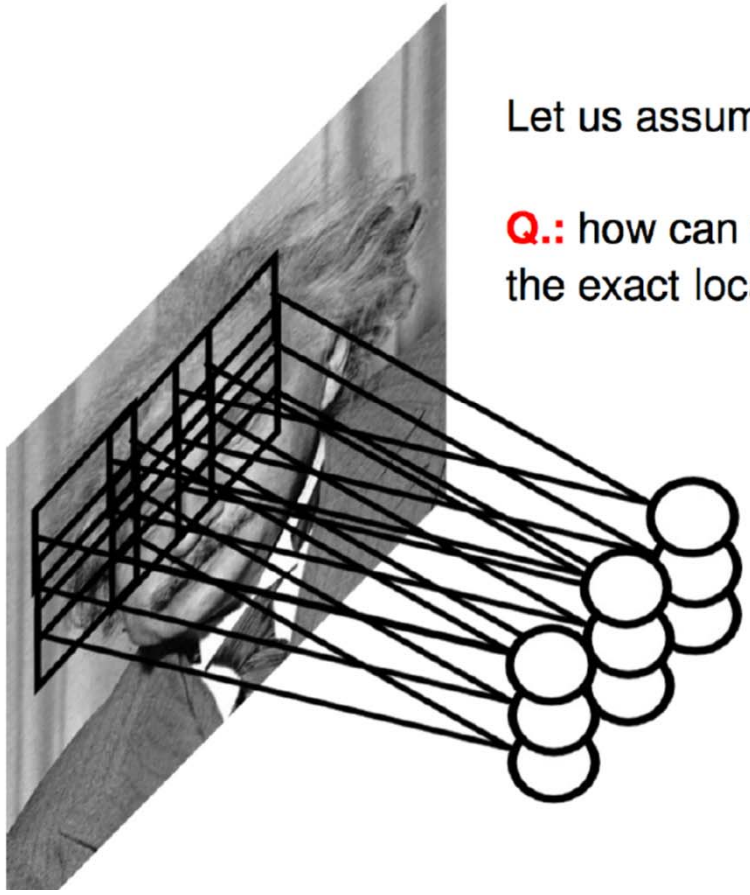
- Spatial Pooling
  - Non-overlapping / overlapping regions
  - Sum or max
  - Boureau et al. ICML'10 for theoretical analysis





Slide: R. Fergus

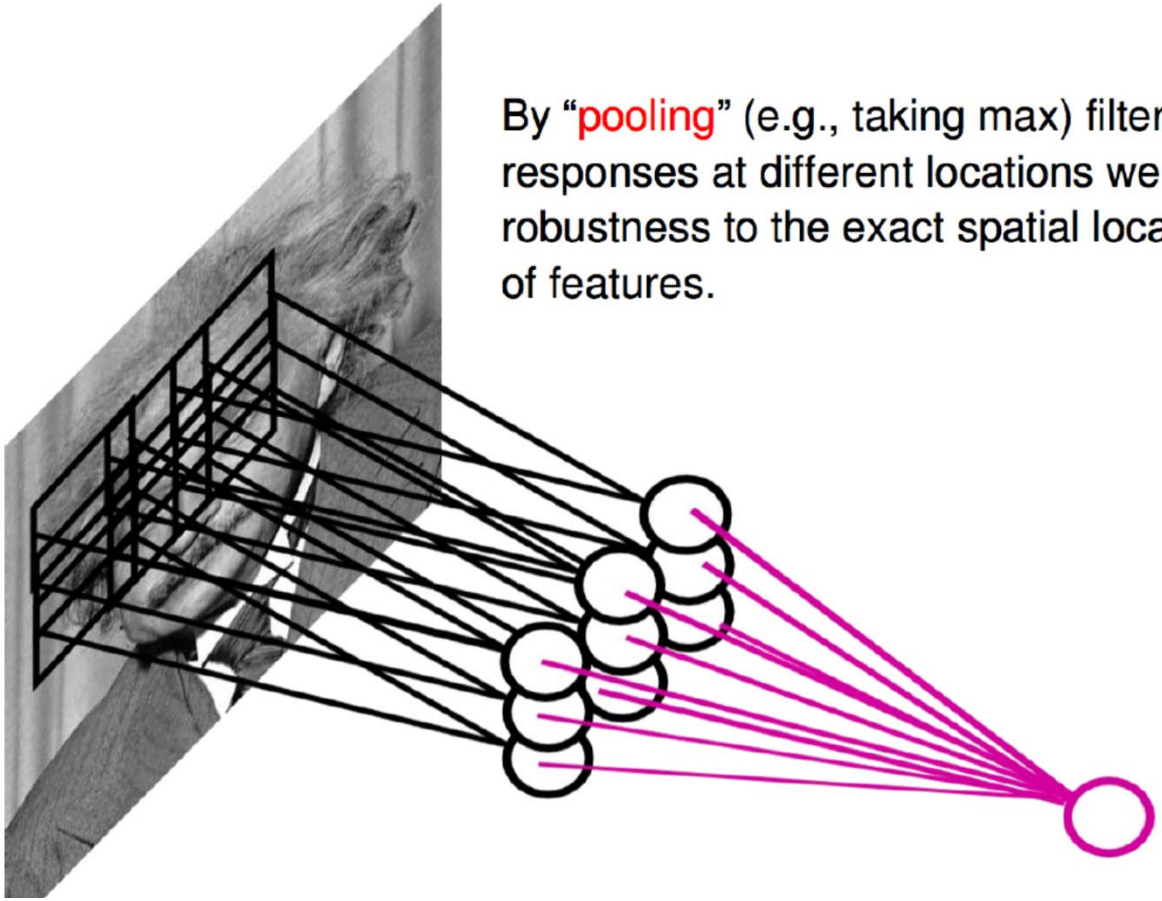
Pooling Layer



Let us assume filter is an “eye” detector.

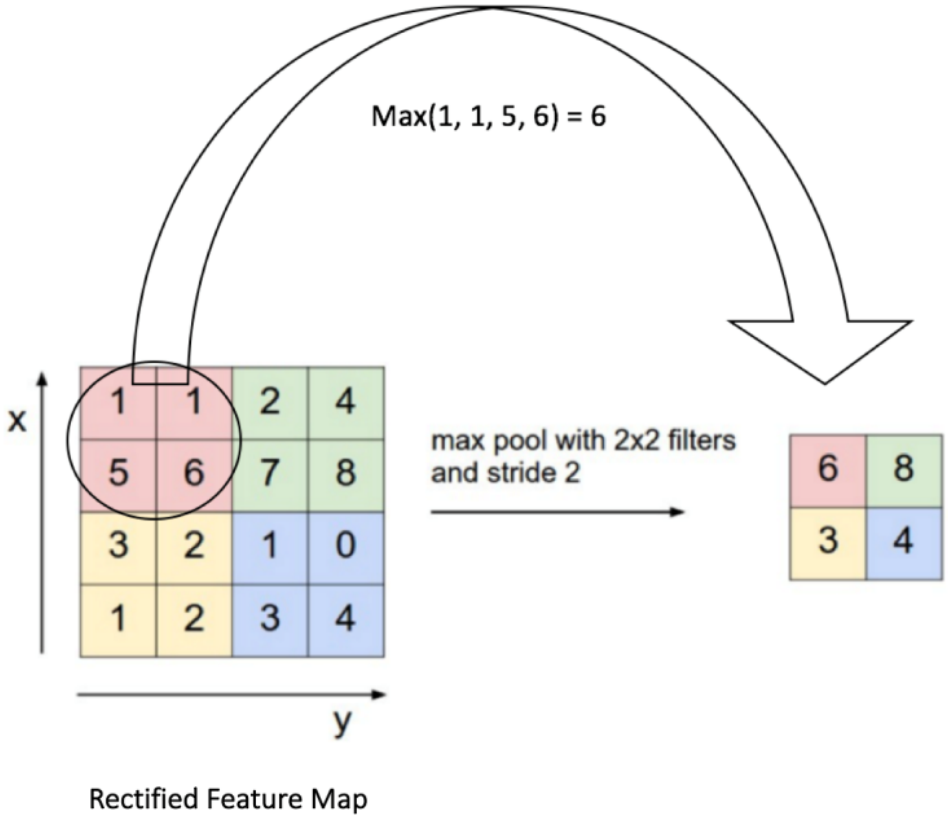
Q.: how can we make the detection robust to the exact location of the eye?

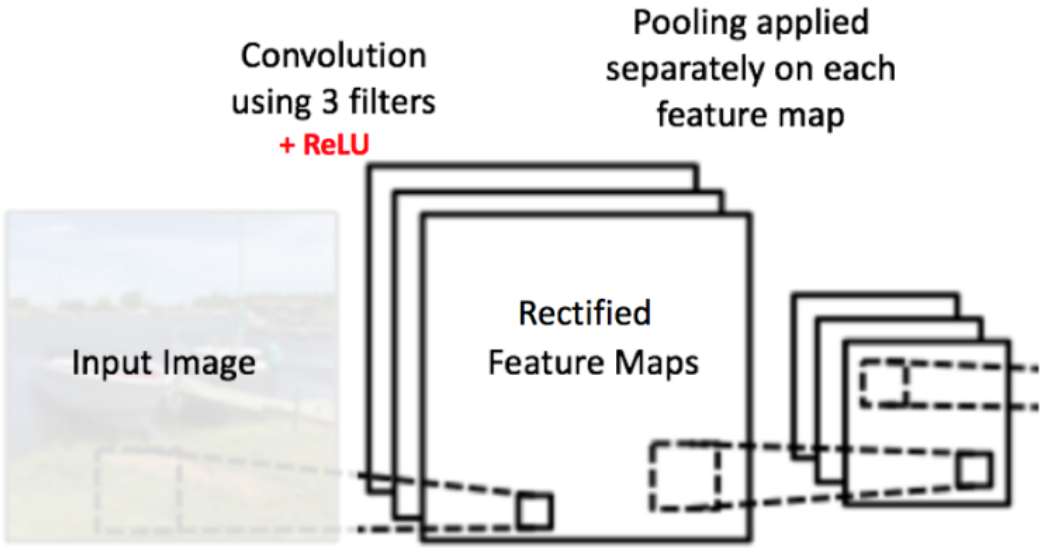
Pooling Layer



By "pooling" (e.g., taking max) filter responses at different locations we gain robustness to the exact spatial location of features.

Max pooling

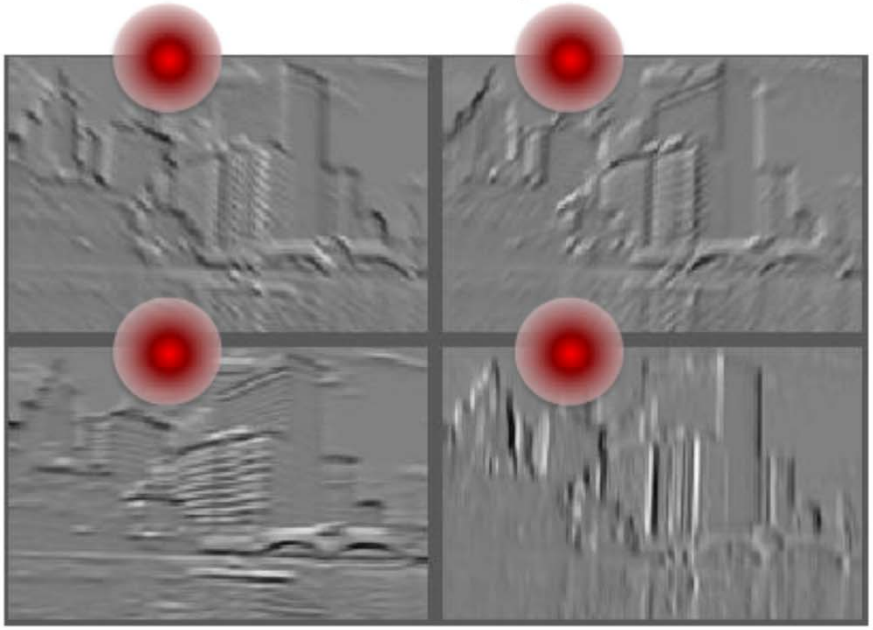

Max(1, 1, 5, 6) = 6
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4
Rectified Feature Map


Convolution
using 3 filters
+ ReLU
Pooling applied
separately on each
feature map
Input Image
Rectified
Feature Maps

