Lecture 1: Deep Learning: Introduction

Overview

This class is about:

- deep learning
- application in computer vision and graphics
- applications in natural language processing
- deep reinforcement learning

It will include:

- 12 lectures
- 5 seminars
- 4 big assignments
- 1 project
- 1 exam

2006

CAPTCHA

asirra

Computer vision = 60%

$0.6^{12} = 0.00217$

2014

Completed • Swag • 215 teams

Dogs vs. Cats

Wed 25 Sep 2013 – Sat 1 Feb 2014 (8 months ago)

Dashboard

Private Leaderboard - Dogs vs. Cats

This competition has completed. This leaderboard reflects the final standings.

See someone

#	Δ1w	Team Name * in the money	Score	Entries	Last Submission UTC (Best – Las
1	–	Pierre Sermanet *	0.98914	5	Sat, 01 Feb 2014 21:43:19 (-
2	↑26	orchid *	0.98309	17	Sat, 01 Feb 2014 23:52:30
3	–	Owen	0.98171	15	Sat, 01 Feb 2014 17:04:40 (-
4	new	Paul Covington	0.98171	3	Sat, 01 Feb 2014 23:05:20
5	↓3	Maxim Milakov	0.98137	24	Sat, 01 Feb 2014 18:20:58

$$0.989^{12} = 0.875$$

2014

Microsoft Research

Search Mi

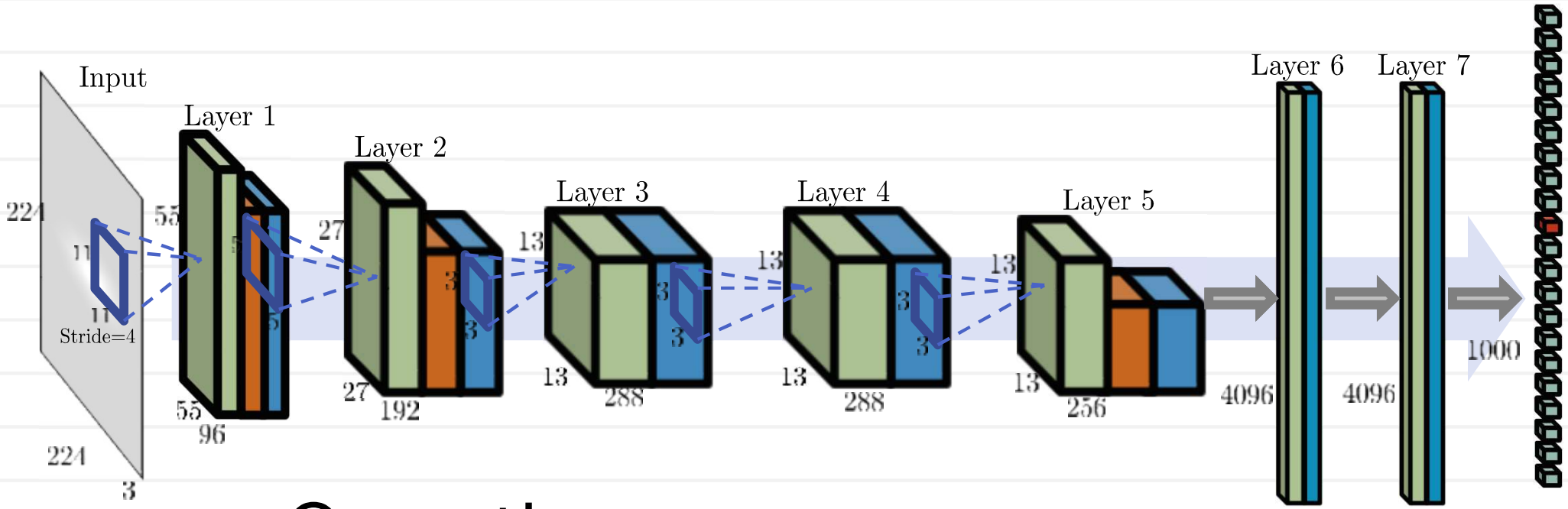
Our research Connections Careers About us

All Downloads Events Groups News People Projects Publications

ASIRRA

After 8 years of operation, Asirra is shutting down effective October 1, 2014. Thank you to all of our users!

The winner: convolutional networks



Operations:

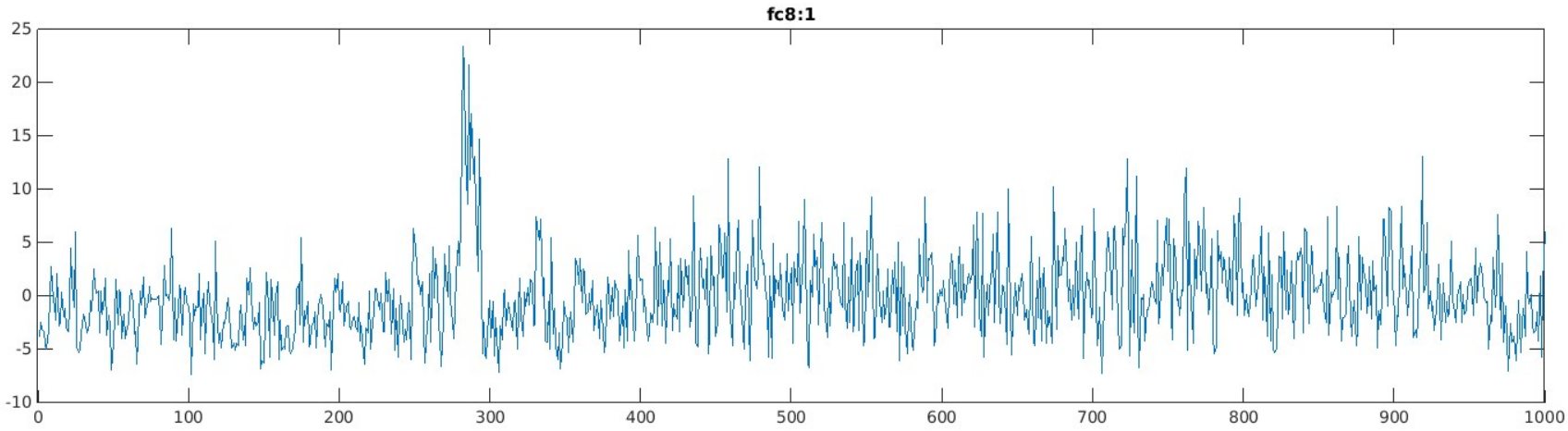
generalized convolutions

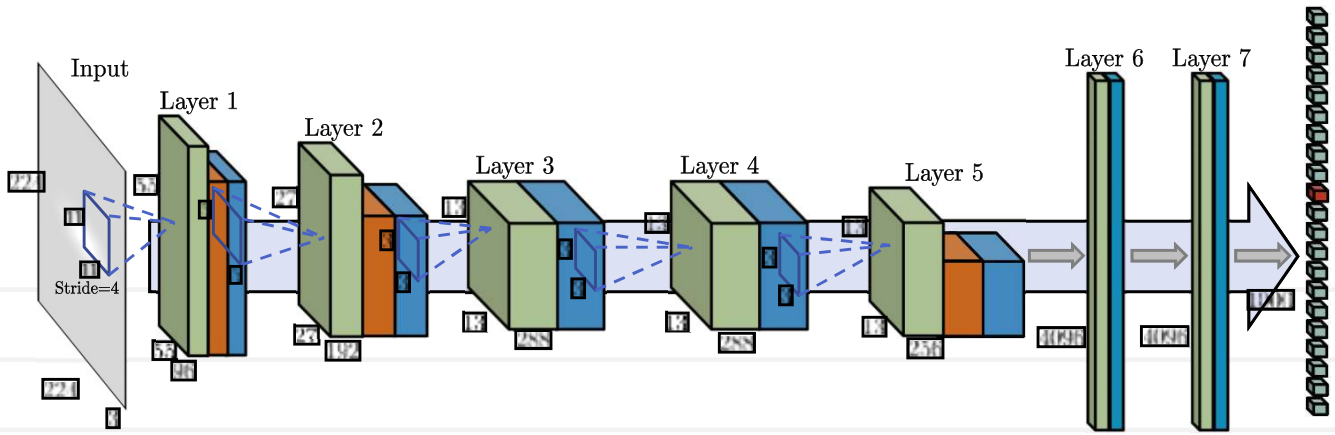
pooling (image resizing)

elementwise non-linearity

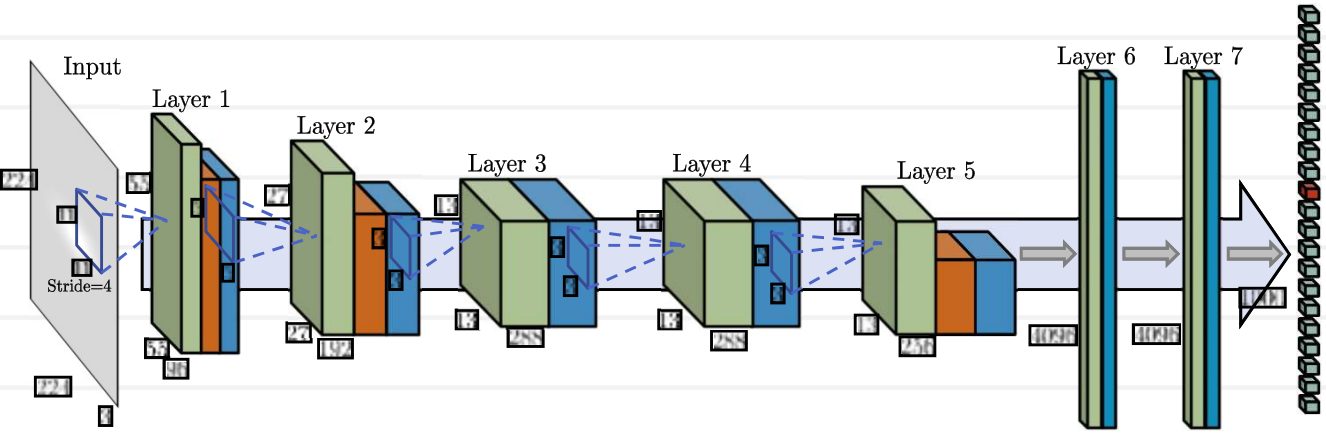
matrix multiplication

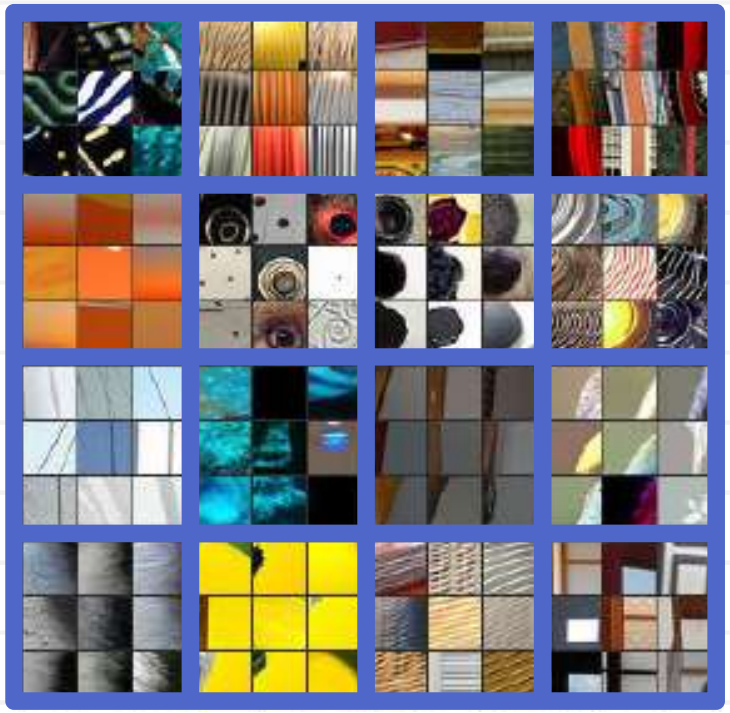
Representations





Left-to-right = "smarter"



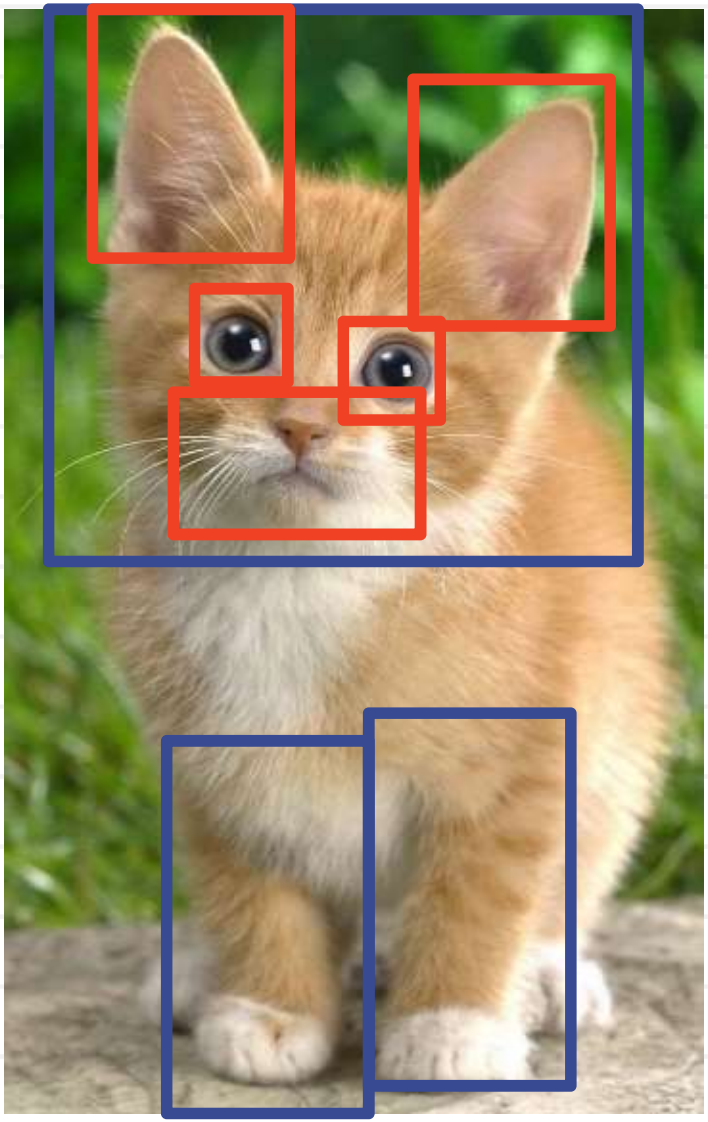


Layer 1
[Zeiler Fergus 14]