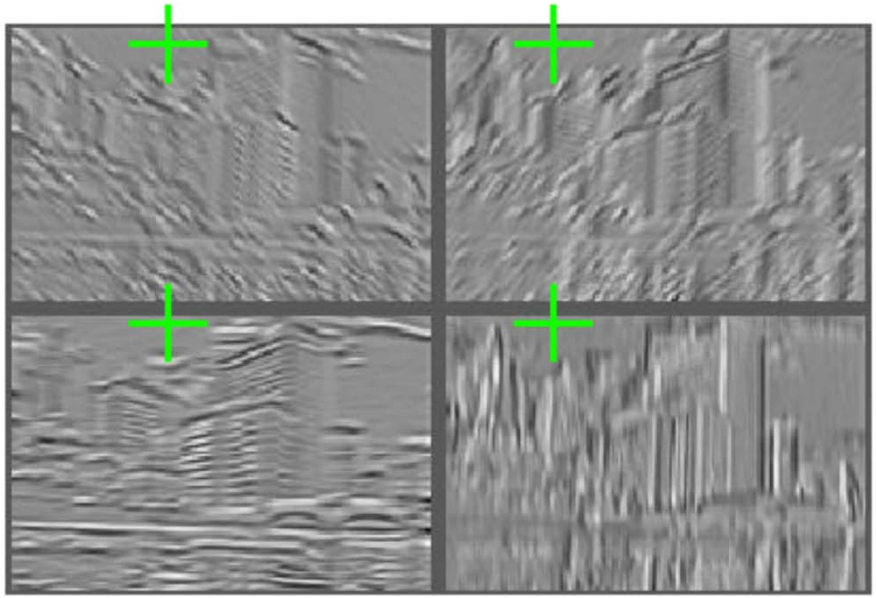
Normalization

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

Feature Maps

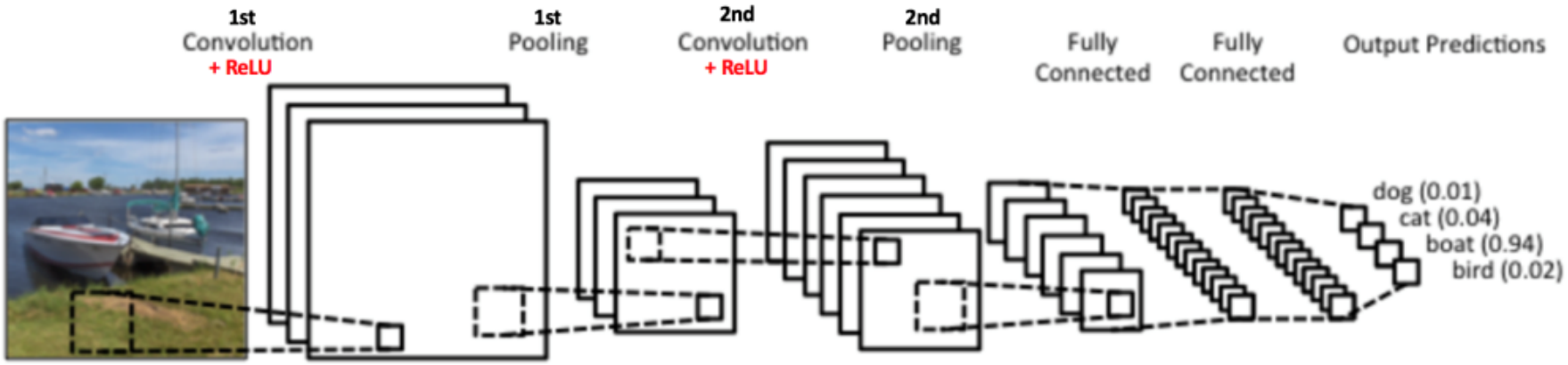


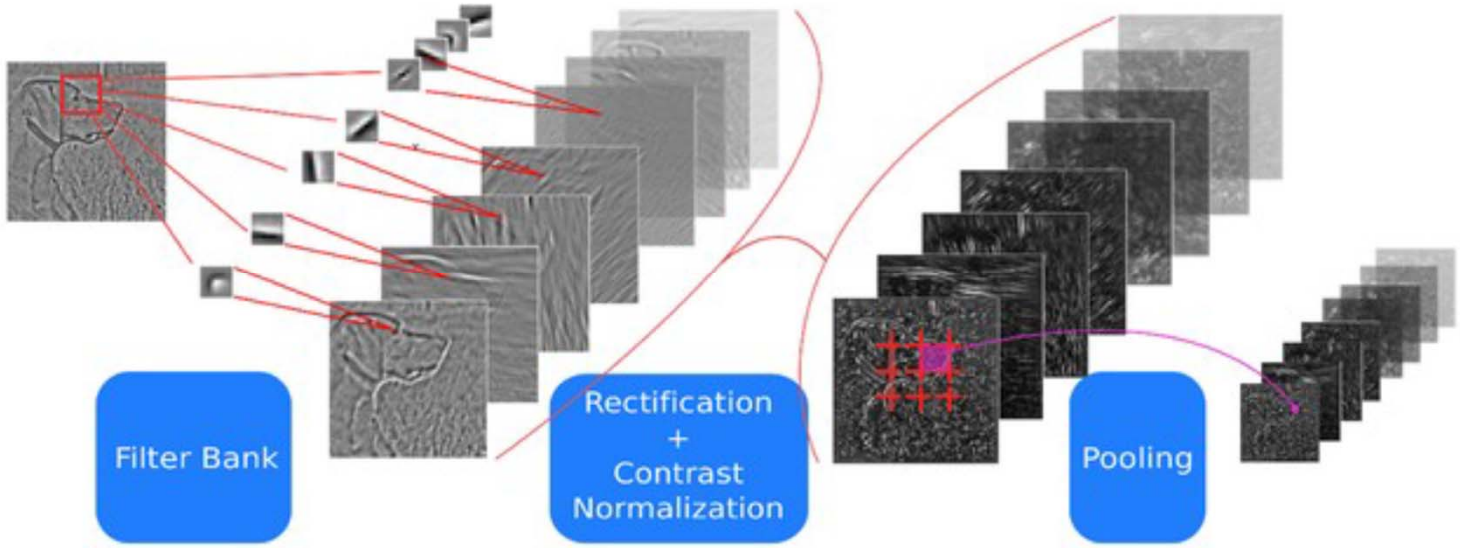
Feature Maps
After Contrast Normalization