Layer 2

Layer 5

High level vision is part-based



Learning intermediate representations

- The essence of modern "deep learning"
- Has been done all along before "deep learning" revolution (will review today)
- Is essential for machine intelligence (will discuss why)
- Can be done via supervised, unsupervised and other types of learning (will briefly discuss how)

Supervised learning

$$\{x_1\ x_2\ x_3\ \ldots\ x_M\} \subset \mathbb{R}^N$$

$$\{y_1\ y_2\ y_3\ \ldots\ y_M\} \subset \mathcal{L}$$

$$f: \mathbb{R}^N \longrightarrow \mathcal{L} \qquad \mathcal{L} = \{-1, 1\}$$

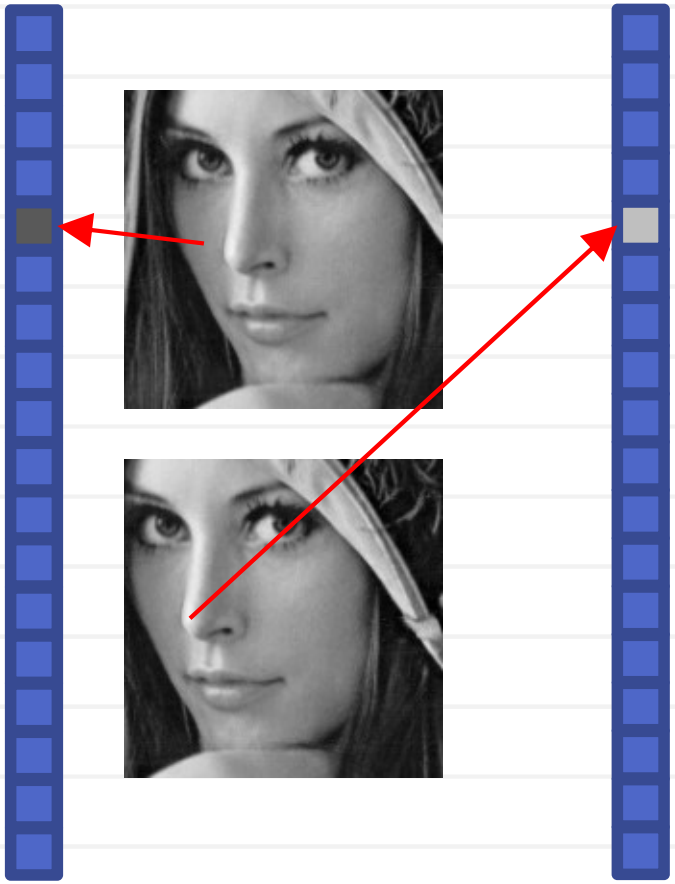
Goal: "recover" *f*.

E.g. linear classifier: $f(x) = \mathrm{sgn}\ w^T x$

Example:

vs.

Face detection challenge

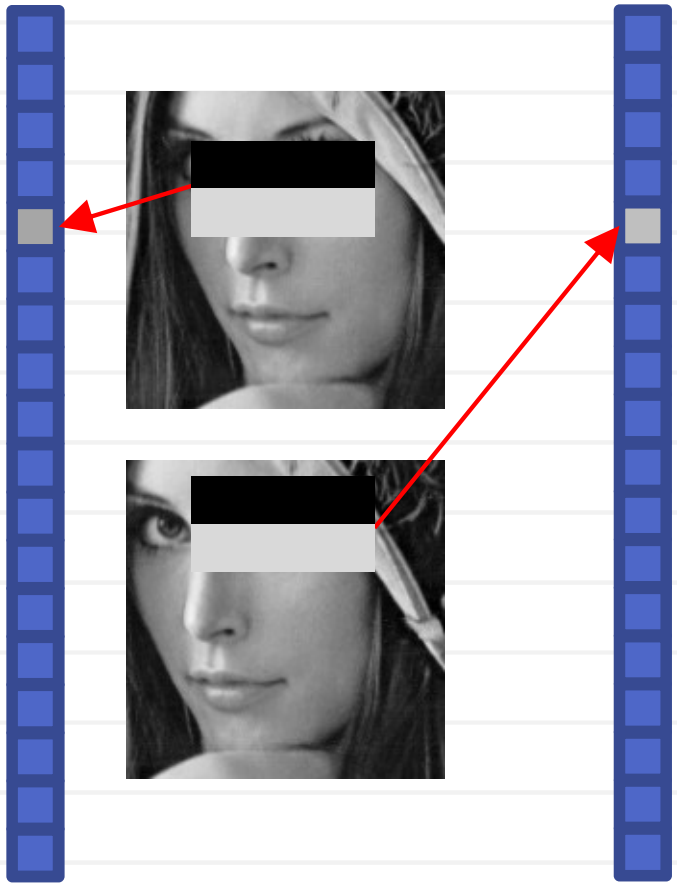


f(x1) f(x2)

Natural feature mapping:

- Highly non-smooth w.r.t. jitter
- Require lots of training samples

Haar features



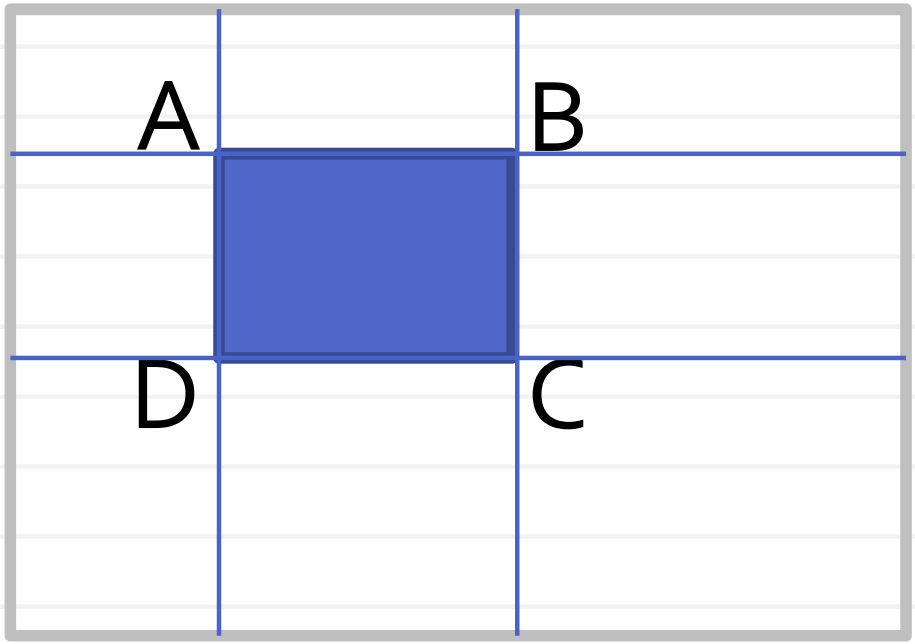
Viola-Jones features:

- Smoother w.r.t. jitter
- Less training examples needed
- *(also fast to compute)*

[Viola Jones, CVPR'01]

Haar features

$$F(A) = \int_{\lesssim A} \int f(x,y)\,dx\,dy$$

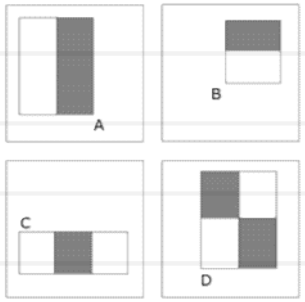


$$\int_{ABCD} \int f(x,y)\,dx\,dy =$$

$$= F(C) + F(A) - F(B) - F(D)$$

[Viola Jones, CVPR'01]

Viola-Jones detector



Haar feature extractor + thresholding

Linear classifier

"face"

"background"

- **Non-shallow**, learnable representation (AdaBoost greedy algorithm)
- Cascaded detector for speed
- Arguably, most impactful paper in CV history

[Viola Jones, CVPR'01]

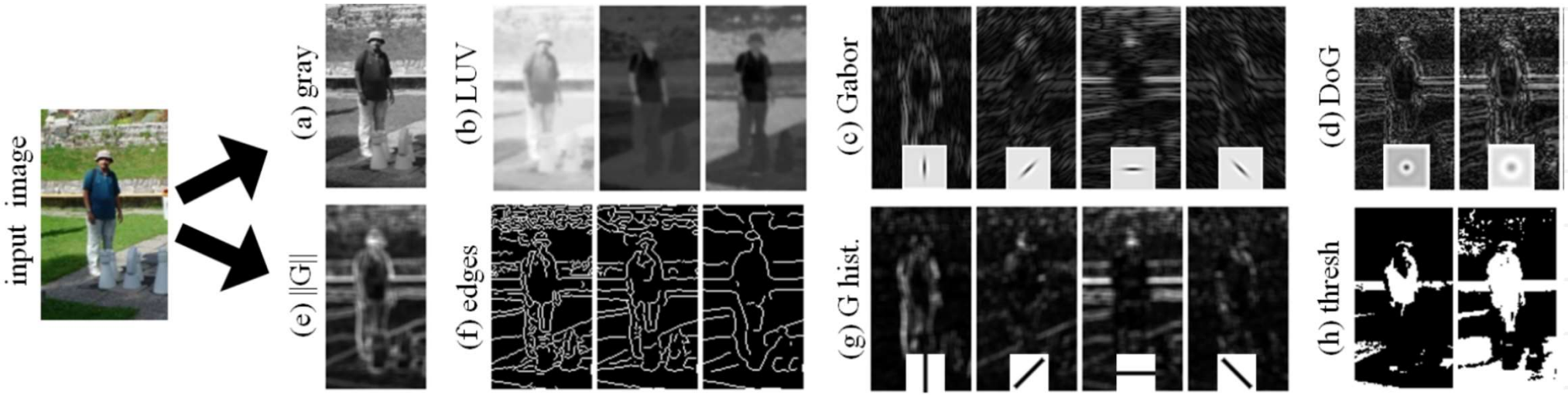
From face detection to pedestrian detection

vs.

Good industy-grade performance by Viola-Jones (for frontal faces)

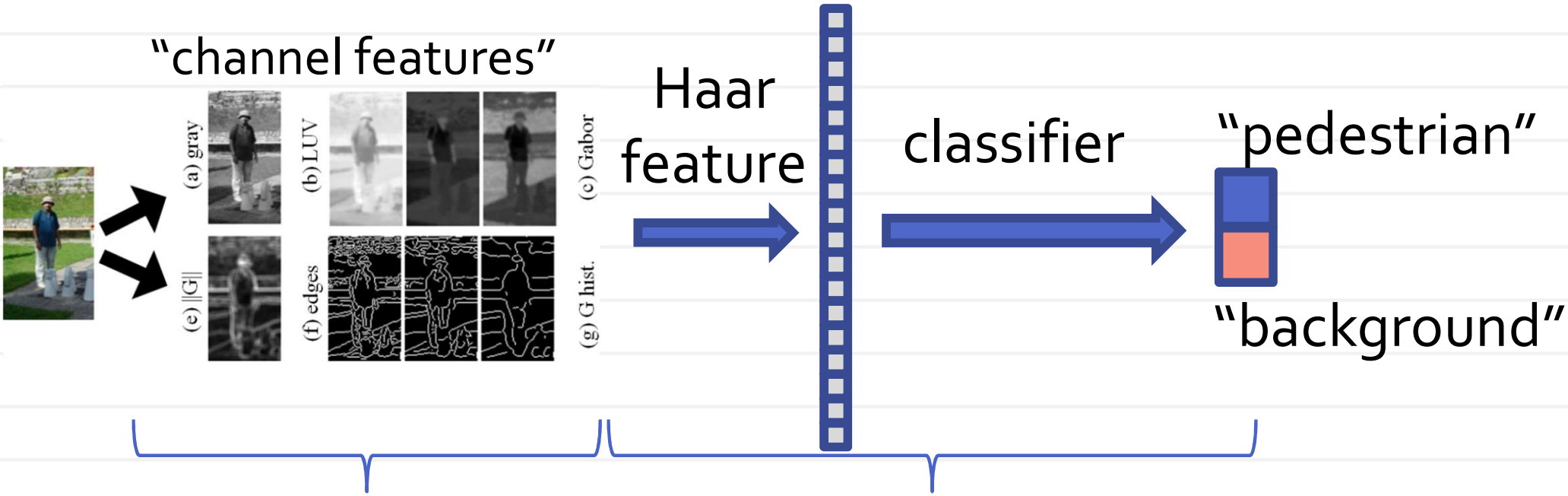
Viola-Jones detector not good enough

Improving pedestrian detection



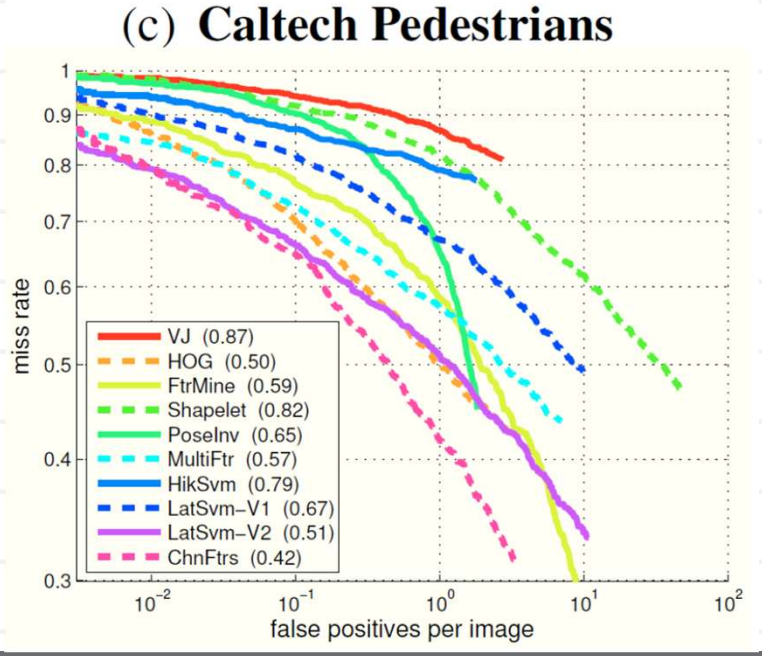
[Dollar et al. BMVC09]