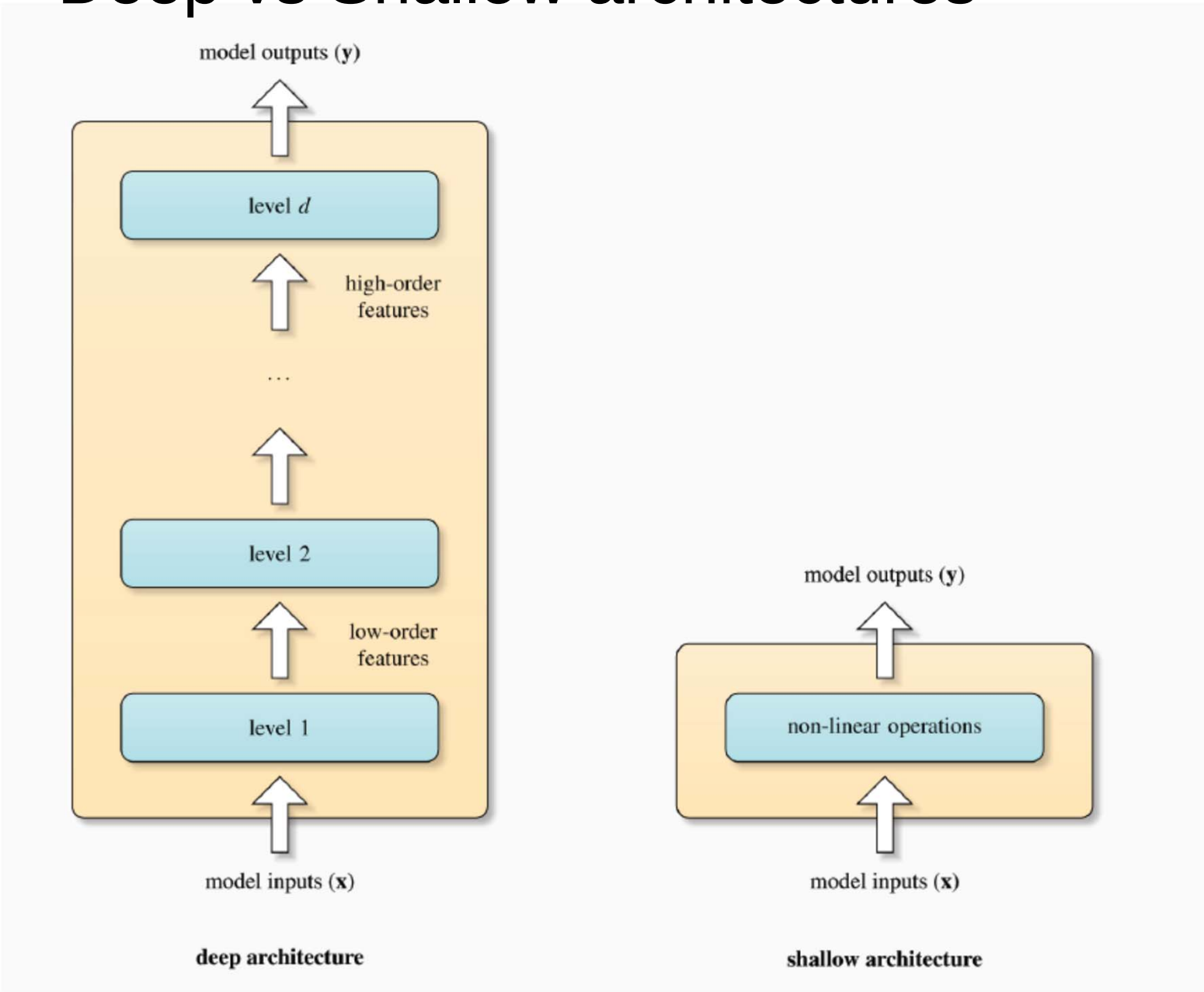
Slide: R. Fergus

CNN Architecture

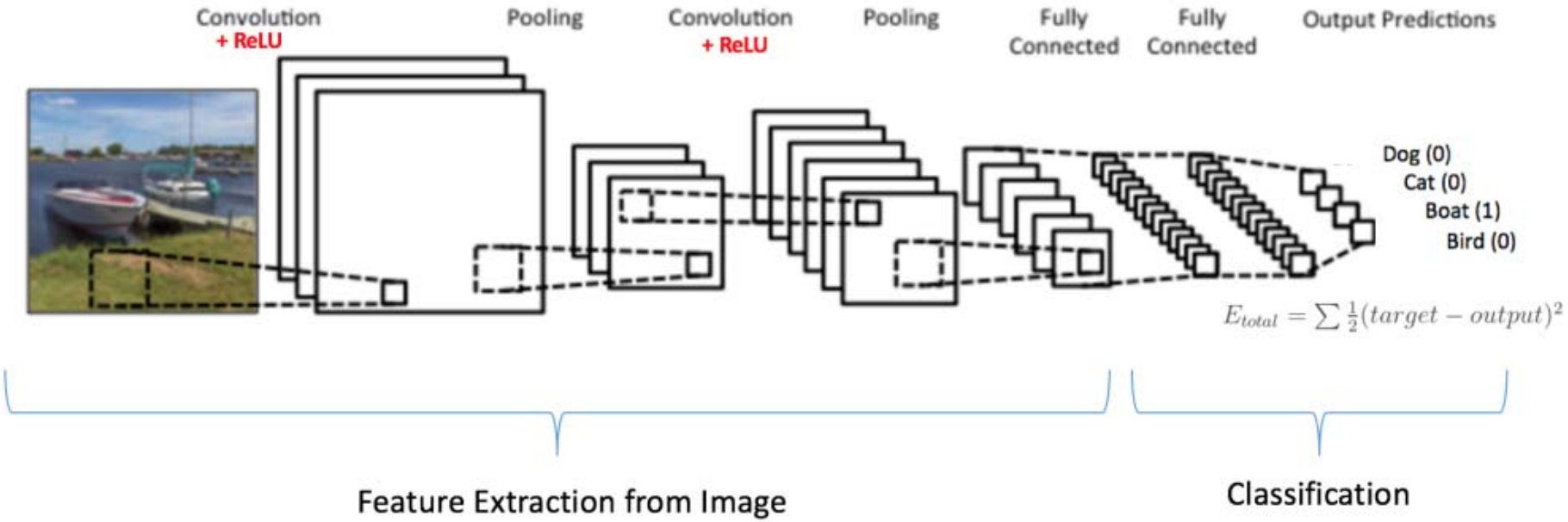




Deep vs Shallow architectures

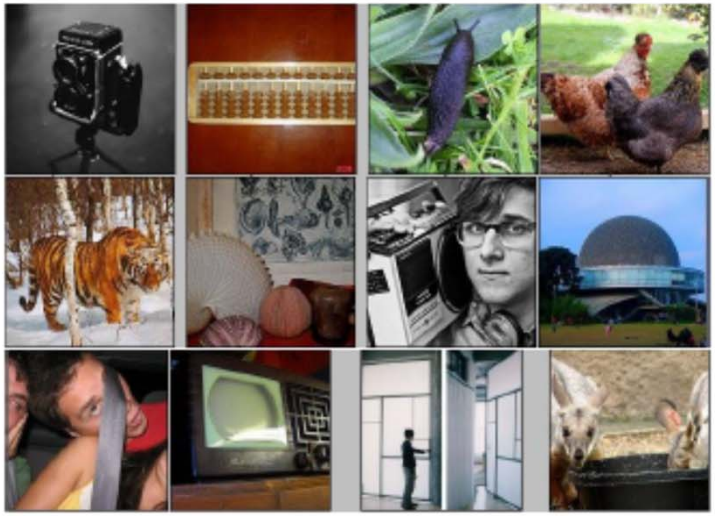


CNNs Training



Application: ImageNet

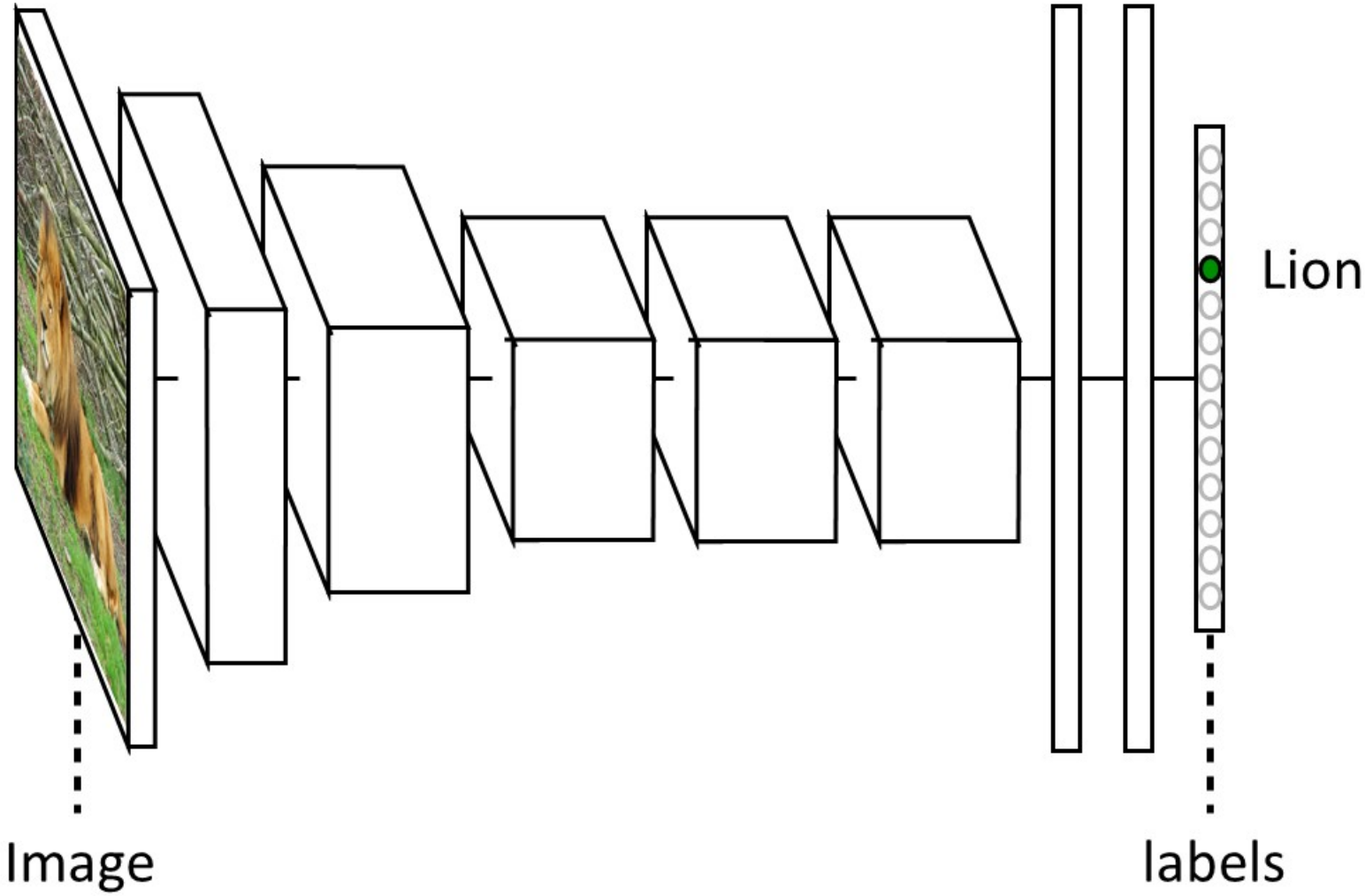
IMAGENET



[Deng et al. CVPR 2009]

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

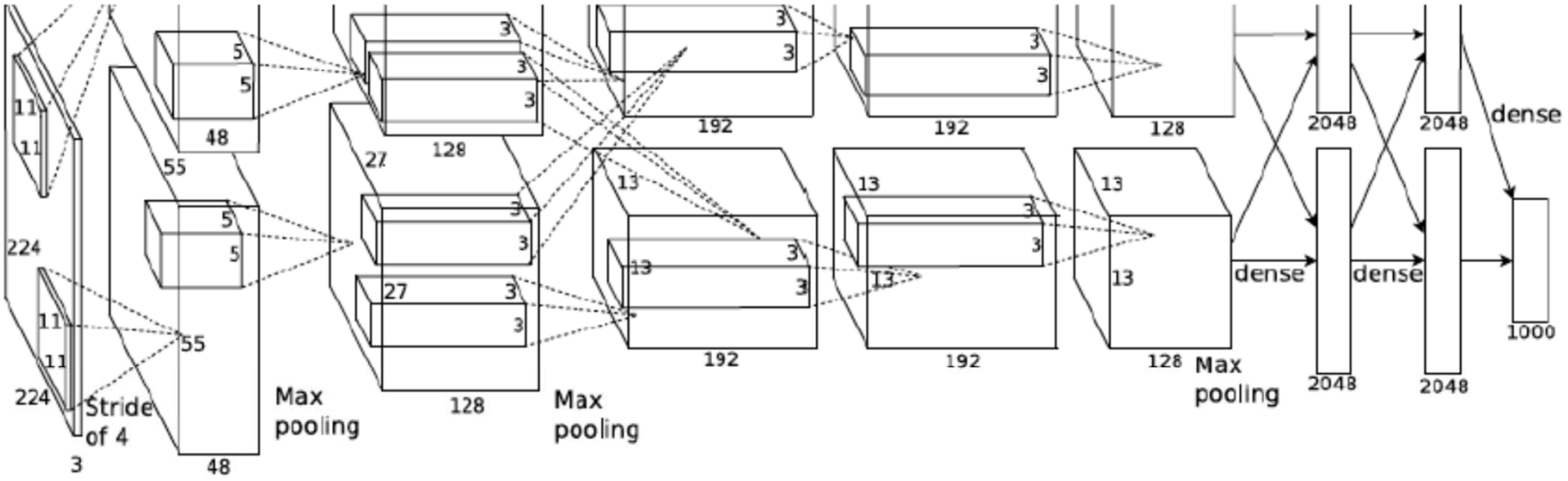
AlexNet architecture



Krizhevsky, Sutskever, Hinton — NIPS 2012

Krizhevsky et al. [NIPS 2012]

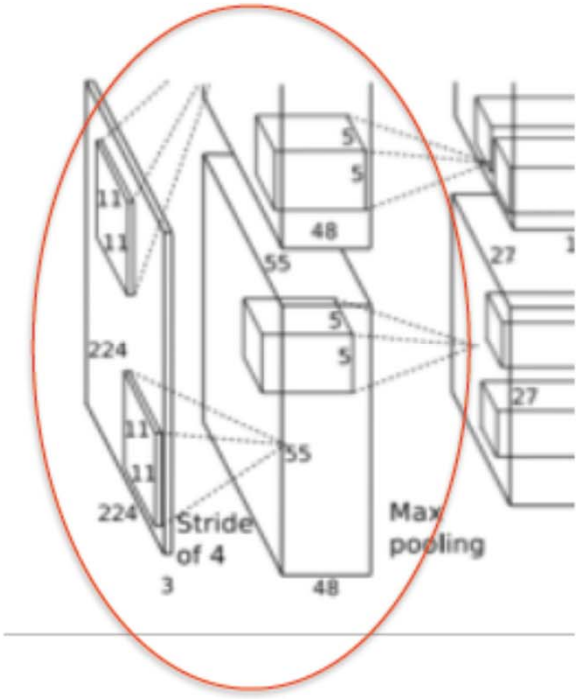
- Same model as LeCun'98 but:
  - Bigger model (8 layers)
  - More data (10^6 vs 10^3 images)
  - GPU implementation (50x speedup over CPU)
  - Better regularization (DropOut)



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

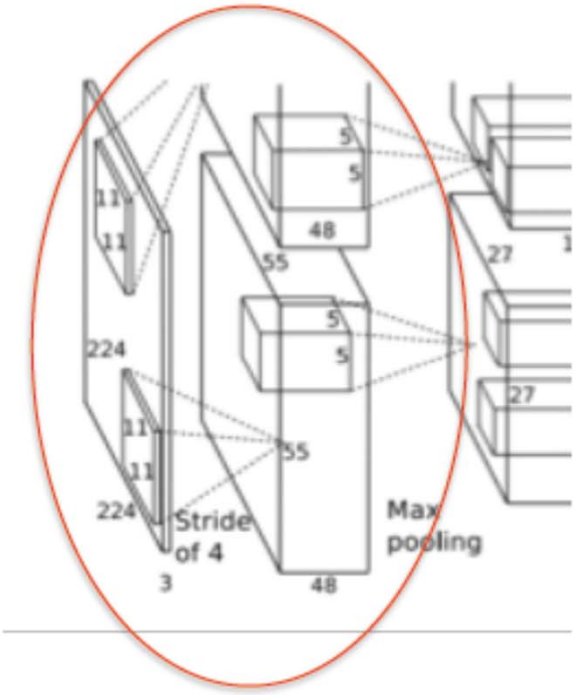
Slide: R. Fergus

AlexNet architecture: Layer 1 (convolutional)



- Images: 227x227x3
- F (receptive field size): 11
- S (stride) = 4
- Conv layer output: 55x55x96

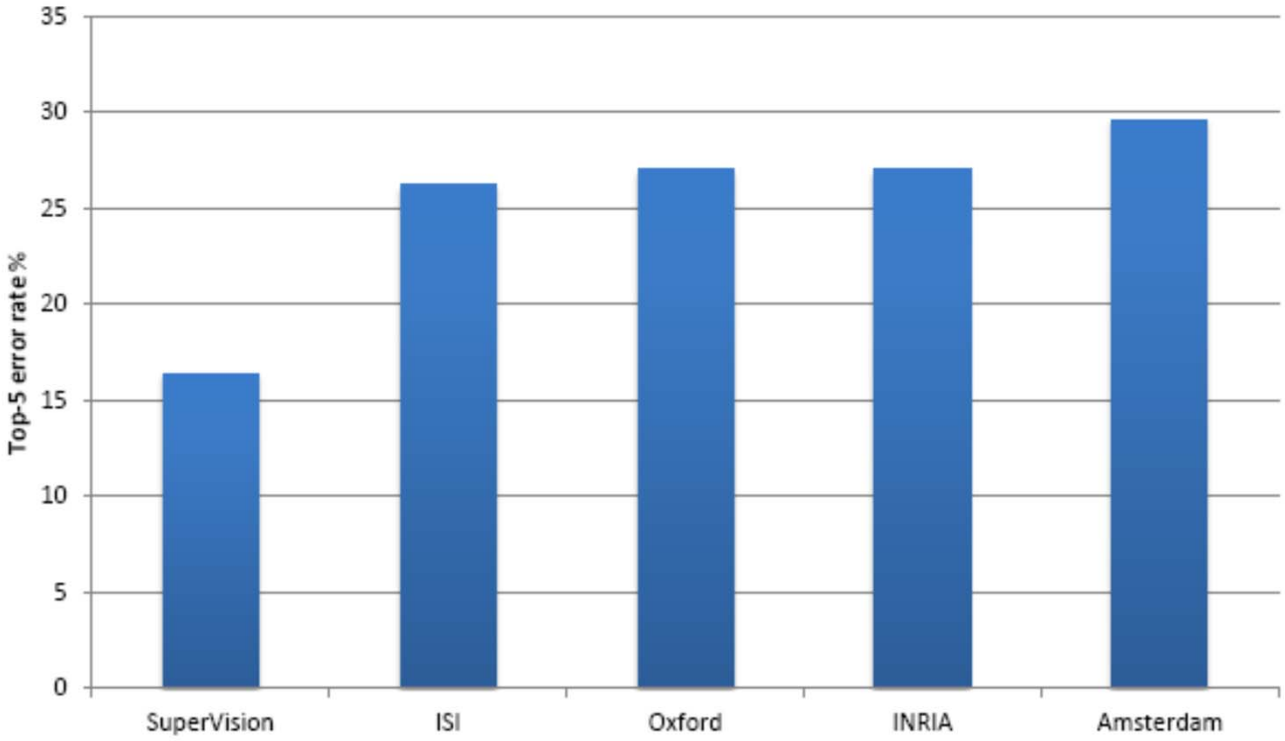
AlexNet architecture: Layer 1 (convolutional)



- 55*55*96 = 290,400 neurons
- each has 11*11*3 = 363 weights and 1 bias
- 290400 * 364 = 105,705,600 paramaters on the first layer of the AlexNet alone!

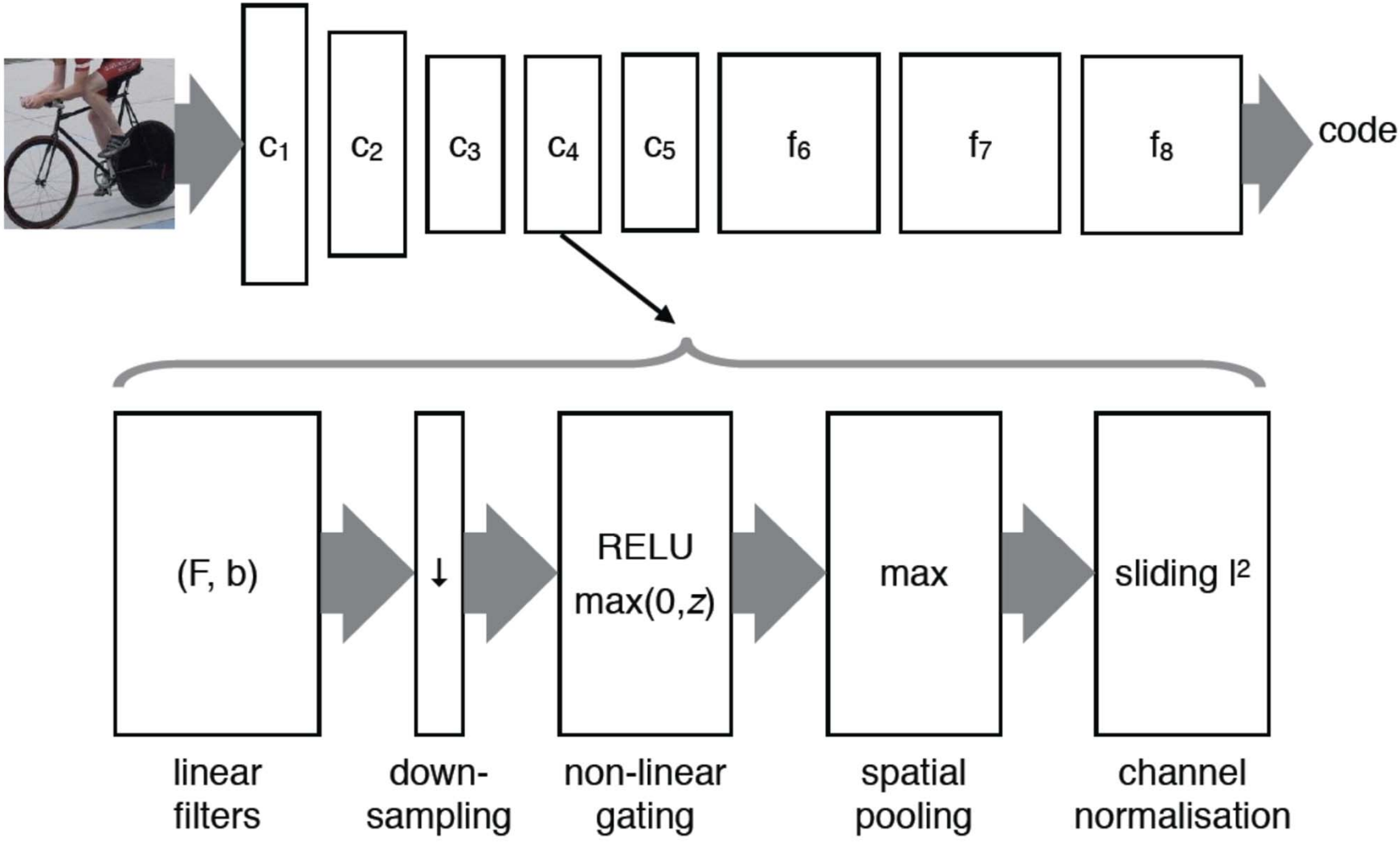
ImageNet Classification 2012

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

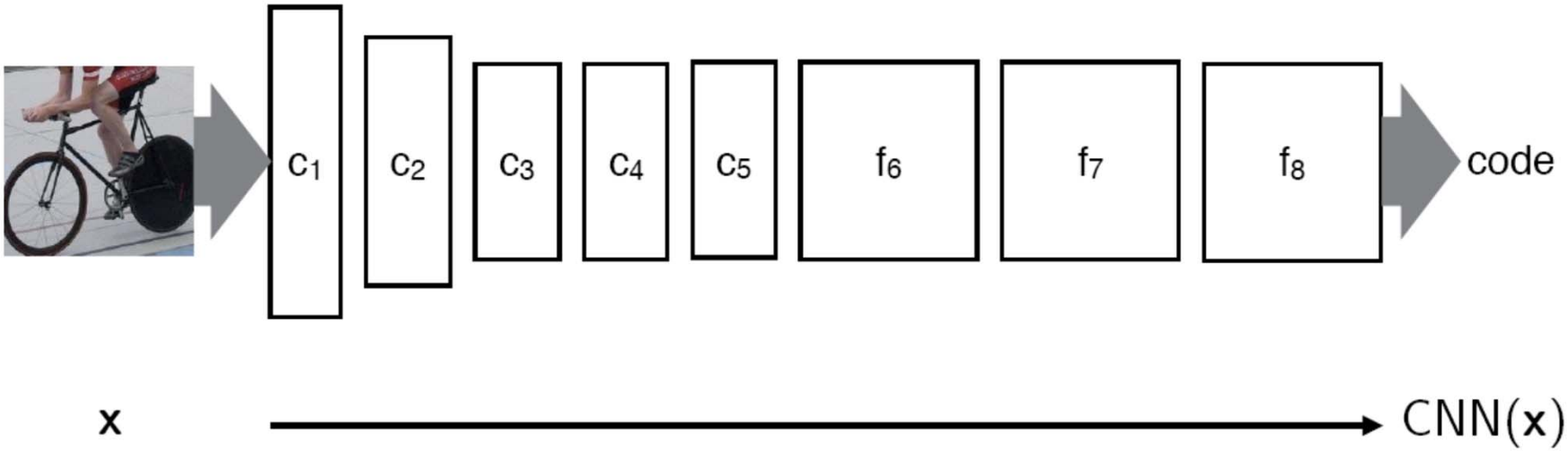


CNNs: Training procedure

Convolutional layers



Convolutional neural networks (CNNs)



From left to right

- decreasing spatial resolution
- increasing feature dimensionality

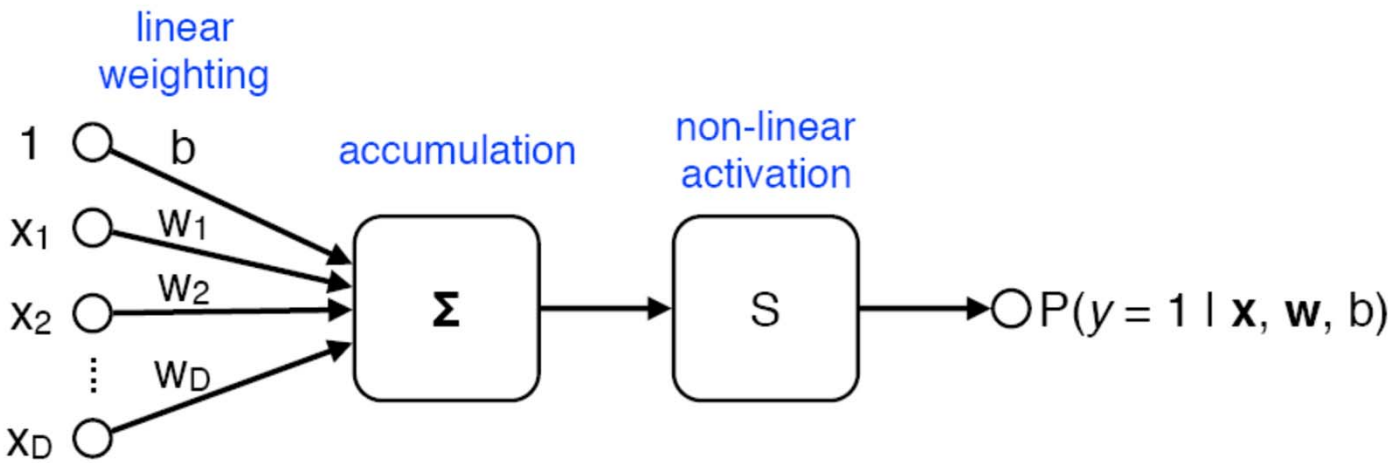
Fully-connected layers

- same as convolutional, but with 1×1 spatial resolution
- contain most of the parameters

Perceptron

[Rosenblatt 57]

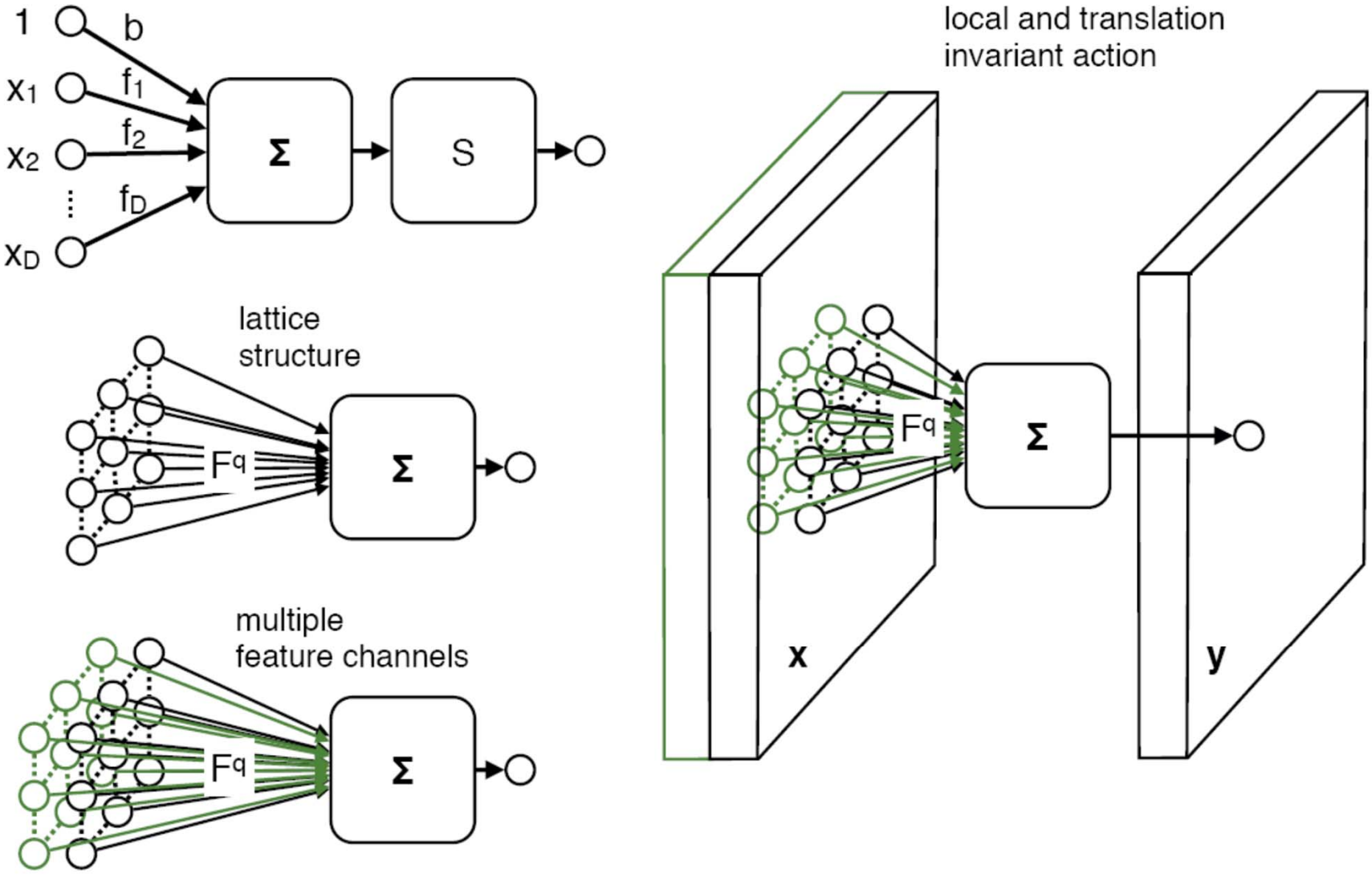
The goal is estimating the posterior probability of the binary label *y* of a vector **x**:



$$\text{output} = \begin{cases} 0 & \text{if } w \cdot x + b \leq 0 \\ 1 & \text{if } w \cdot x + b > 0 \end{cases}$$