Improved pedestrian detector



Hand-crafted

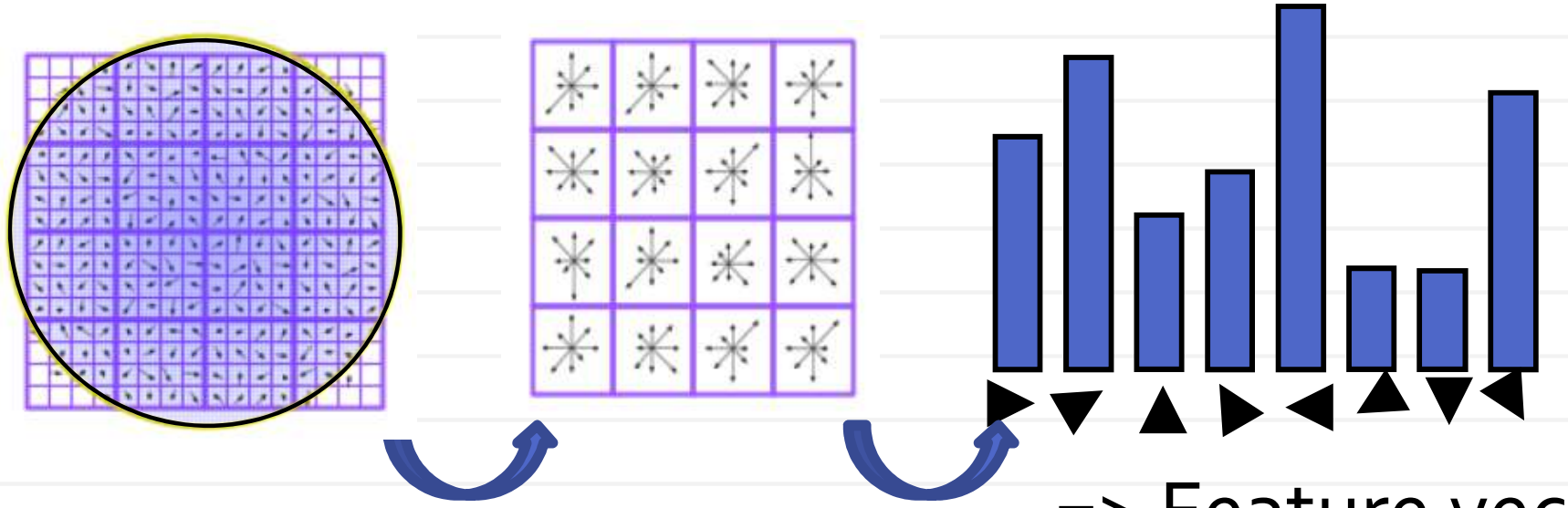
Trained using boosting



[Dollar, Tu, Perona, Belongie. *Integral Channel Features.* BMVC09]

SIFT descriptor

SIFT – "scale invariant feature transform"

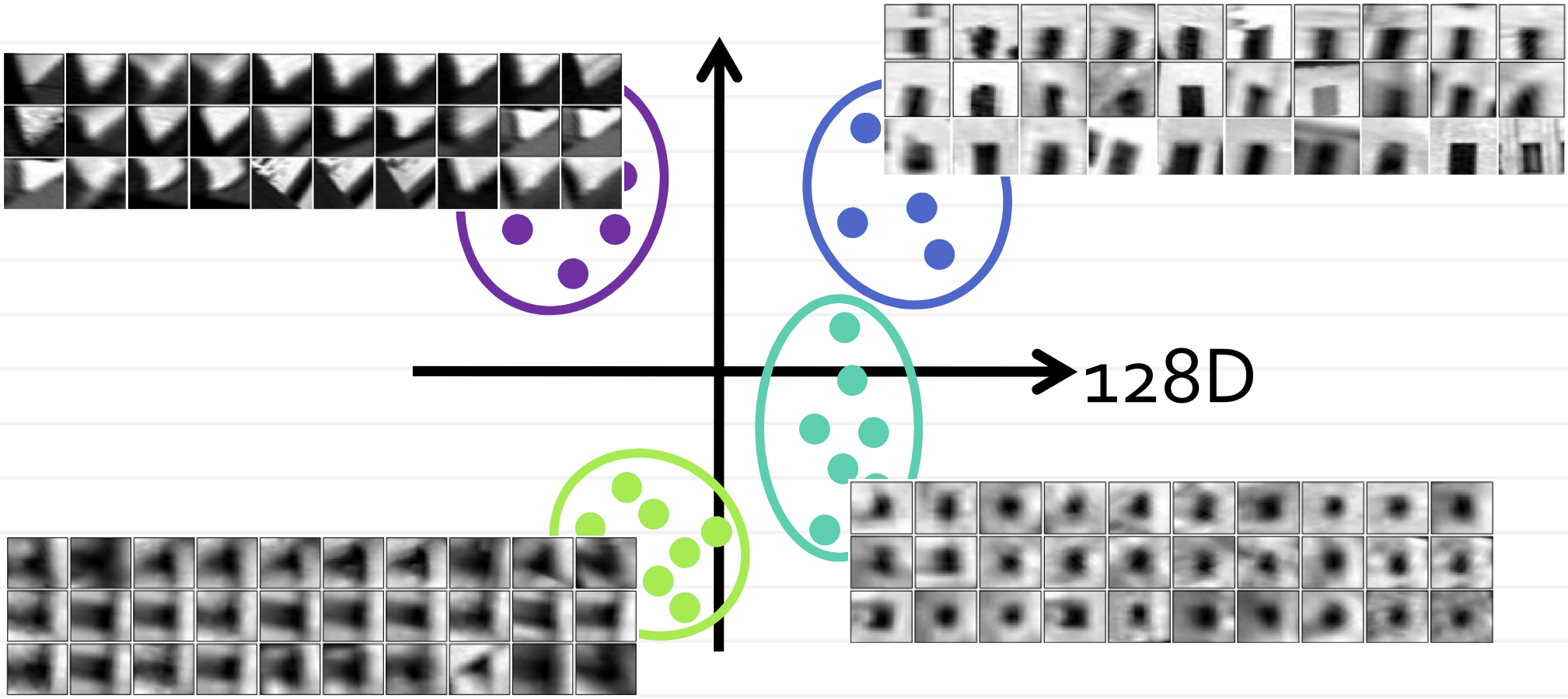


=> Feature vector (128D)

David G. Lowe: **Distinctive Image Features from Scale-Invariant Keypoints. IJCV'04**