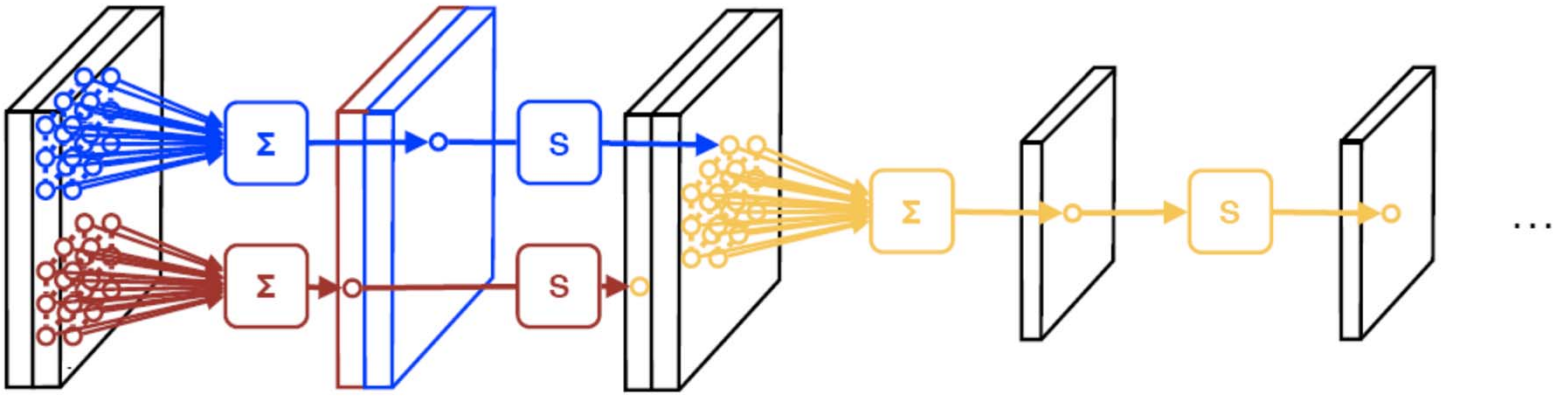
Linear convolution

As a neural network



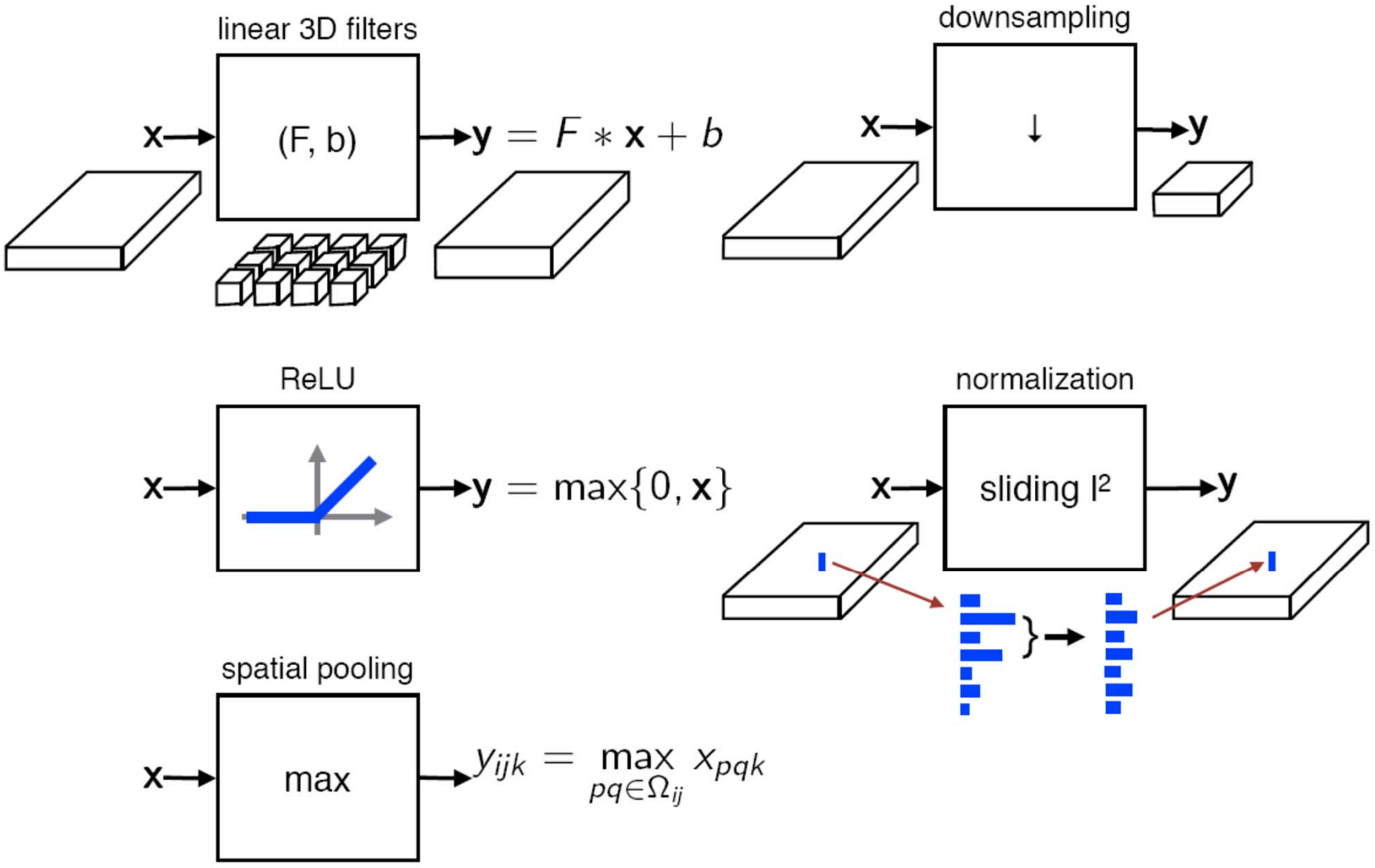
Linear / non-linear chains

The basic blueprint of most architectures



filtering ReLU filtering & downsampling ReLU ...

CNN components

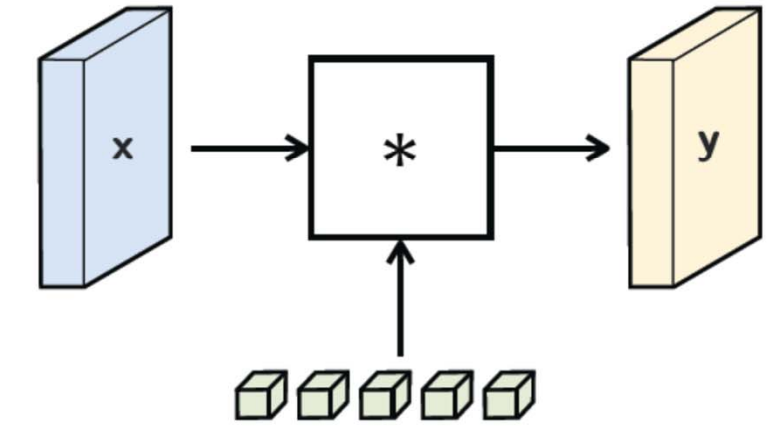


Convolutional Neural Network (CNN)

A sequence of local & shift invariant layers

Example: convolution layer

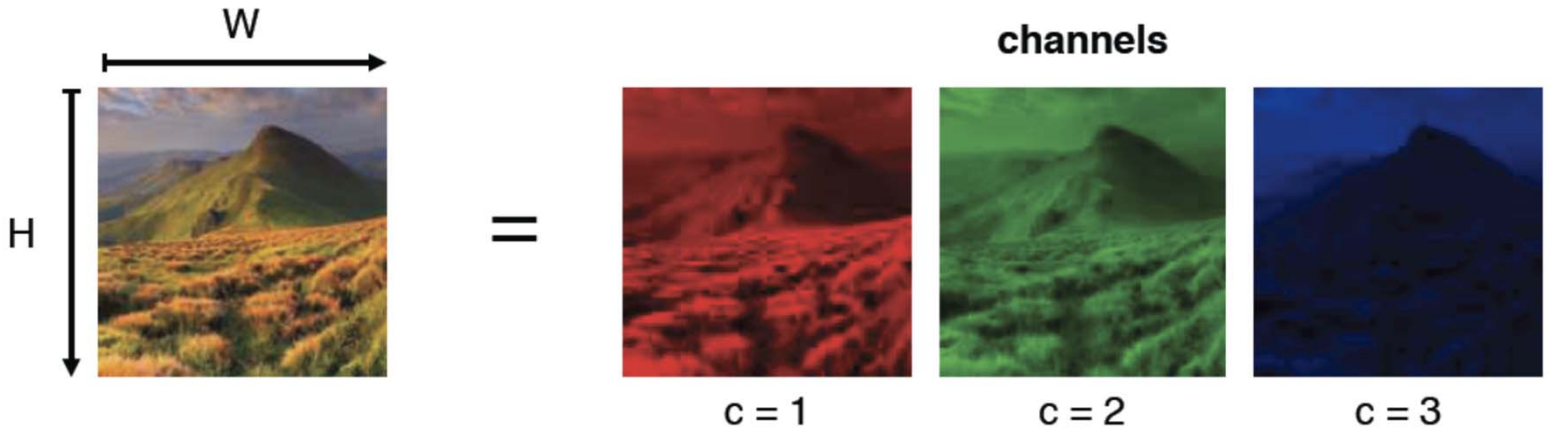
$$\mathbf{y} = F * \mathbf{x} + b$$

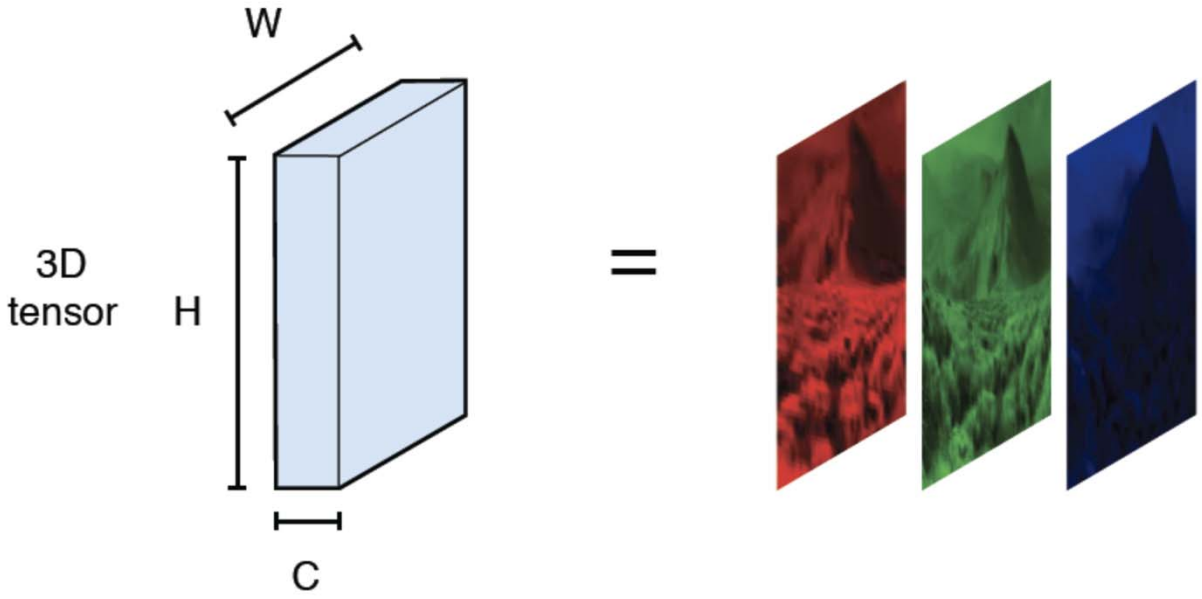


input data **x** filter bank *F* output data **y**

Data = 3D tensors

There is a vector of feature channels (e.g. RGB) at each spatial location (pixel).



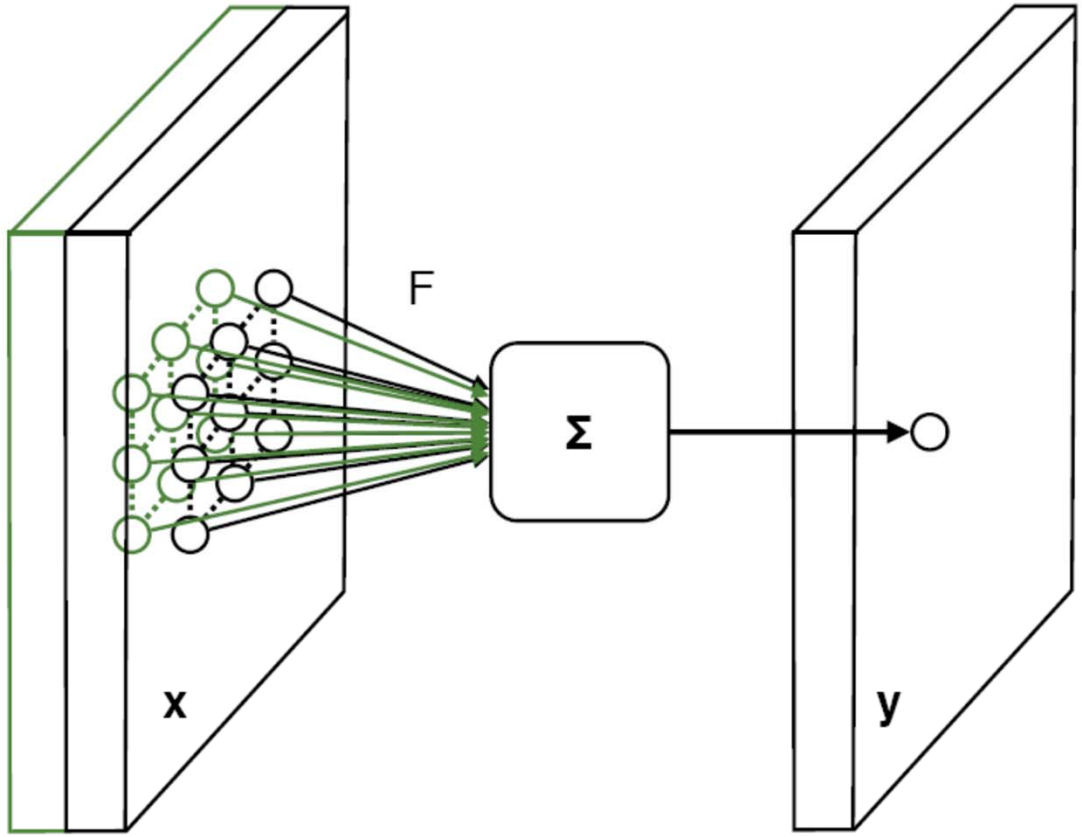


Convolution with 3D filters

Each filter acts on multiple input channels

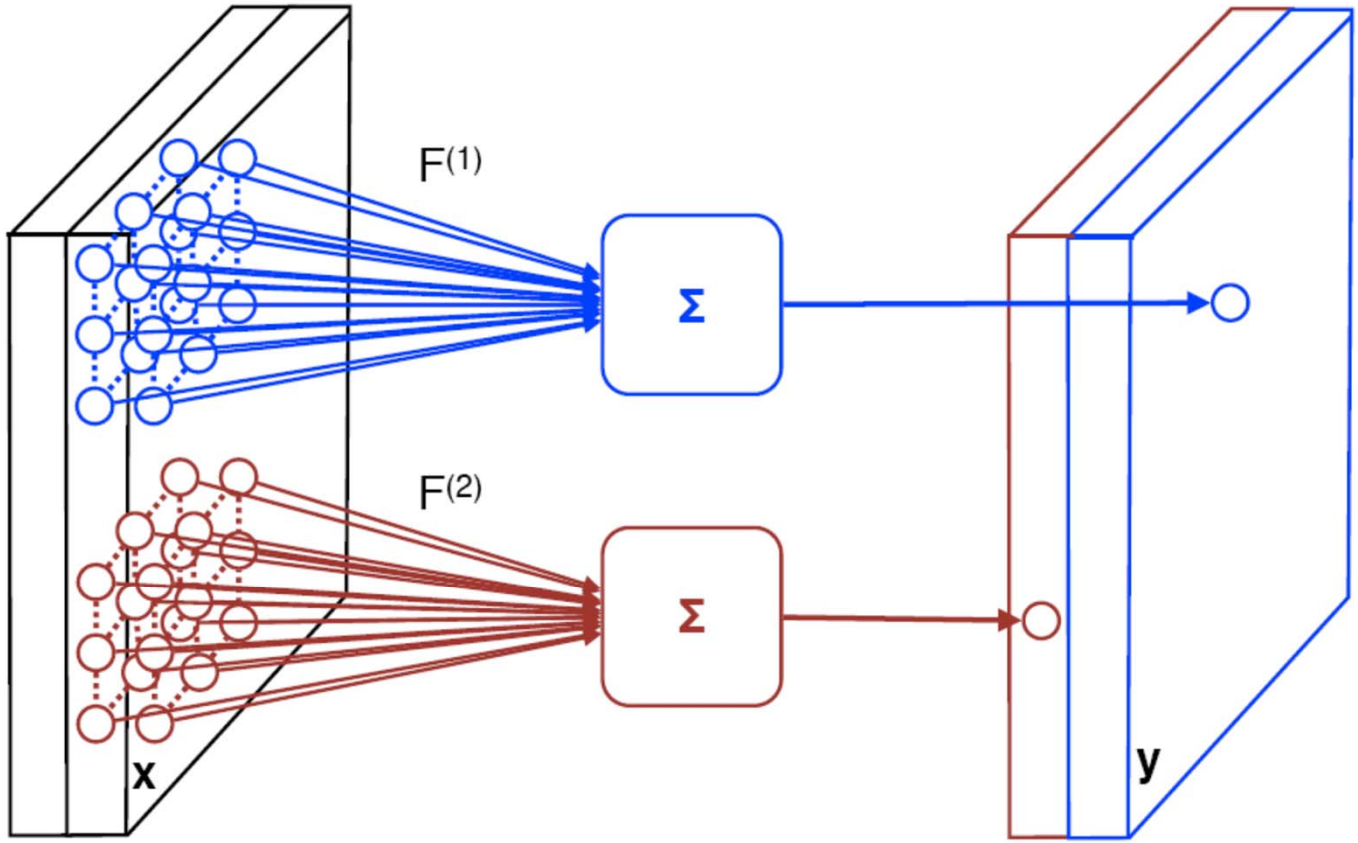
Local
Filters look locally

Translation invariant
Filters act the same everywhere



Filter banks

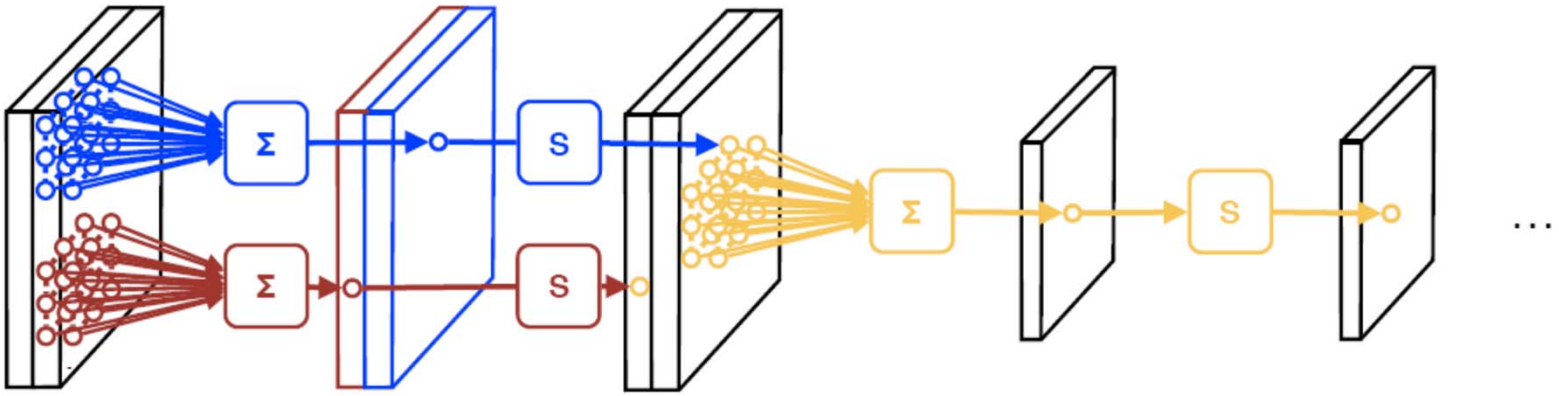
Multiple filters produce multiple output channels



One filter = one output channel

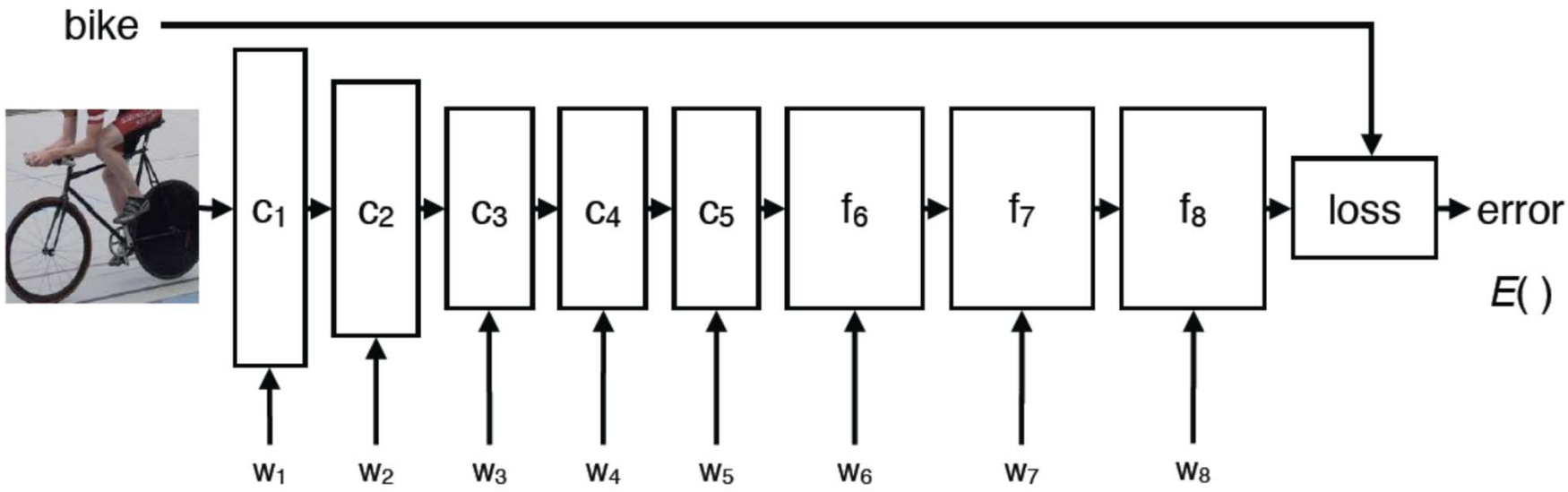
Linear / non-linear chains

The basic blueprint of most architectures



filtering ReLU filtering & downsampling ReLU ...

Learning a CNN

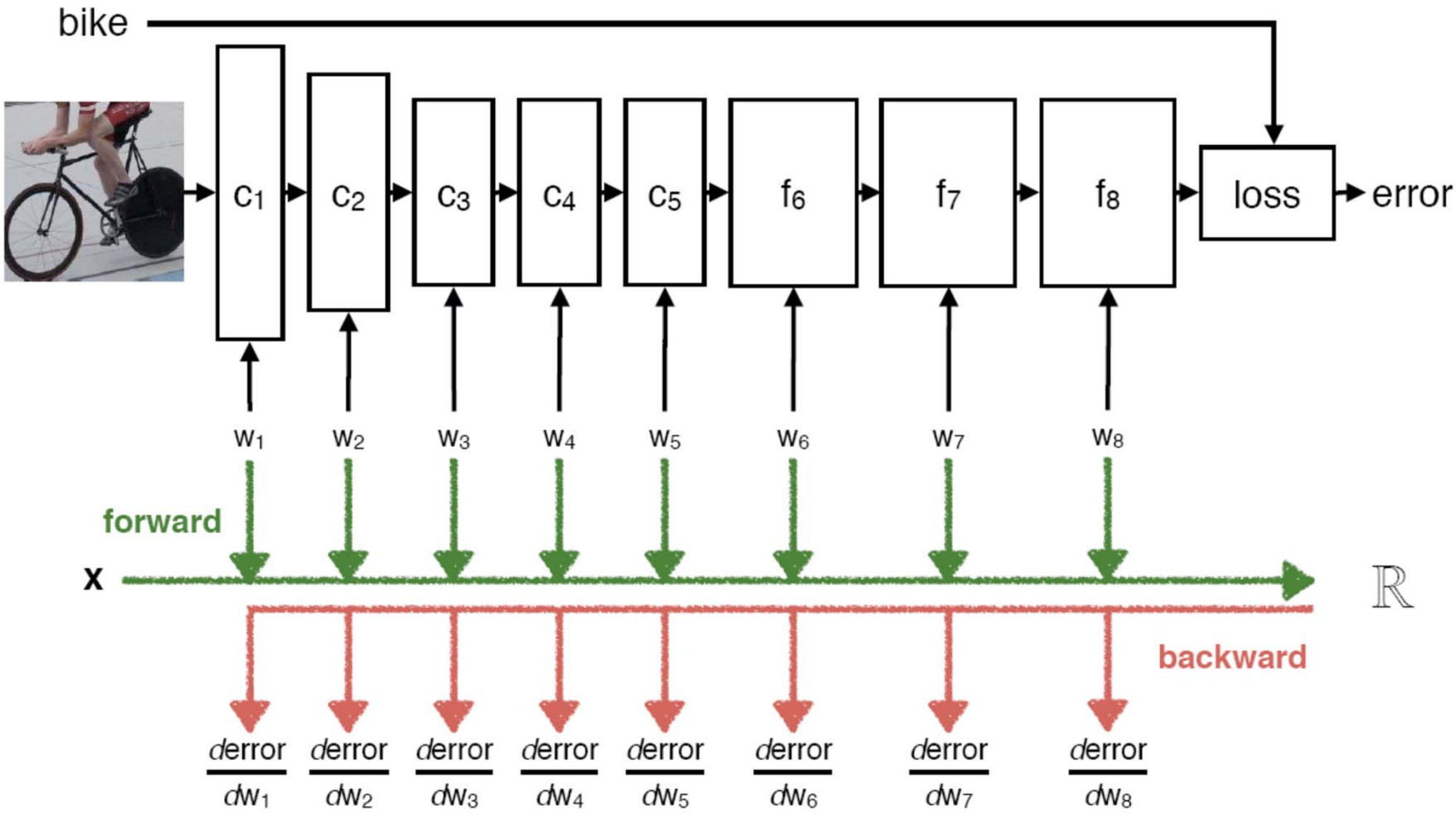


$$\text{argmin}\, E(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_8)$$

Stochastic gradient descent
(with momentum, dropout, …)