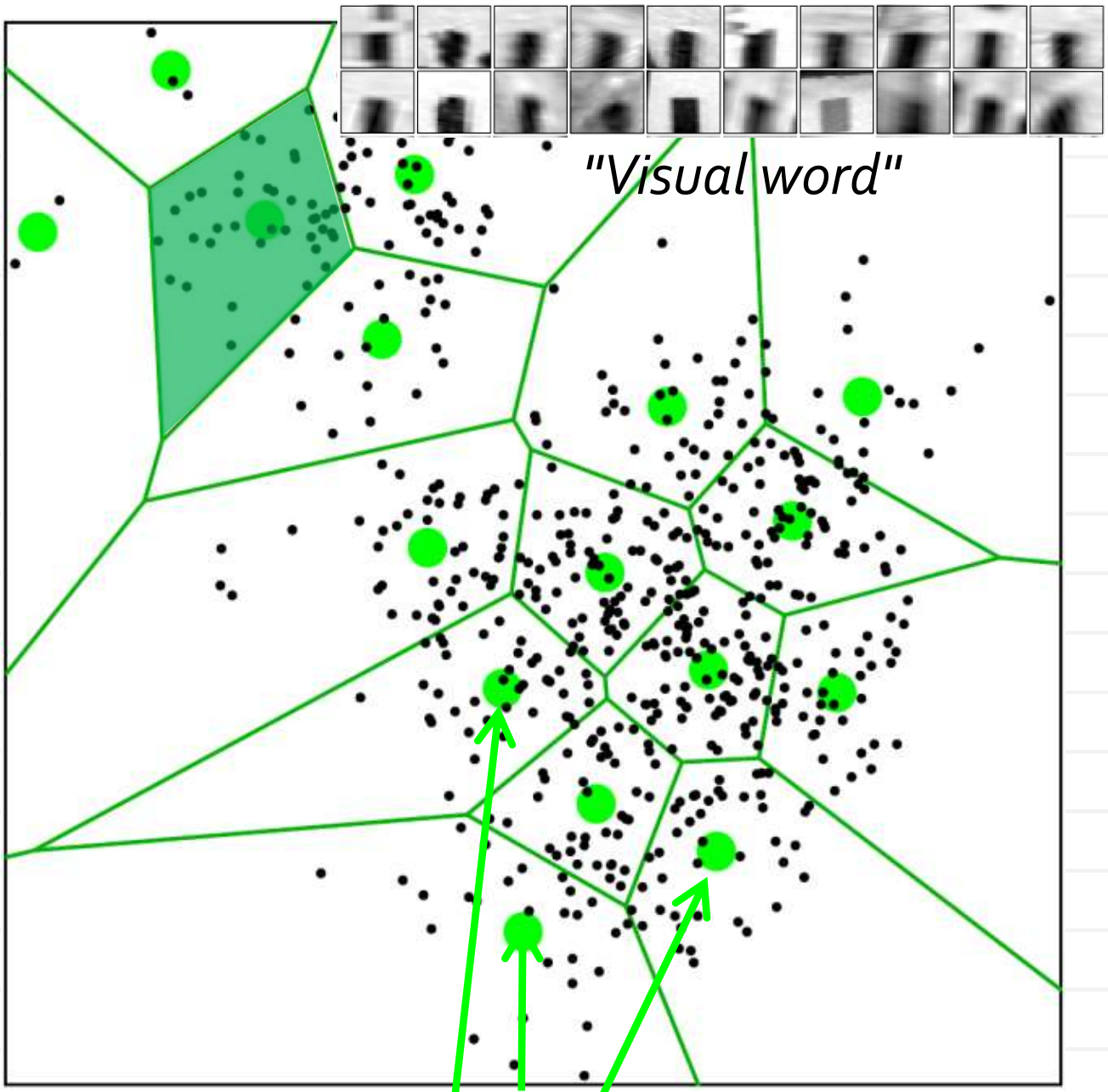
From words to *visual words*

VideoGoogle
Sivic & Zisserman, ICCV 2003



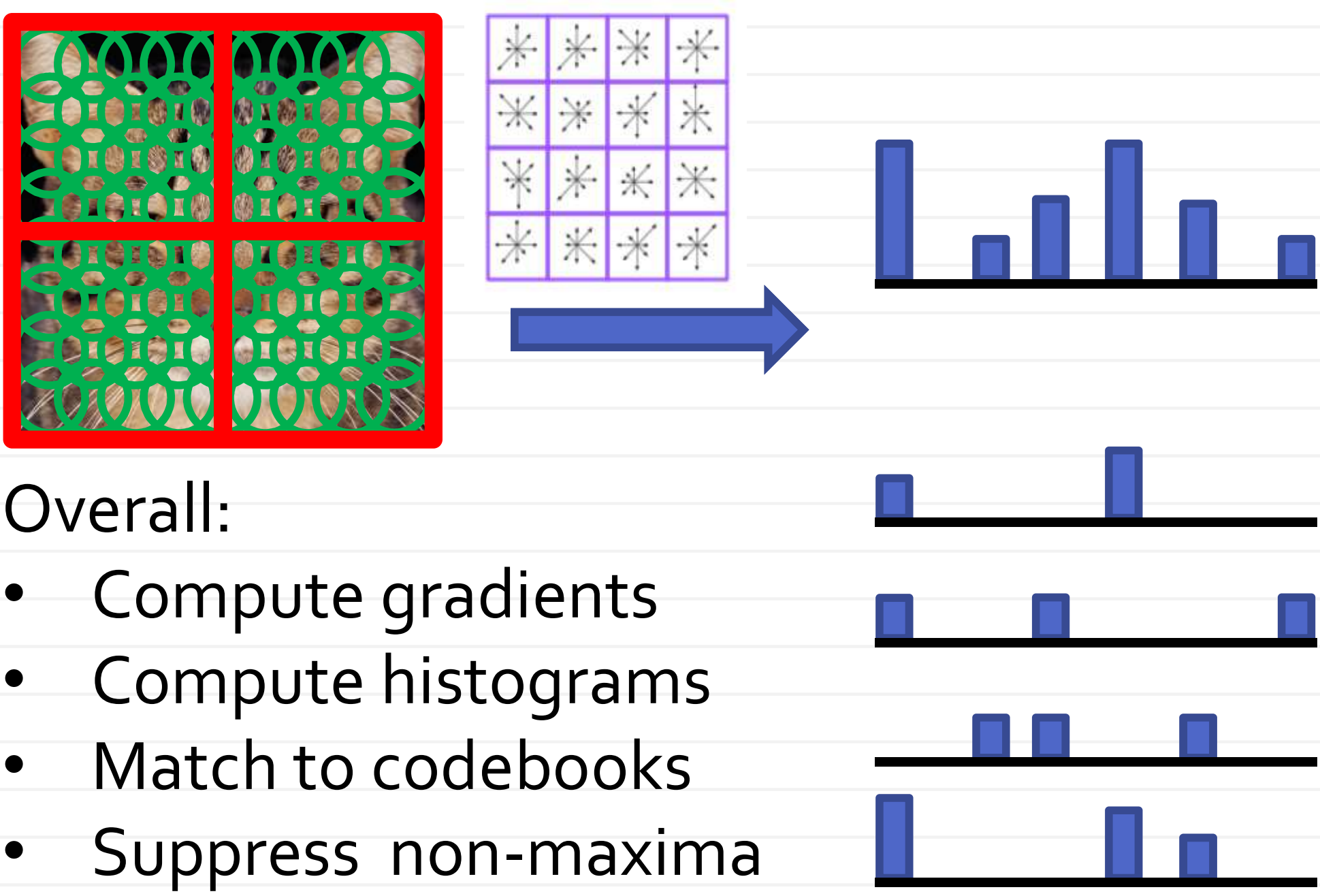
visual vocabulary/dictionary/codebook

Visual codebook



Sivic & Zisserman ICCV 2003

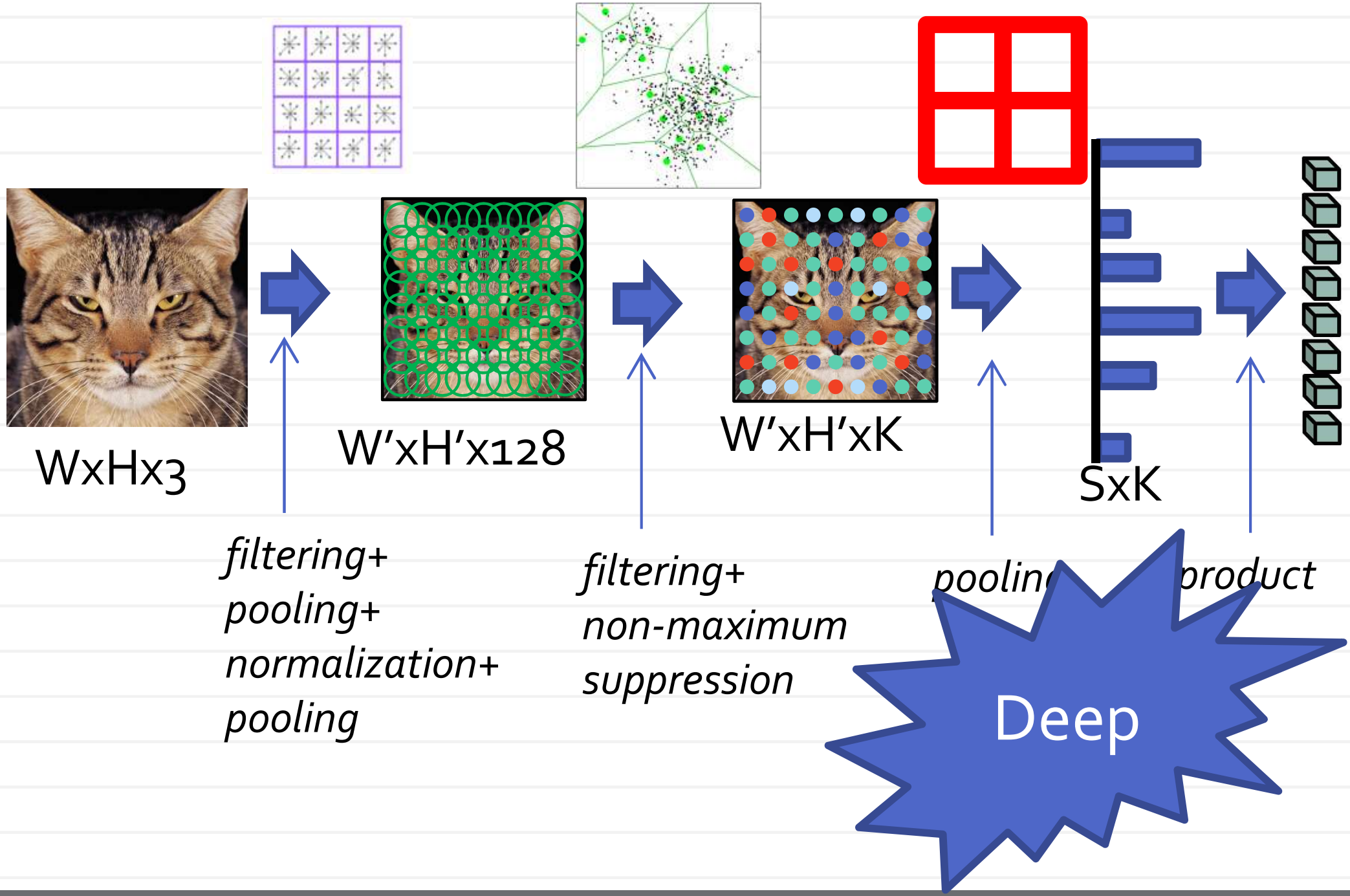
Bag-of-Words + Spatial pyramids



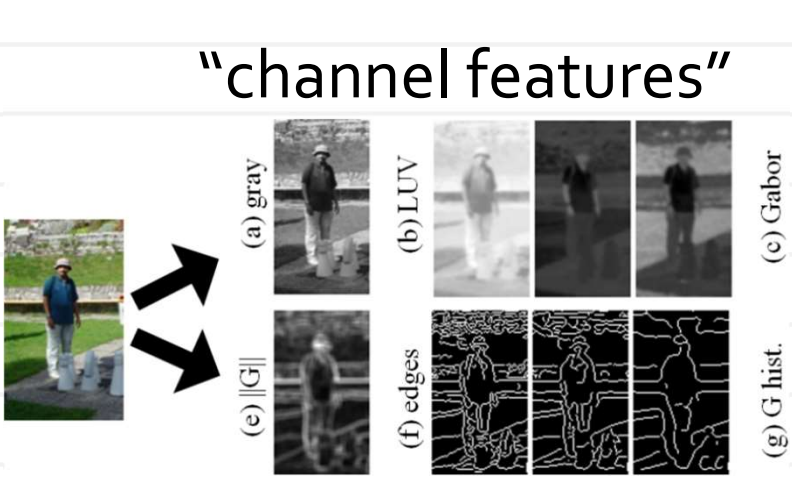
Overall:

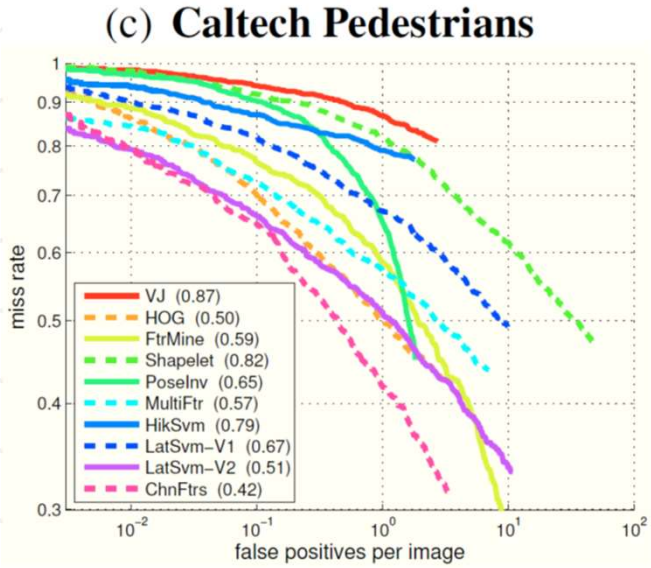
- Compute gradients
- Compute histograms
- Match to codebooks
- Suppress non-maxima

BoW image classifier: overview



Improved pedestrian detector



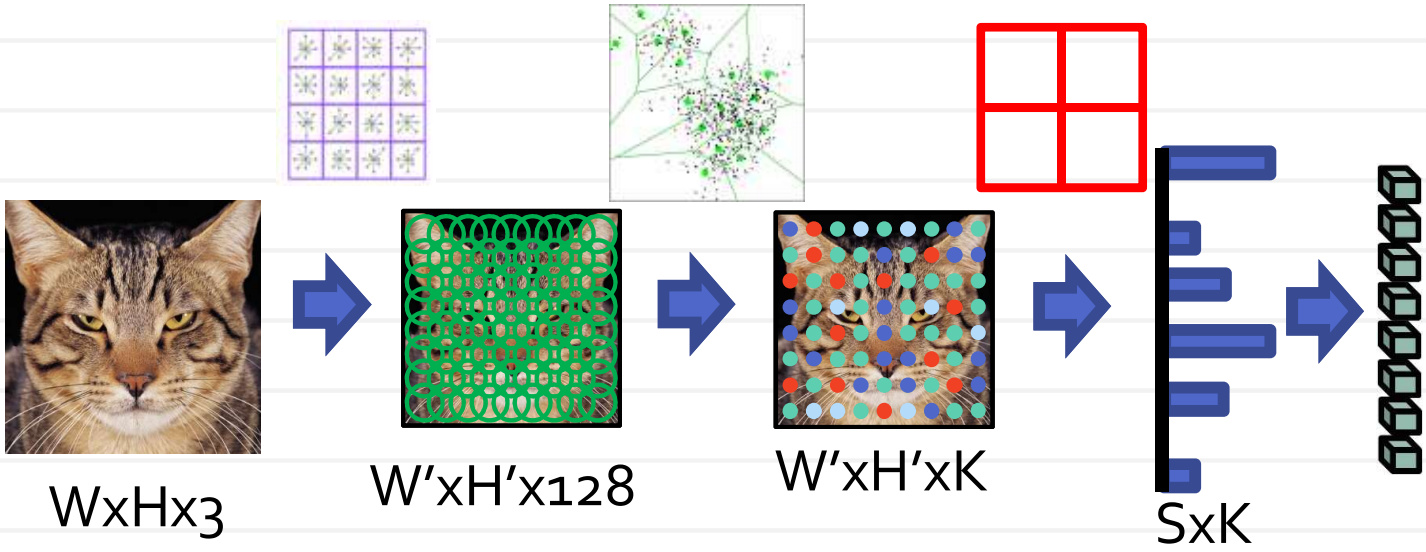


[Dollar, Tu, Perona, Belongie. *Integral Channel Features.* BMVC09]

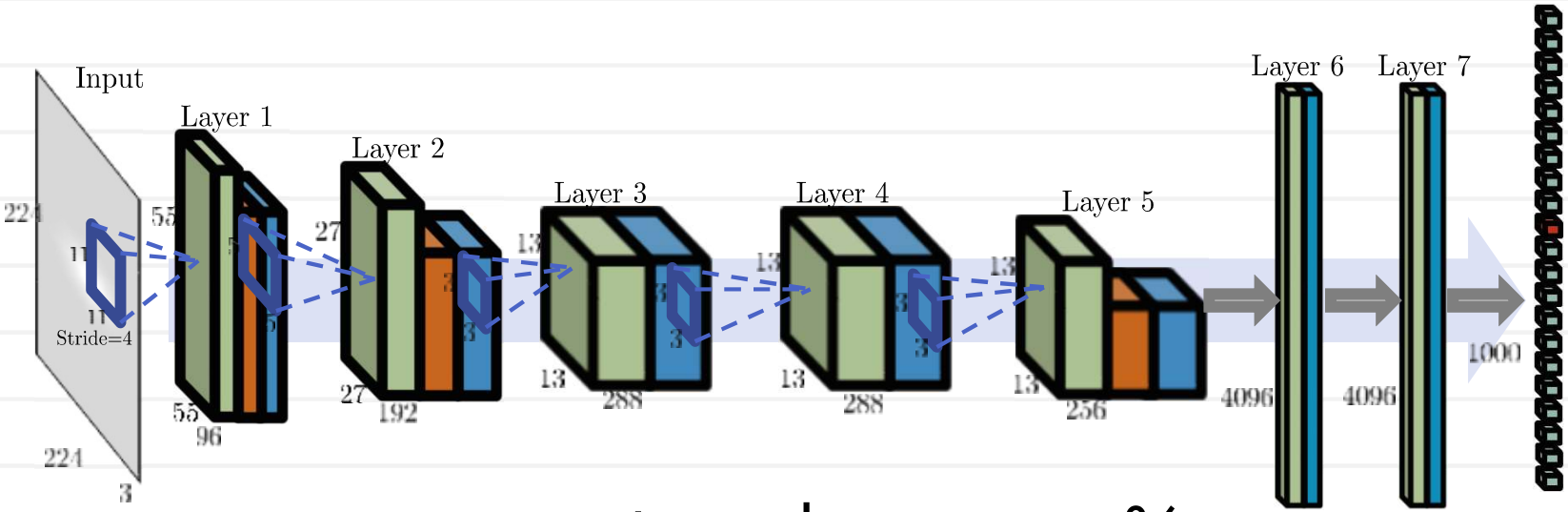
Then, what is "deep learning"?

- Previous CV systems were "deep", they used multiple layers of representation with success
- The main "novelty" in modern age *deep learning*: **end-to-end joint learning** of multiple (10+) layers

2006 vs 2014



2006: cat-vs-dog error 40%



2014: cat-vs-dog error 1%

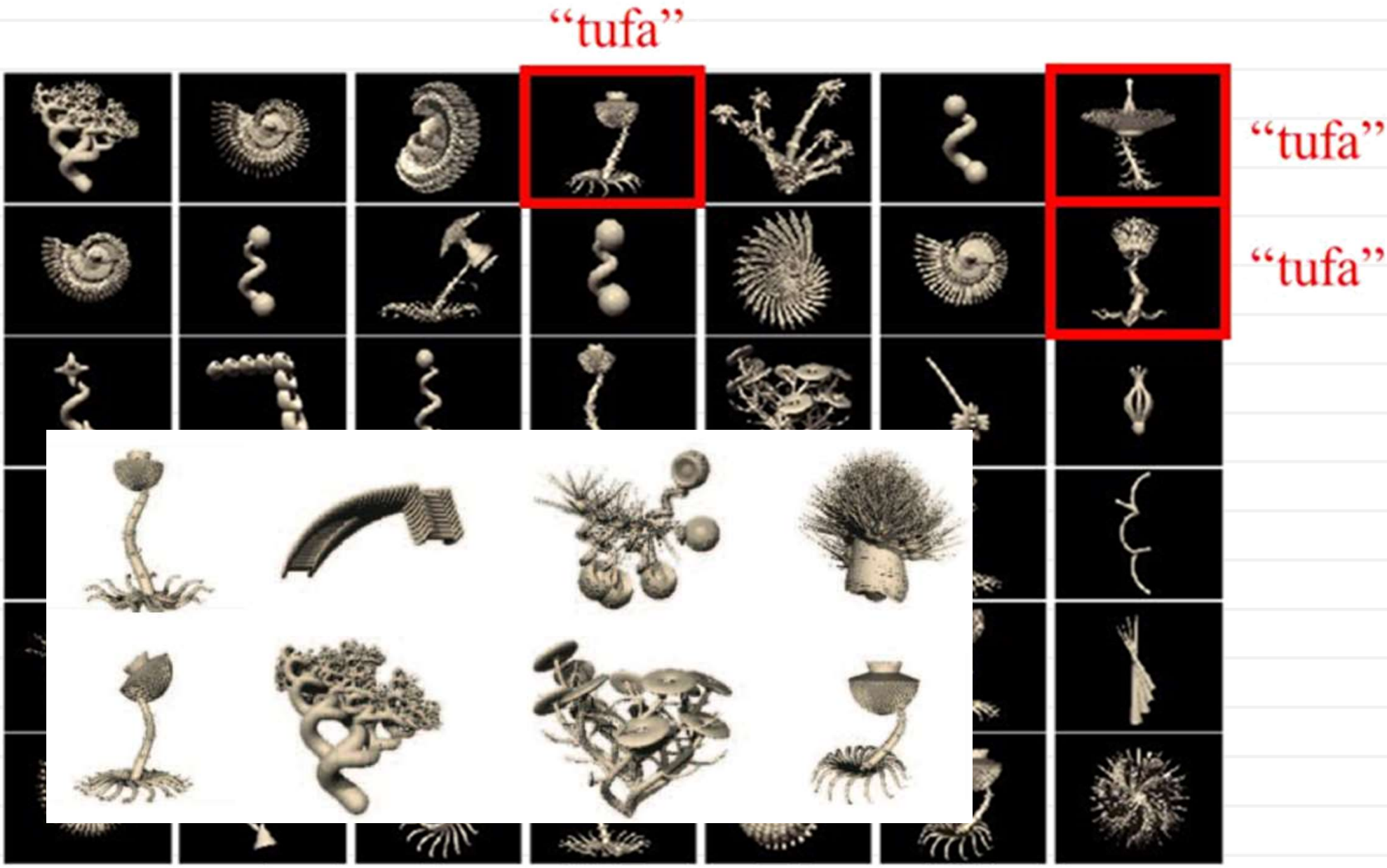
Then, what is "deep learning"?

End-to-end joint learning of all layers:

- multiple assemblable blocks
- each block is piecewise-differentiable
- gradient-based optimization
- gradients computed by backpropagation

Deep learning "revolution" (2012? – now): rapid engineering improvements following these principles

Gazoob world



[Tenenbaum et al. Science 2011]

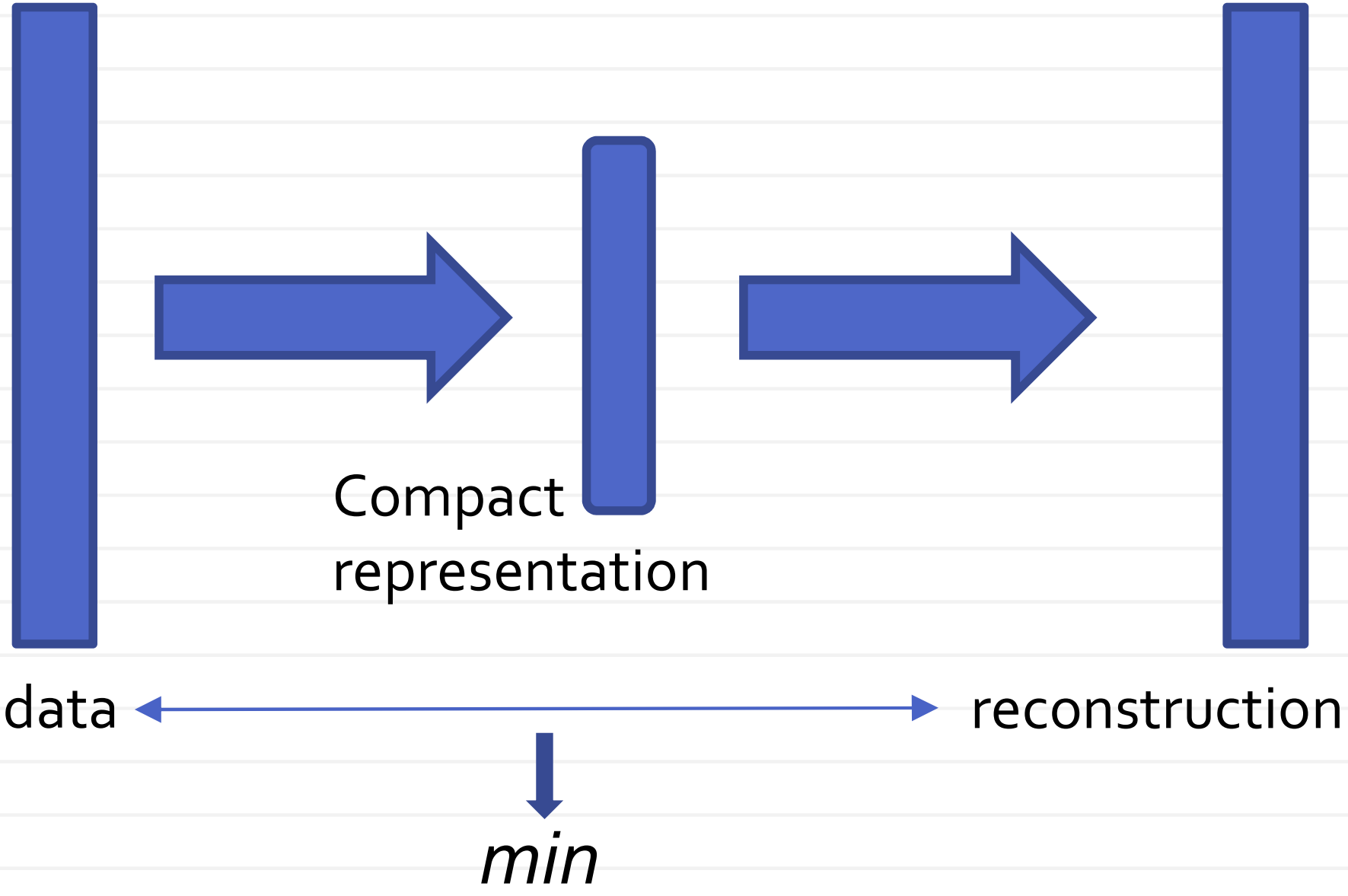
Bringing in extra knowledge

- *Data augmentation*
- Pre-learning bottom layers using unlabeled data (unsupervised learning)
- Transferring bottom layers from related supervised tasks
- *Domain adaptation:* same task, shifted data
- *Multi-task learning:* related task, similar data

Need intermediate level representations!

Learning representation from unlabeled data

Most popular approach:



Principal component analysis

$$\{x_1\ x_2\ x_3\ \ldots\ x_M\} \in \mathbb{R}^N$$

Idea: represent each vector as a linear combination of few *components*.

Then, all we have to store are a few (<<M) components from R^N, and a small number of coefficients (<<N) per each vector.

Thus, we will use much less then O(MN) memory (our representation is "compact").

Picking the components

We start by picking the mean $\bar{x}$ and subtracting it from all vectors.

$$\{x_1\ x_2\ x_3\ \ldots\ x_M\} \in \mathbb{R}^N$$

From now on assume $\sum_{i=1}^{M} x_i = \vec{0}$

We now want to pick the *D* optimal components $\{v_1, \ldots v_D\} \subset \mathbb{R}^N$

We will assume that *v* have unit lengths.

Reconstruction error

We now want to pick the *d* optimal components $\{v_1, \dots v_D\} \subset \mathbb{R}^N$

Let $\tilde{X} = [\tilde{x}_1, \tilde{x}_2 \dots \tilde{x}_M]$

be the reconstruction of the original vectors.

We want to minimize the reconstruction error:

$$\sum_{i=1}^{M} \| x_i - \tilde{x}_i \|_2^2 \to \min$$

$$\| X - \tilde{X} \|_F^2 \to \min$$

Finding the best components

$\tilde{X}$ is defined by the fact that it has rank *D*.

$$\min_{\tilde{X}} \| X - \tilde{X} \|_F^2$$
$$s.t.: \quad rk\, \tilde{X} = D$$

SVD: $X = U S V^T$

$[x_1 \dots x_M]$

orthogonal(NxN)

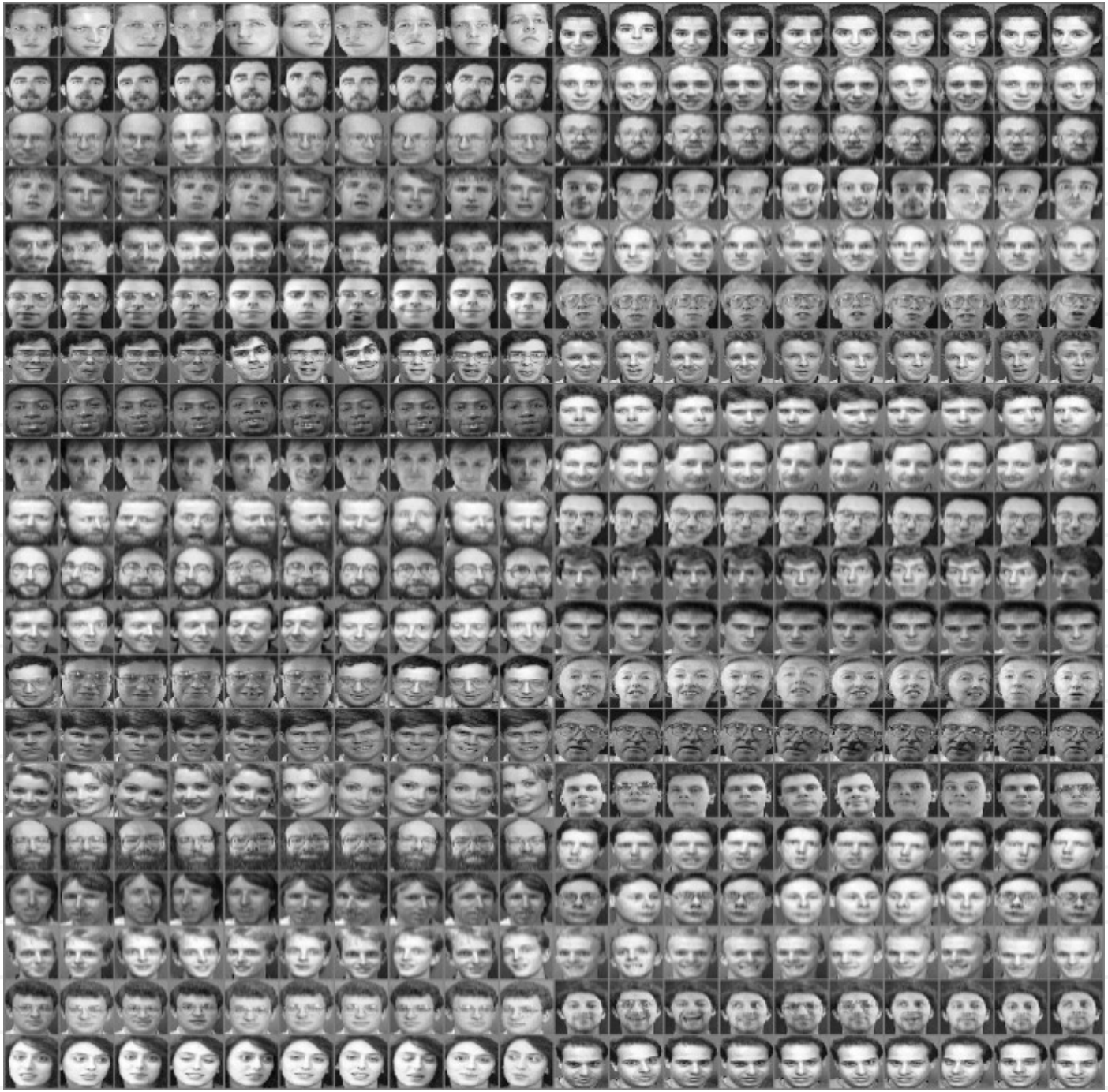
diagonal (NxM)

orthogonal (MxM)

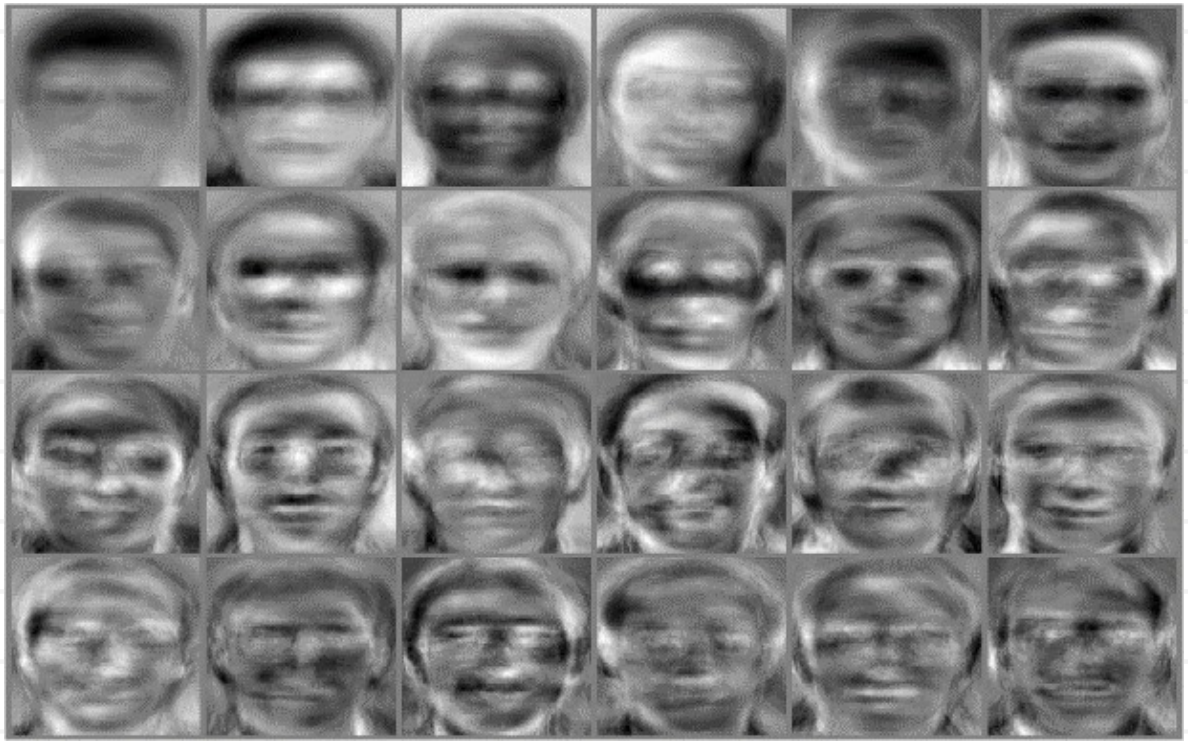
nullifying all but D largest diagonal values