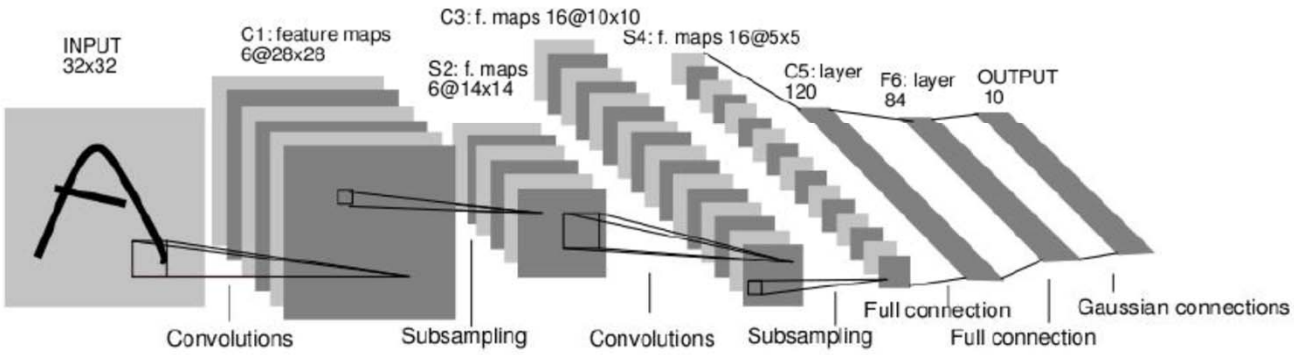
Backpropagation

Compute derivatives using the chain rule

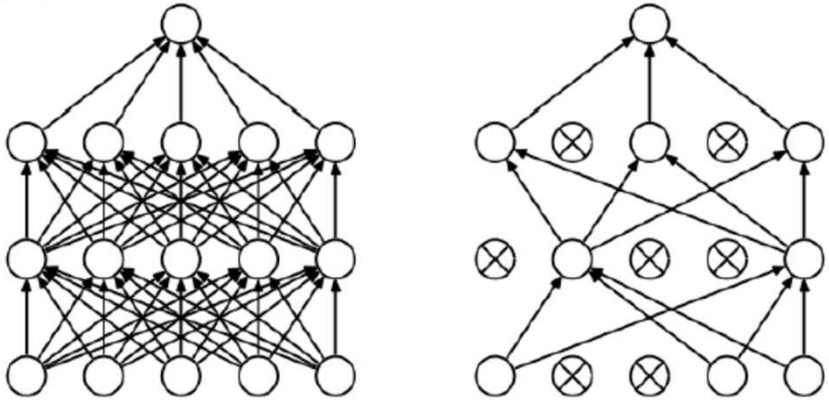


Today: two regularization methods

Convolutional Networks



Dropout

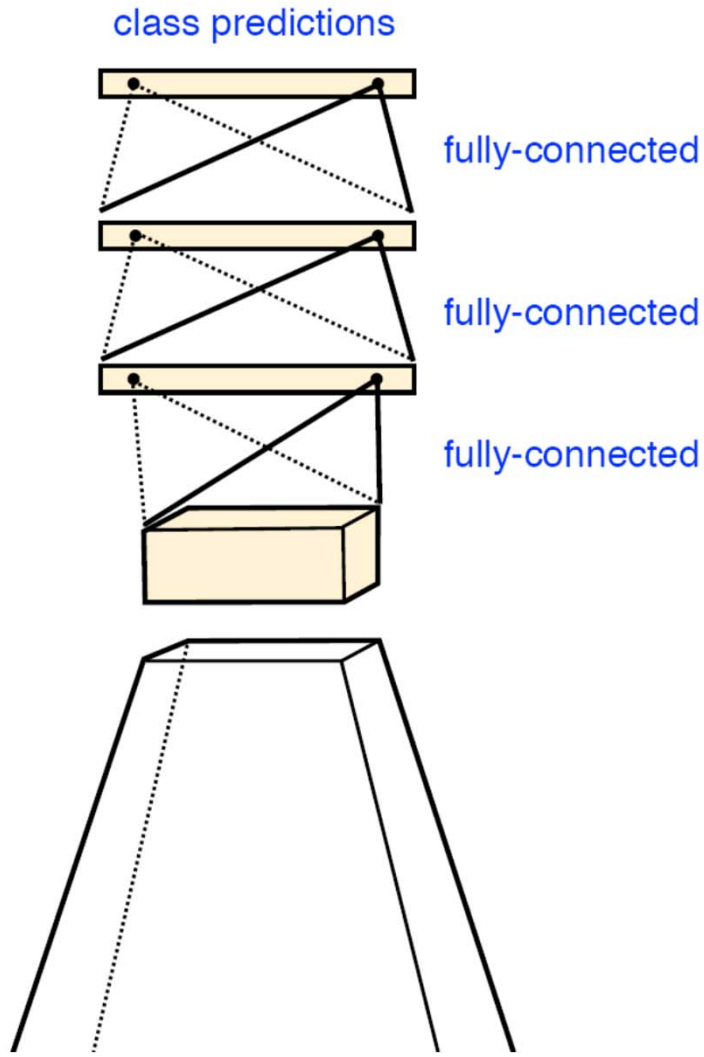


(a) Standard Neural Net

(b) After applying dropout.

Fully connected layers

Global receptive field

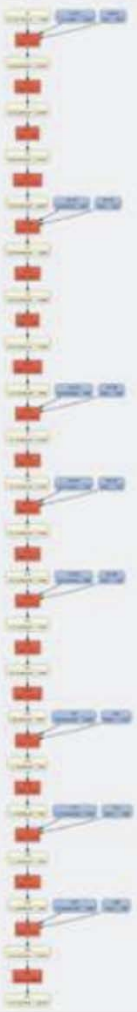


How deep is enough?

AlexNet (2012)

5 convolutional layers

3 fully-connected layers



How deep is enough?

AlexNet (2012)

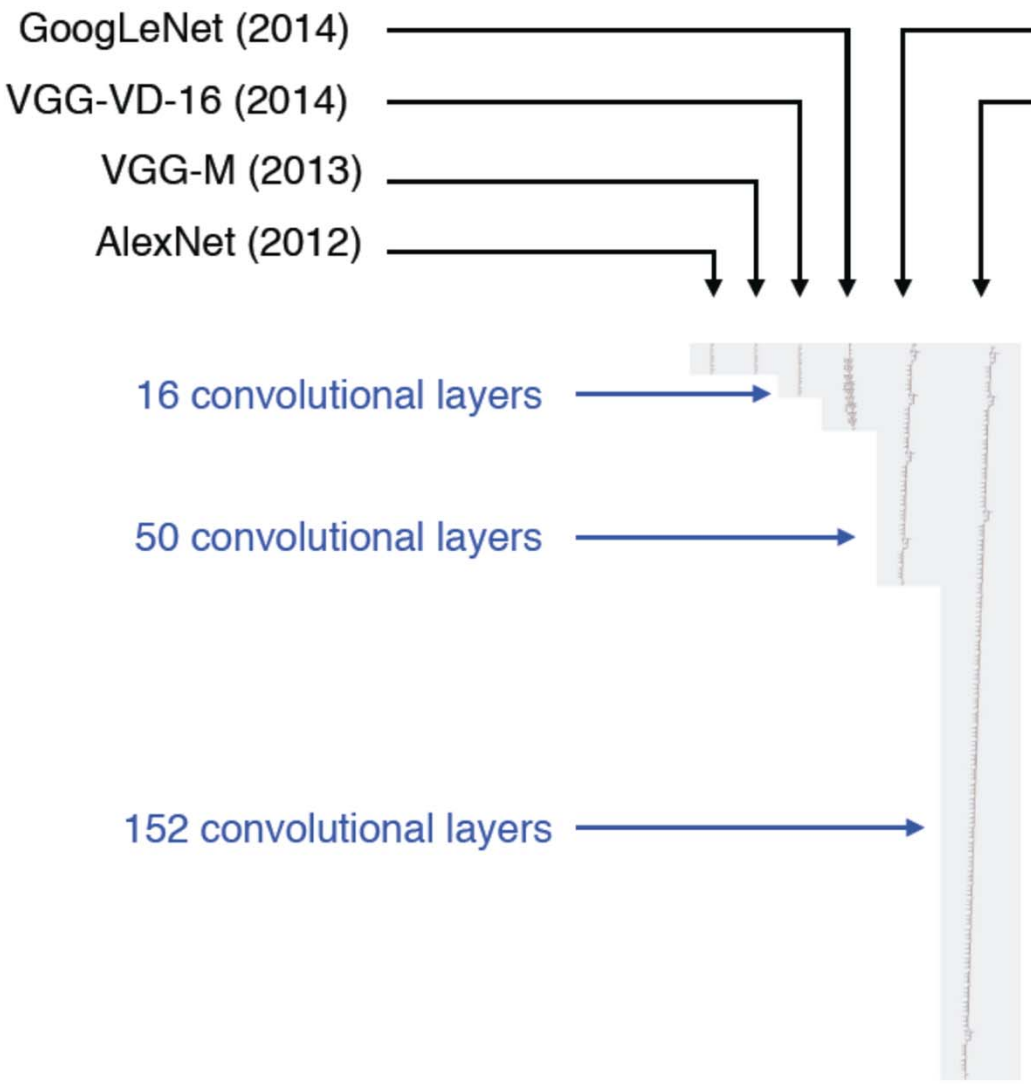
VGG-M (2013)

VGG-VD-16 (2014)

How deep is enough?

AlexNet (2012) VGG-M (2013) VGG-VD-16 (2014) GoogLeNet (2014)

How deep is enough?



Krizhevsky, I. Sutskever, and G. E. Hinton. *ImageNet classification with deep convolutional neural networks*. In Proc. NIPS, 2012.

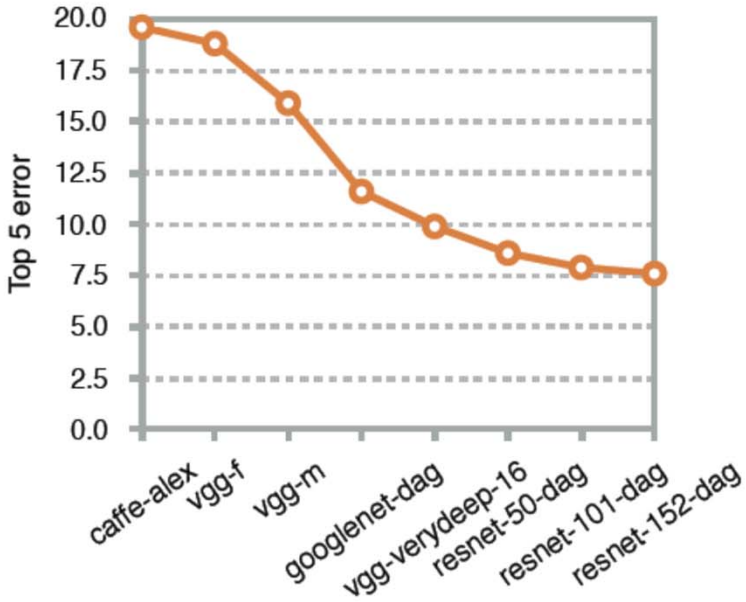
C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. *Going deeper with convolutions*. In Proc. CVPR, 2015.

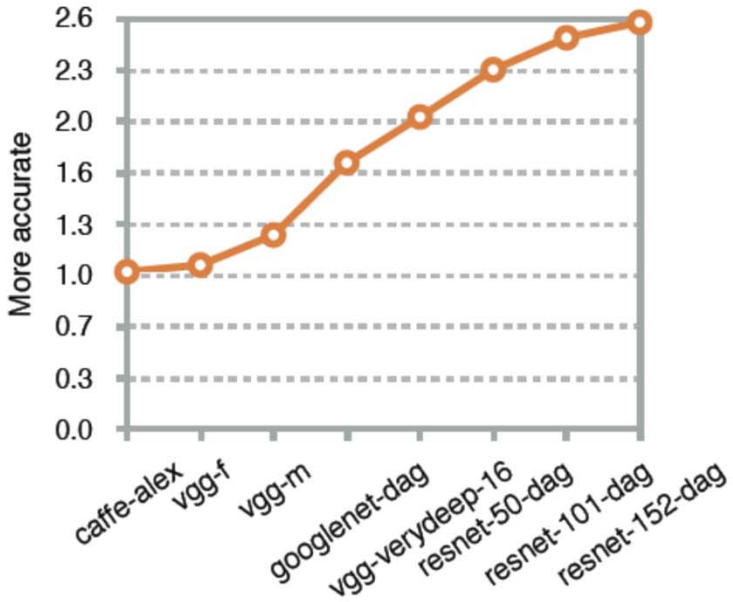
K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition*. In Proc. ICLR, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. *Deep residual learning for image recognition*. In Proc. CVPR, 2016.

Accuracy

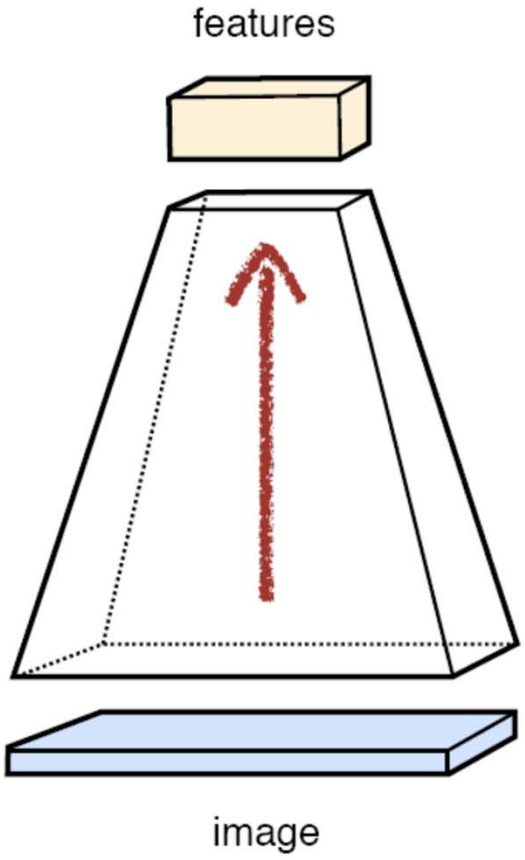
3 × more accurate in 3 years





Design guidelines

Guideline 1: *Avoid tight bottlenecks*



From bottom to top

- The *spatial resolution* $H \times W$ decreases
- The *number of channels* C increases

Guideline

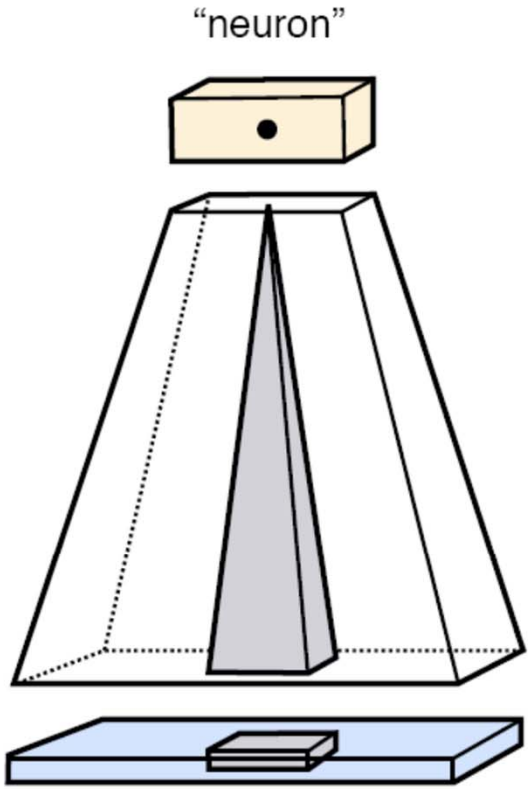
- Avoid tight information bottleneck
- Decrease the *data volume* $H \times W \times C$ slowly

K. Simonyan and A. Zisserman. *Very deep convolutional networks for large-scale image recognition.* In Proc. ICLR, 2015.

C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens. *Rethinking the inception architecture for computer vision.* In Proc. CVPR, 2016.