The solution: $\tilde{X} = U \tilde{S} V^T$

Eigenfaces

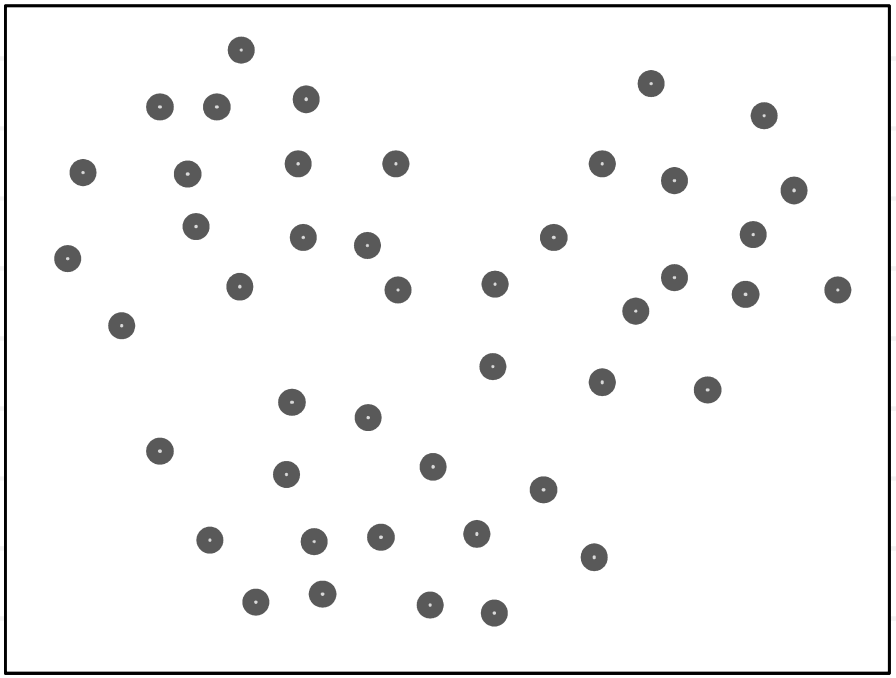


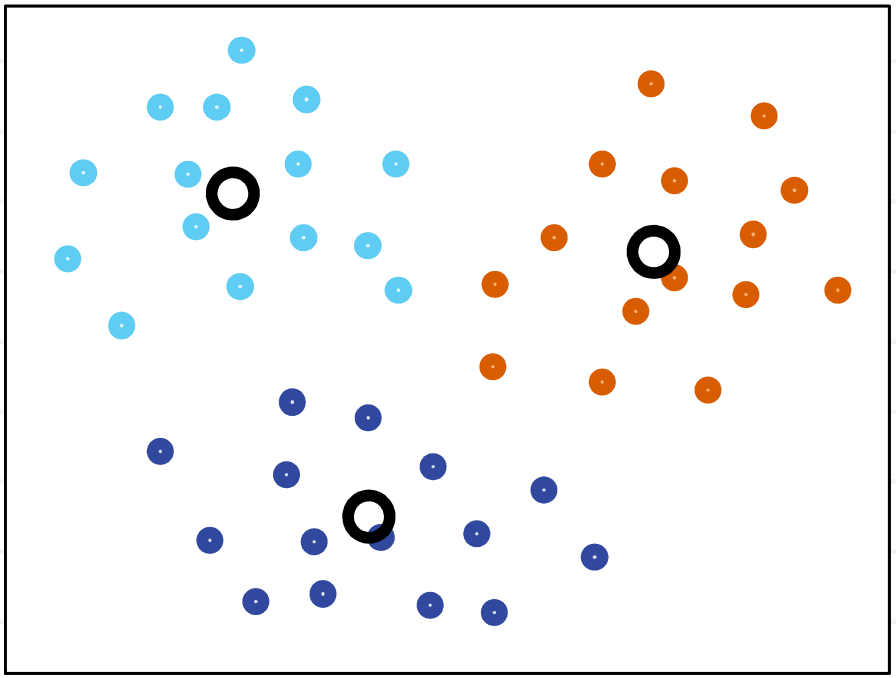
Eigenfaces



Eigenfaces = principal components of the dataset

k-means clustering





Task: split the points $\{x\}$ into k clusters

Input points:

$$x_1, x_2 \dots x_M \in \mathbb{R}^N$$

$$\min_{c,n} \sum_{i=1}^{M} \|x_i - c_{n_i}\|_2^2$$

Cluster centers:

$$c_1, c_2 \dots c_K \in \mathbb{R}^N$$

Point assignments:

$$n_i \in \{1, 2, \dots K\}$$

Solving the k-means problem

$$\min_{c,n} \sum_{i=1}^{M} \|x_i - c_{n_i}\|_2^2$$ - "hard" problem

$$\min_{n} \sum_{i=1}^{M} \|x_i - c_{n_i}\|_2^2$$

Exact solution:

$$n_i = \arg\min_t \|x_i - c_t\|_2^2$$

(for each point pick the closest center)

$$\min_{c} \sum_{i=1}^{M} \|x_i - c_{n_i}\|_2^2$$

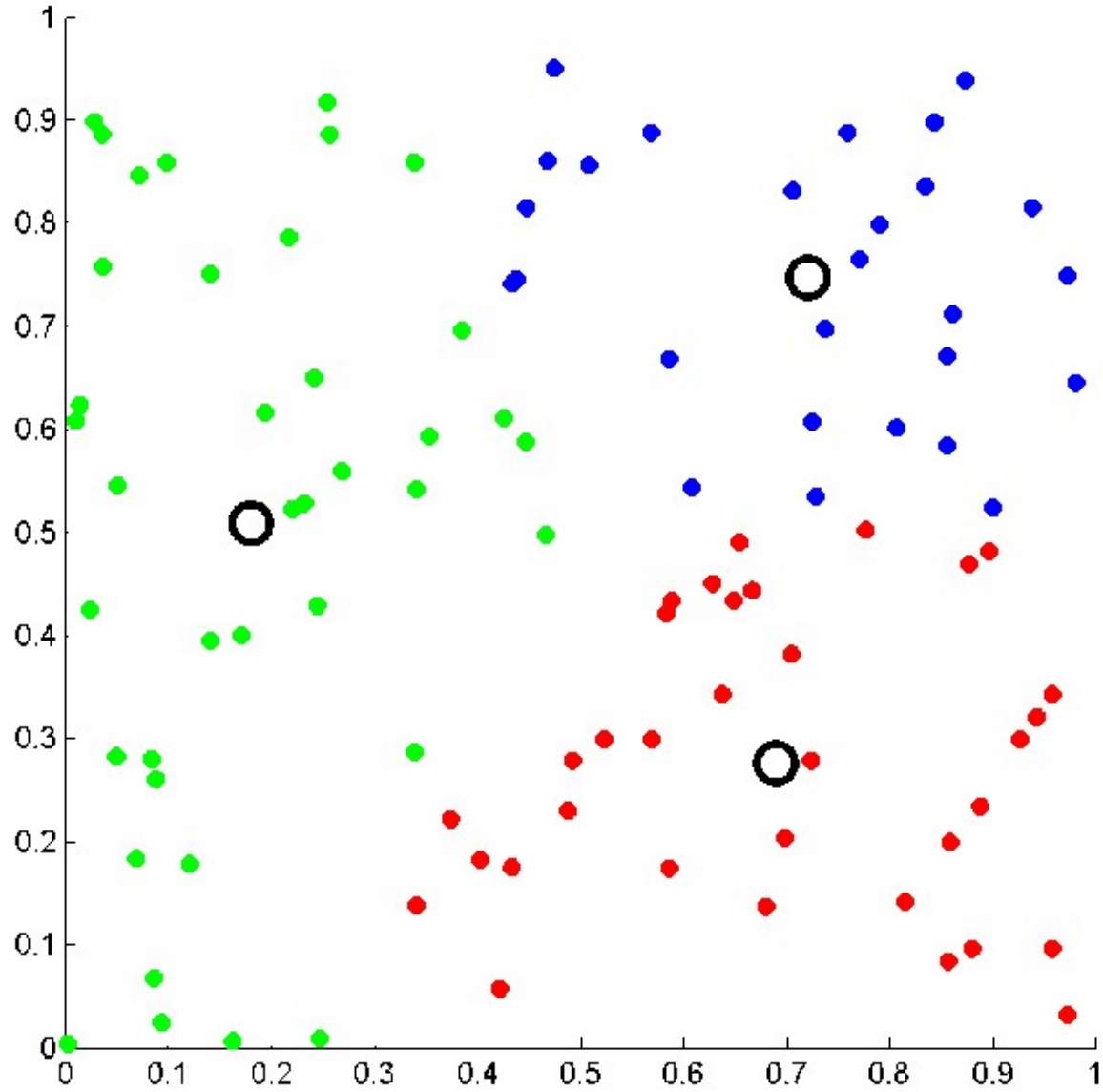
Exact solution:

$$N_t = \{ i \mid n_i = t \}$$

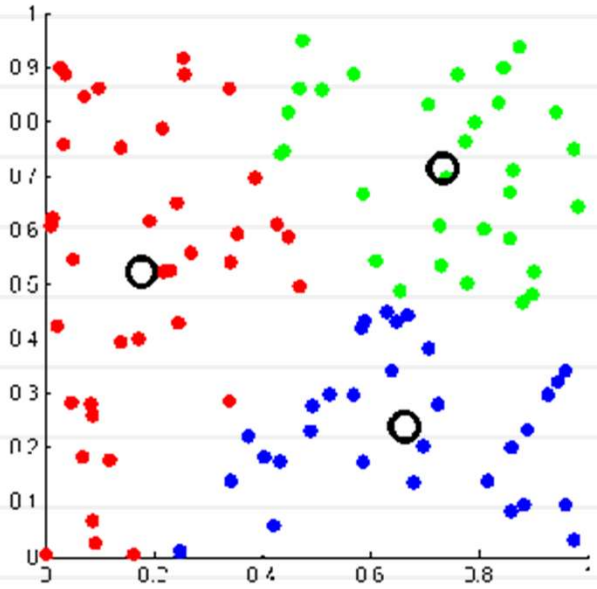
$$c_t = \frac{1}{|N_t|} \sum_{i \in N_t} x_i$$

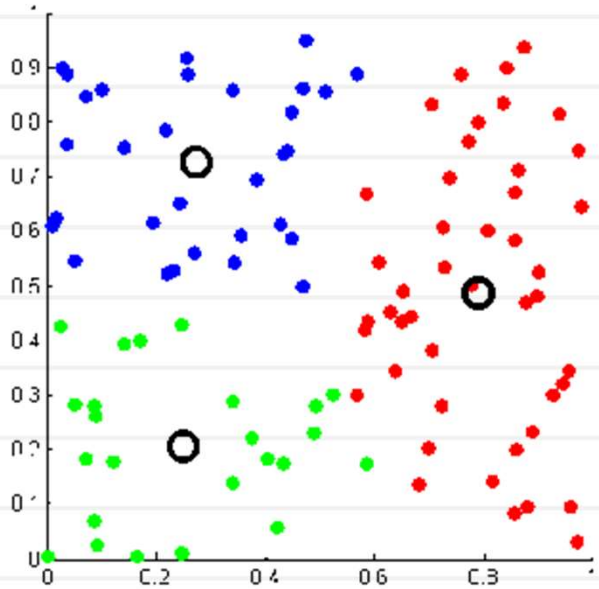
(average points that belong to each cluster)

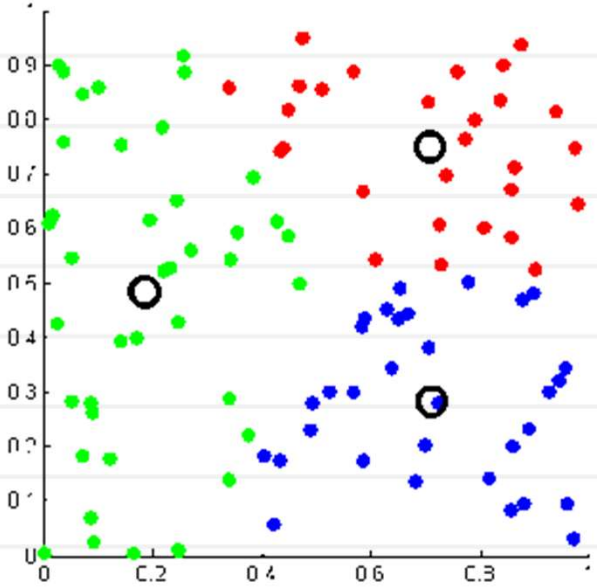
K-means example



k-means clustering







- Same input points
- Three different local minima
- Multiple restarts would help

How does it fit into our picture?

$$x_1, x_2 \dots x_M \in \mathbb{R}^N$$

Cluster centers:

$$c_1, c_2 \dots c_K \in \mathbb{R}^N$$

Point assignments:

$$n_i \in \{1, 2, \dots K\}$$

"Small"representation!

Optimizing the reconstruction error:

$$\min_{c,n} \sum_{i=1}^{M} \|x_i - c_{n_i}\|_2^2$$

Sparse coding: encoding scheme

$$x_1, x_2 \dots x_M \in \mathbb{R}^N$$

Dictionary (bases):

$$d_1 d_2 \dots d_K \in \mathbb{R}^N$$

Coefficients (weights):

$$\alpha_1 \dots \alpha_M \in \mathbb{R}^K$$

sparse (few non zeros)

$$x_i \approx \sum_k \alpha_i^k d_k = D \alpha_i$$

Sparse coding: encoding scheme

Dictionary (bases):

$$d_1 d_2 \dots d_K \in \mathbb{R}^N$$

sparse (few non zeros)

$$x \approx \sum_k \alpha^k d_k = D\alpha$$

How to find such α?

1. Greedy algorithm (orthogonal matching pursuit)
2. L1-minimization

Example of sparse coding

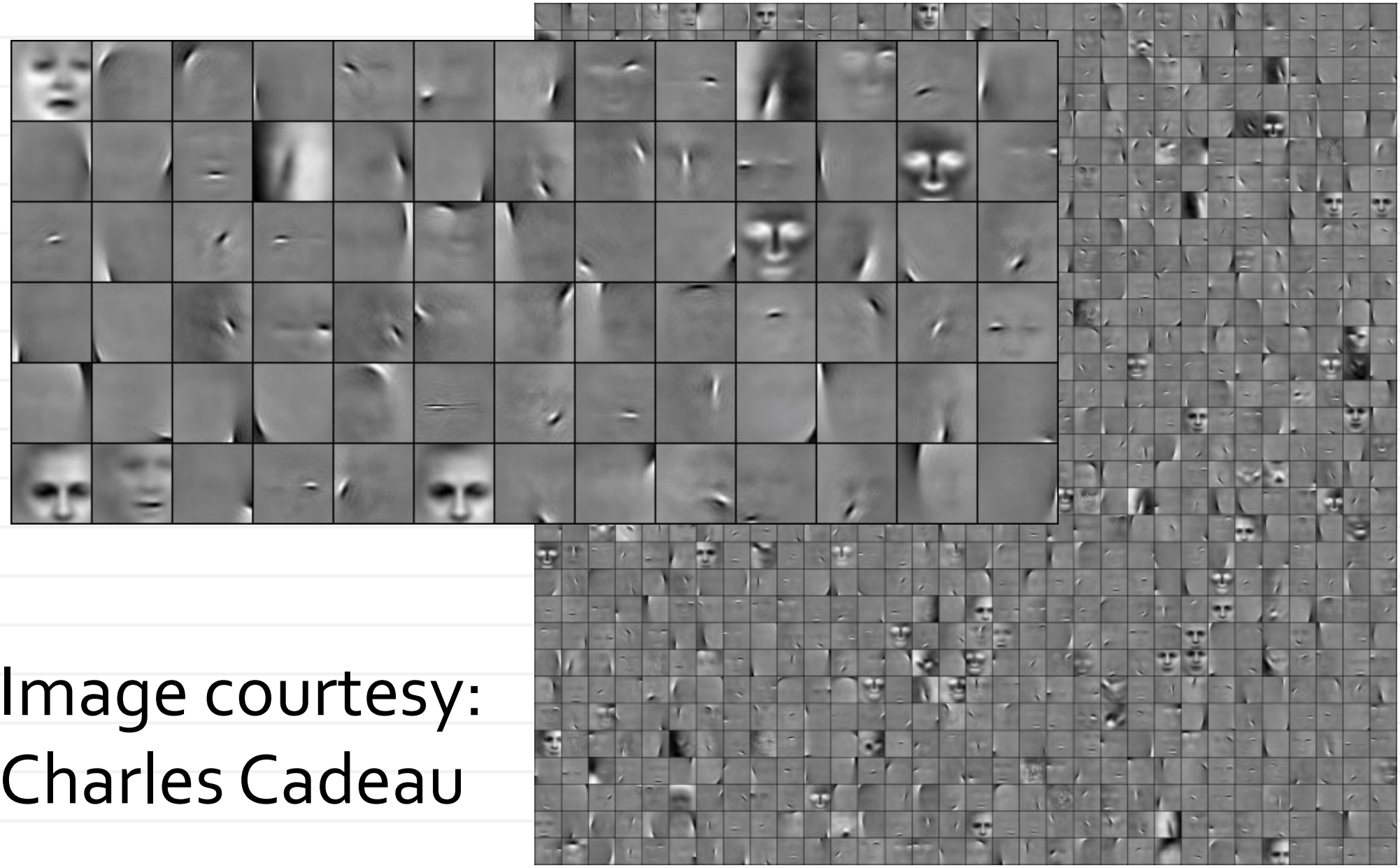
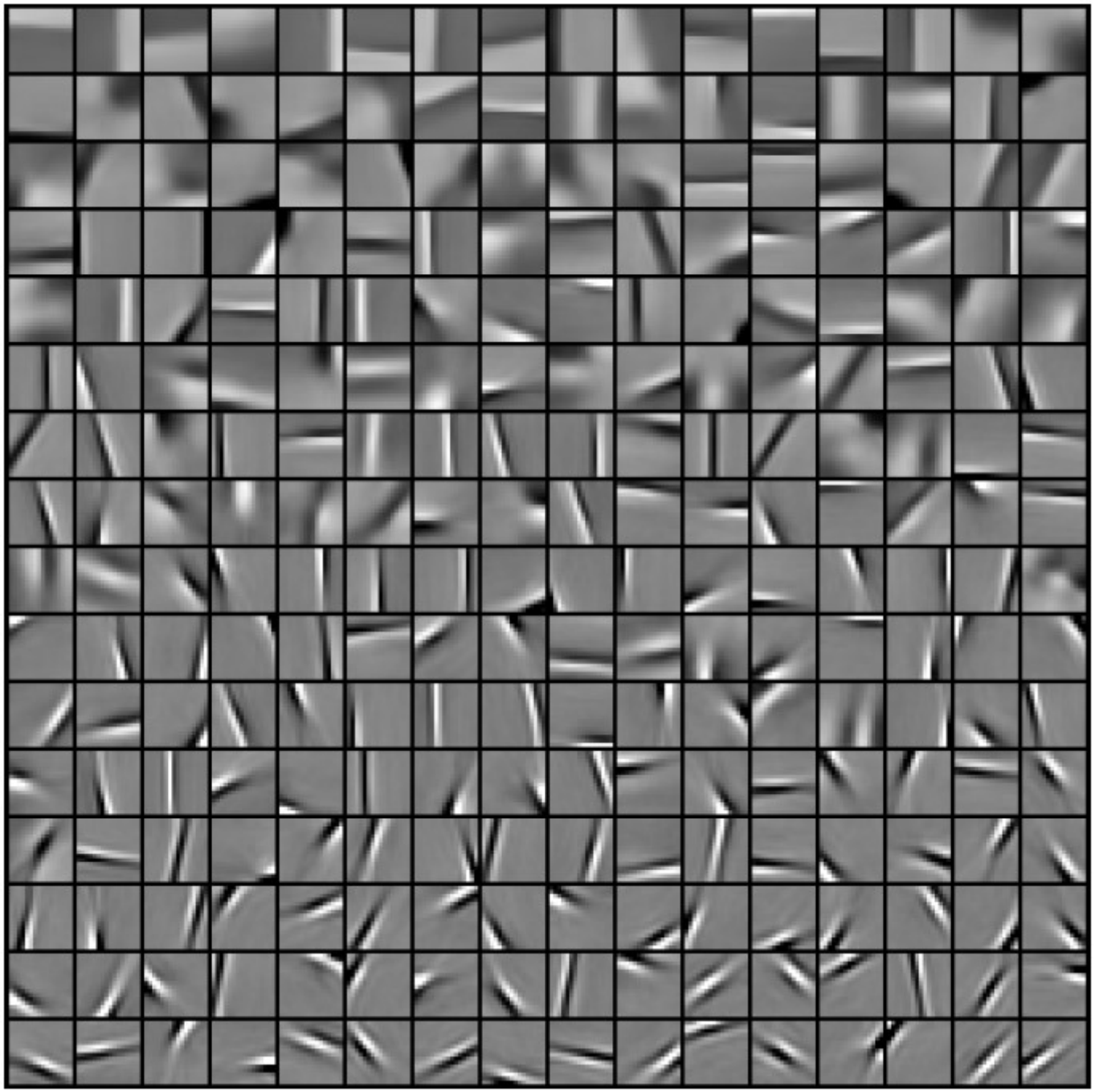


Image courtesy:
Charles Cadeau

Sparse coding of image patches



Huber and Wiesel 1968

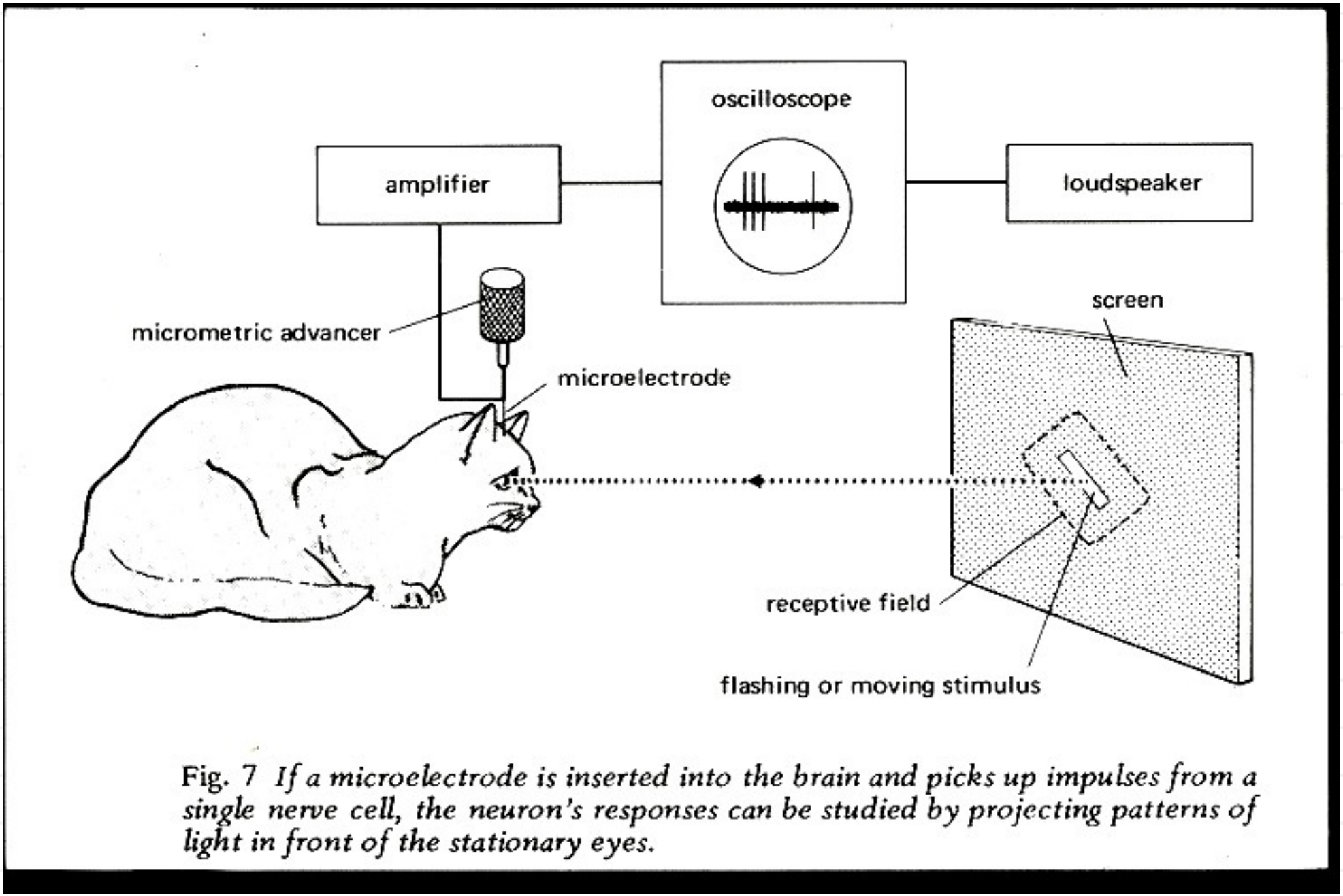
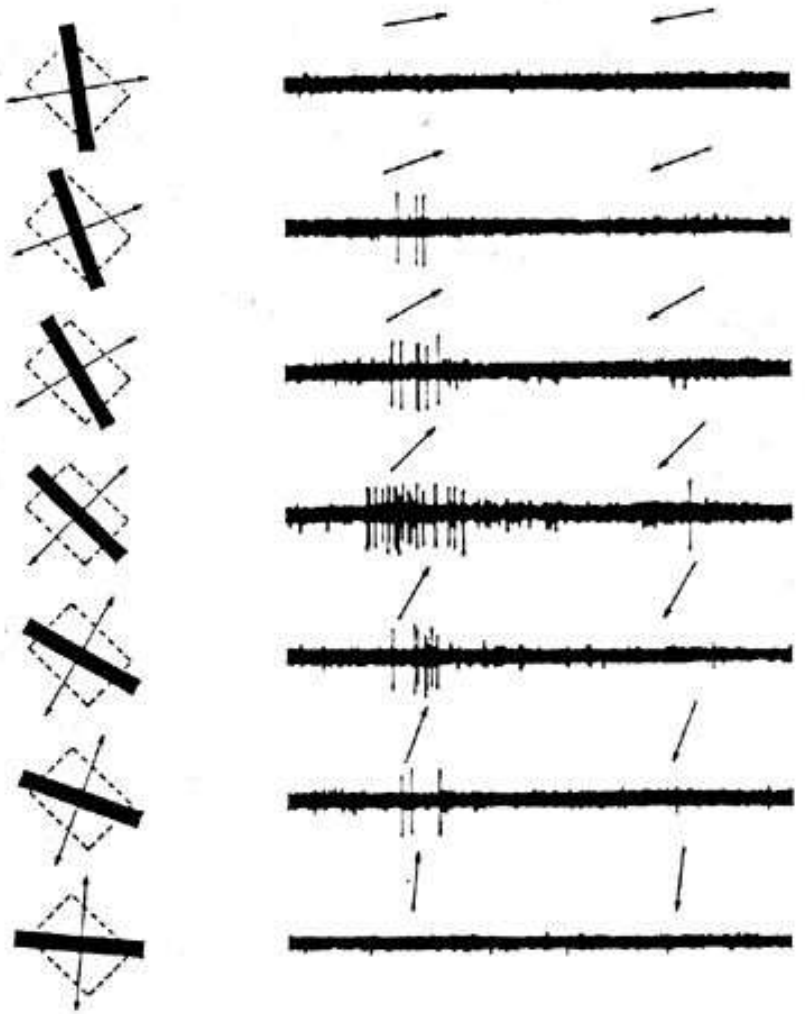


Fig. 7 *If a microelectrode is inserted into the brain and picks up impulses from a single nerve cell, the neuron's responses can be studied by projecting patterns of light in front of the stationary eyes.*