Receptive field

Must be large enough



Receptive field of a neuron

- The image region influencing a neuron
- Anything happening outside is invisible to the neuron

Importance

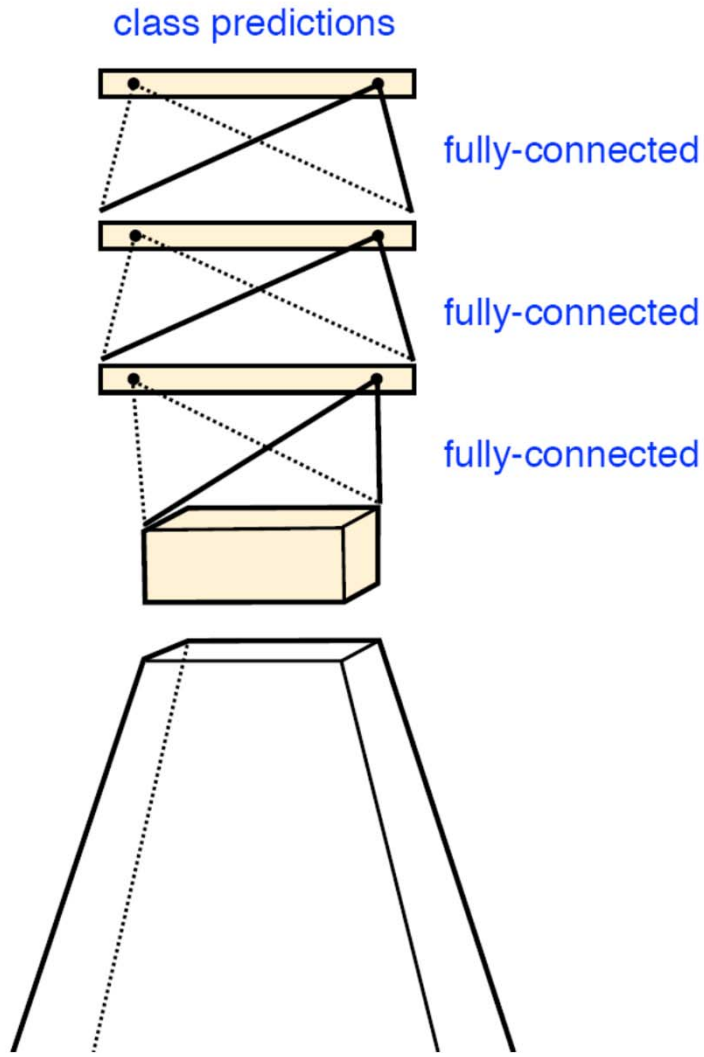
- Large image structures cannot be detected by neurons with small receptive fields

Enlarging the receptive field

- Large filters
- Chains of small filters

Fully connected layers

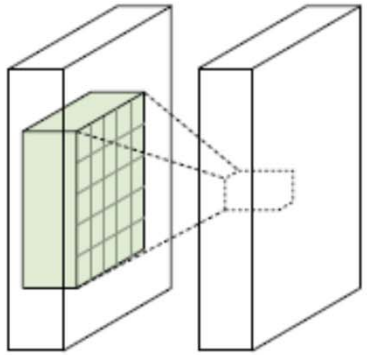
Global receptive field



Design guidelines

Guideline 2: *Prefer small filter chains*

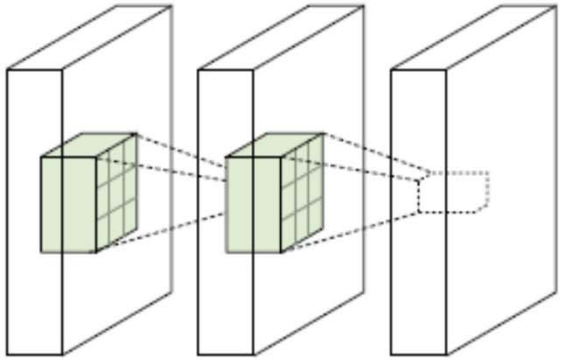
One big filter bank



5 × 5 filters
+ ReLU

prefer

Two smaller filter banks



3 × 3 filters
+ ReLU

3 × 3 filters
+ ReLU

Benefit 1: less parameters, possibly faster

Benefit 2: same receptive field of a bigger filter

Benefit 3: packs two non-linearities (ReLUs)

Applications

Semantic image segmentation

Label individual pixels

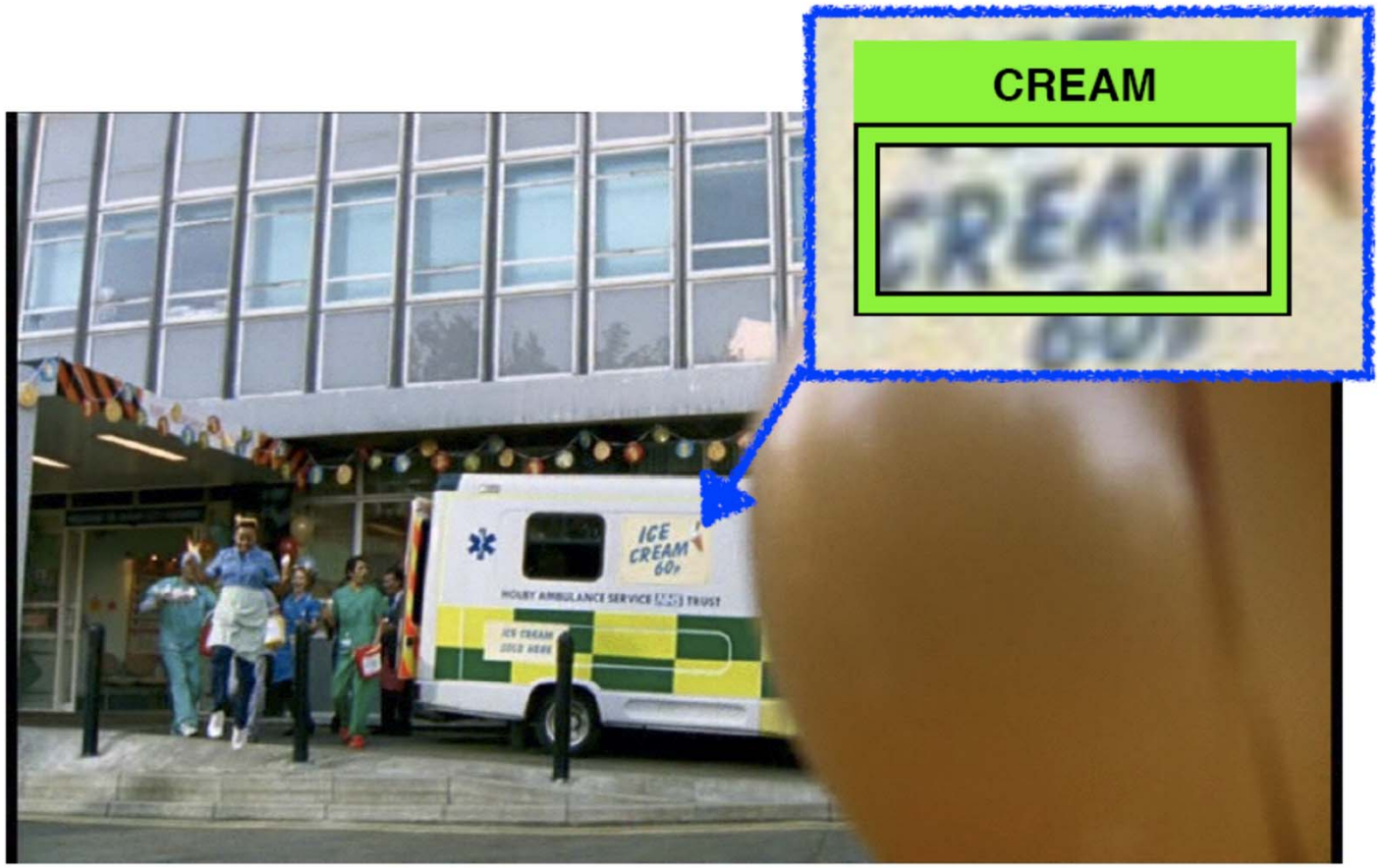
Face analysis

Detection, verification, recognition, emotion, 3D fitting

E.g. VGG-Face

Text spotting

Detection, word recognition, character recognition

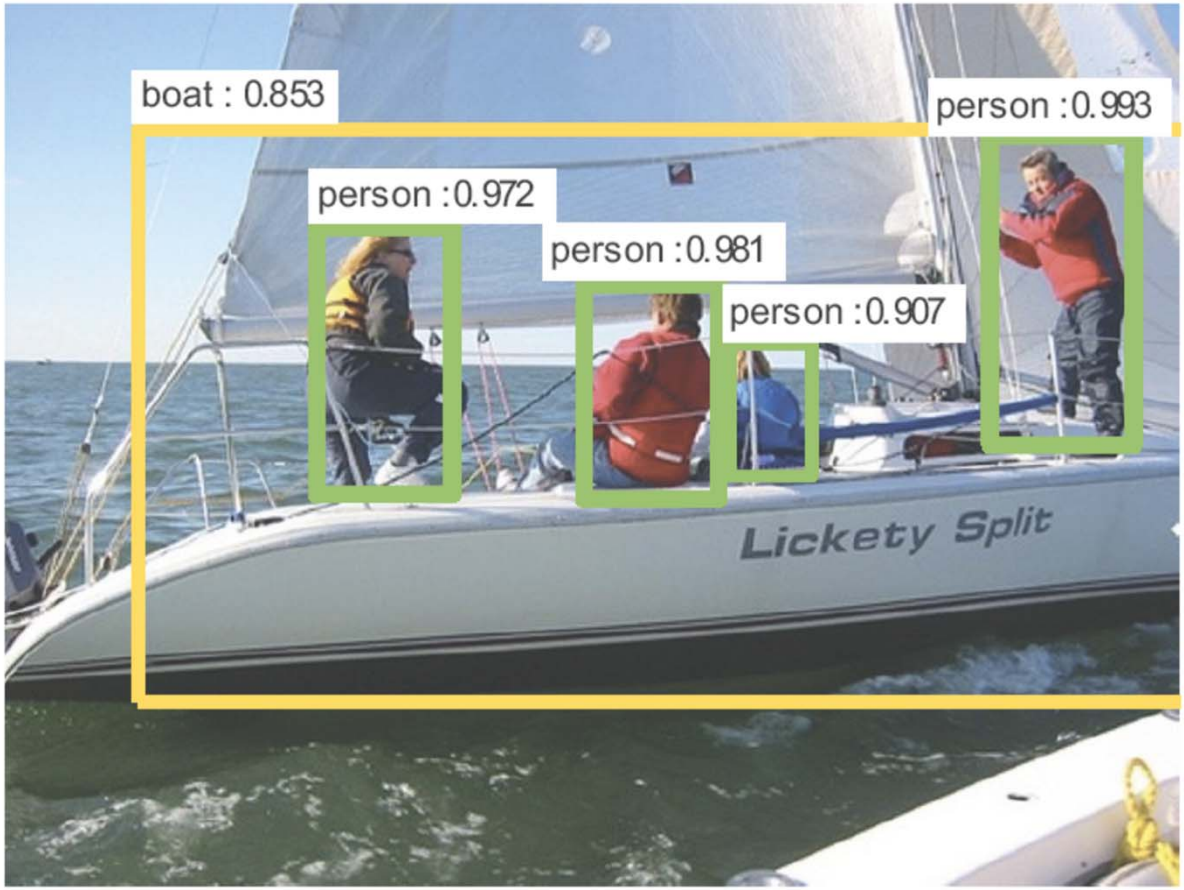


E.g. SynthText and VGG-Text

http://zeus.robots.ox.ac.uk/textsearch/#/search/

Object detection

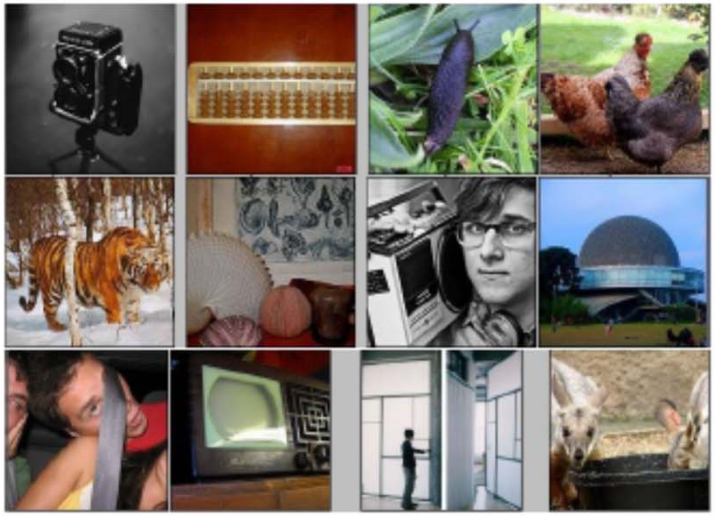
Extract individual object instances



Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation
R. Girshick, J. Donahue, T. Darrell, J. Malik, CVPR 2014

Application: ImageNet

IMAGENET



[Deng et al. CVPR 2009]

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

Bibliography

Slides extracted from the following tutorials:

Andrea Vedaldi (2017). Tutorials, codes and presentations
www.robots.ox.ac.uk/~vedaldi/

G. Taylor et al. “Deep learning for computer vision” CVPR tutorial 2014

M.A. Ranzato “Supervised Deep Learning” CVPR turorial 2014