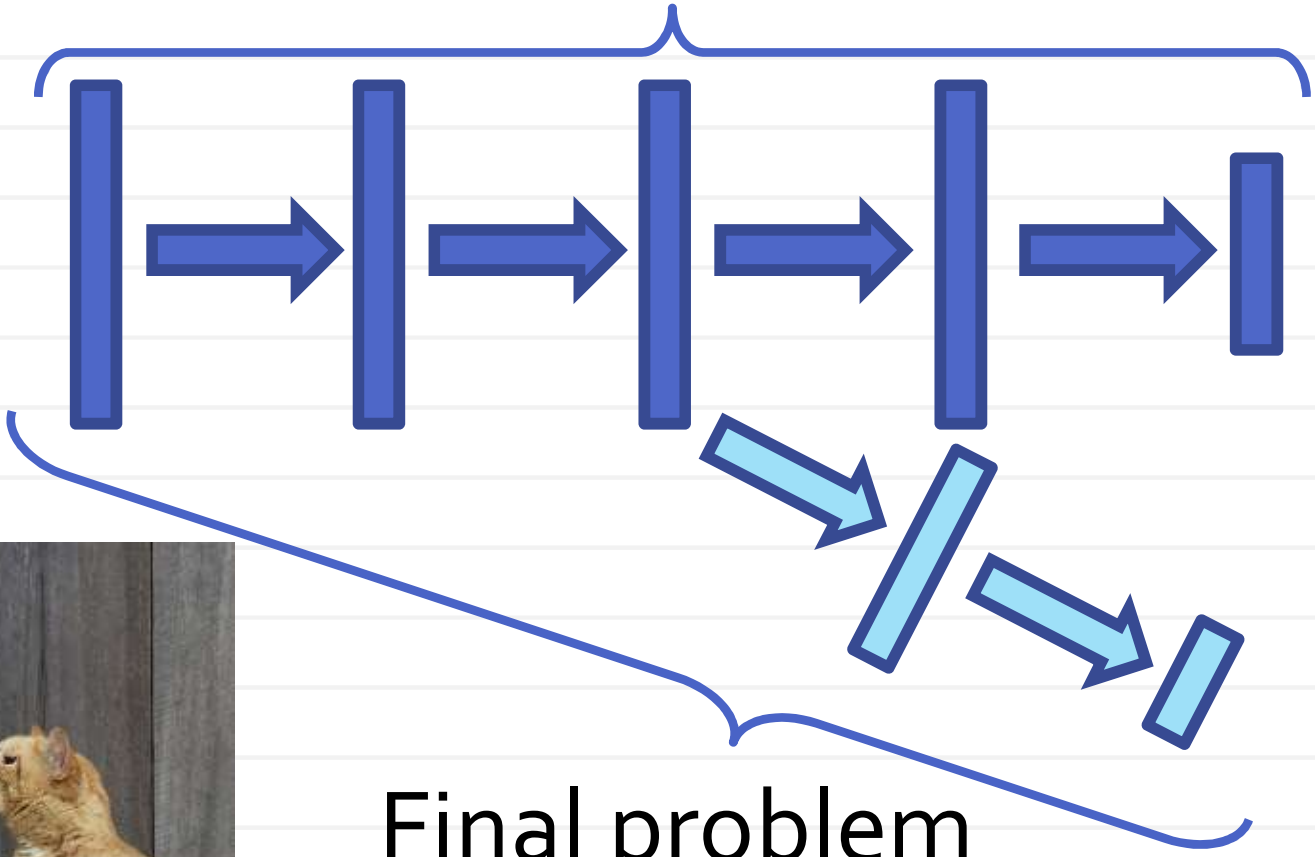
Huber and Wiesel 1968

V1 physiology: direction selectivity



Transfer learning

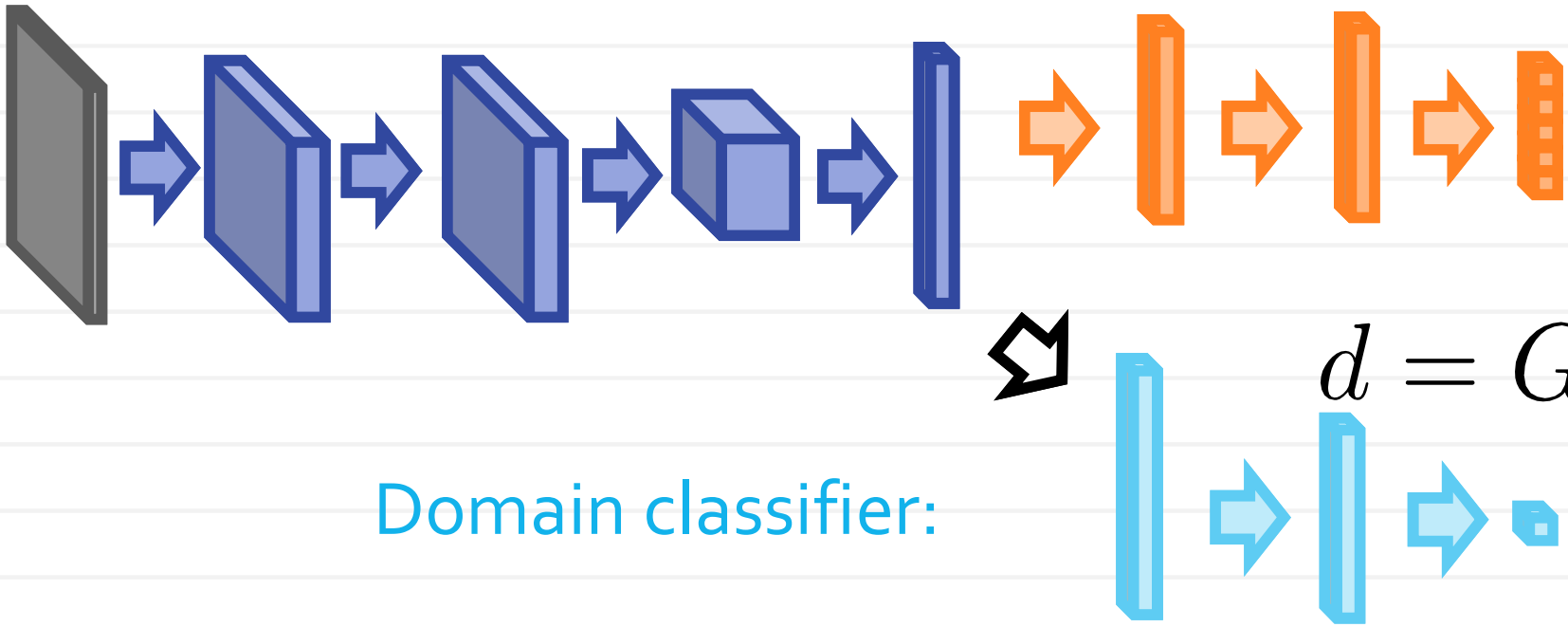
Task where we have a lot of data



Final problem

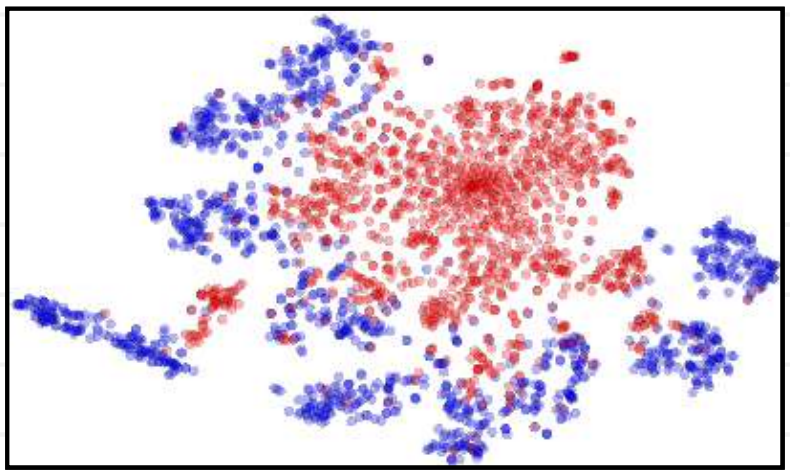
Domain adaptation setting

Idea: measuring domain shift

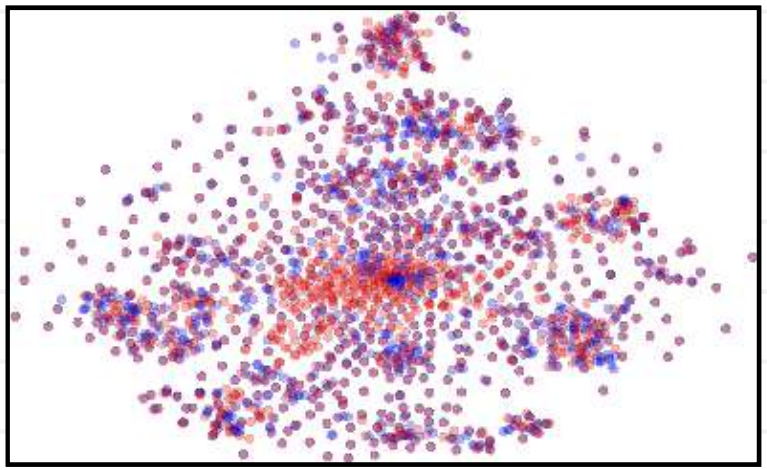


$$d = G_d(\mathbf{f}; \theta_d)$$

Domain classifier:



Domain loss low



Domain loss high

[Ganin et al. 2016]

Deep learning: recap

End-to-end joint learning of all layers:

- multiple assemblable blocks
- each block is piecewise-differentiable
- gradient-based optimization
- gradients computed by backpropagation

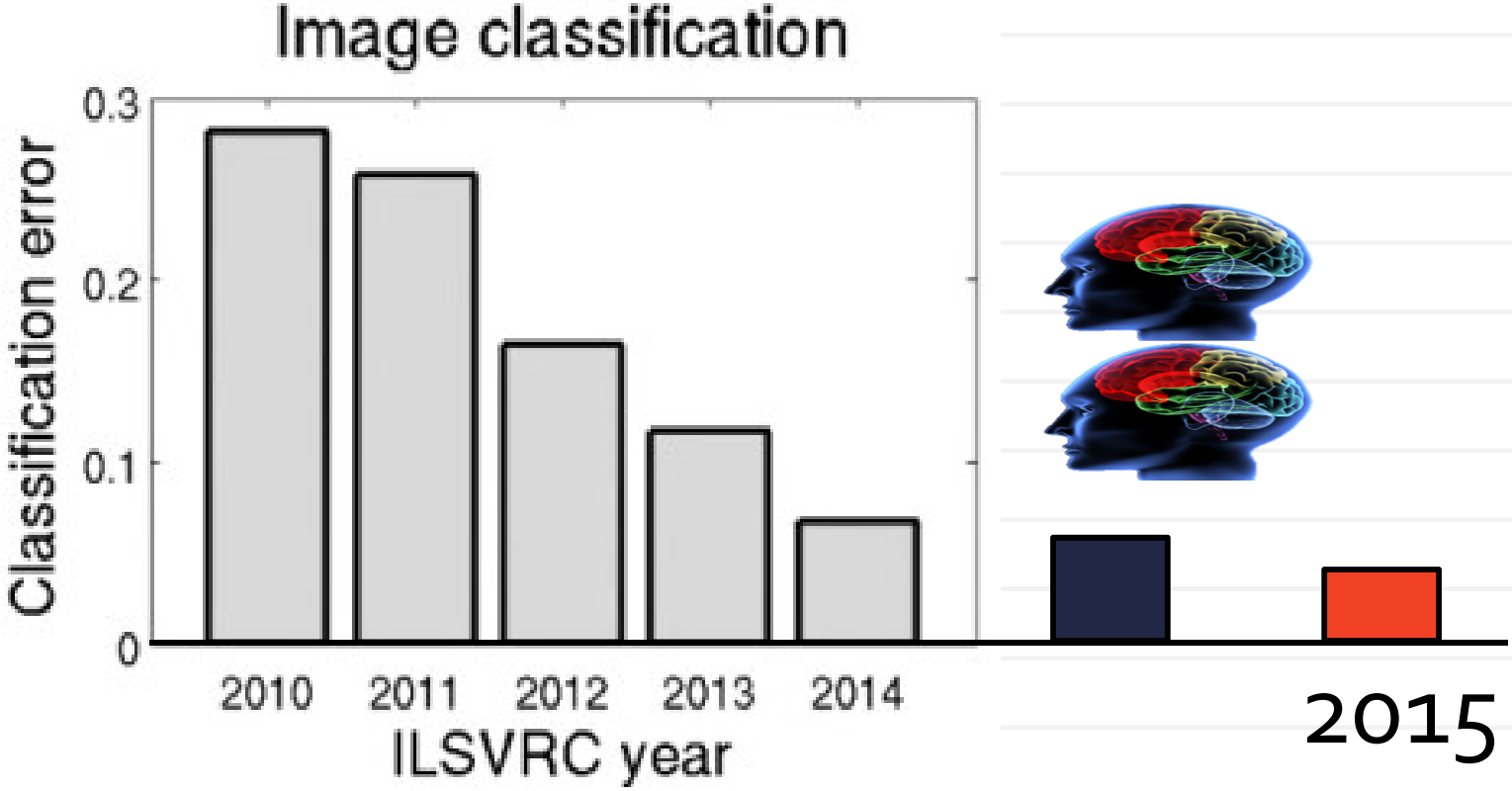
Big gains in many domains using **supervised learning**

Deep learning: recap

- *Data augmentation*
- Pre-learning bottom layers using unlabeled data (unsupervised learning)
- Transferring bottom layers from related supervised tasks
- *Domain adaptation:* same task, shifted data
- *Multi-task learning:* related task, similar data

Need intermediate level representations!

Recent highlights: vision



Recent highlights: graphics

- Neural style [Gatys et al. CVPR 2016]
- NB: deep learning not so strongly needed! [Ustyuzhaninov et al. ICLR 2017]

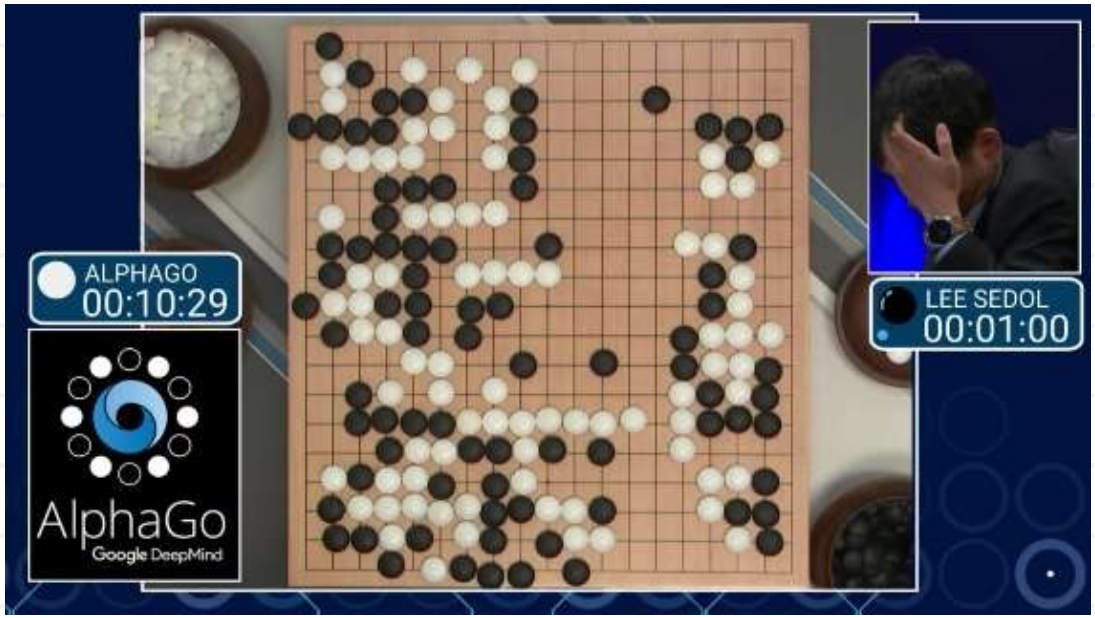
Recent highlights: machine translation

Source	An admitting privilege is the right of a doctor to admit a patient to a hospital or a medical centre to carry out a diagnosis or a procedure, based on his status as a health care worker at a hospital.
Reference	Le privilège d'admission est le droit d'un médecin, en vertu de son statut de membre soignant d'un hôpital, d'admettre un patient dans un hôpital ou un centre médical afin d'y délivrer un diagnostic ou un traitement.
RNNenc-50	Un privilège d'admission est le droit d'un médecin de reconnaître un patient à l'hôpital ou un centre médical d'un diagnostic ou de prendre un diagnostic en fonction de son état de santé.
RNNsearch-50	Un privilège d'admission est le droit d'un médecin d'admettre un patient à un hôpital ou un centre médical pour effectuer un diagnostic ou une procédure, selon son statut de travailleur des soins de santé à l'hôpital.

[Bahdanau et al. ICLR 2015]

Model	All	No UNK°
RNNencdec-30	13.93	24.19
RNNsearch-30	21.50	31.44
RNNencdec-50	17.82	26.71
RNNsearch-50	26.75	34.16
RNNsearch-50*	28.45	36.15
Moses	33.30	35.63

Recent highlights: games



AlphaGo [DeepMind, Nature'2016]

More examples:

- Speech recognition
- Deep genomics
- Speech synthesis
- ….

What to expect

Small gains:

- Machine translation
- Computer graphics
- Speech synthesis
- Some games
- More work to be done ☺

BIG GAINS:

- Computer vision
- Speech recognition
- Some games
- Lots of annotated training data
- Hard to hand-craft intermediate representations