Recognition and Classification in Images and Video

203.4780

http://www.cs.haifa.ac.il/~rita/visual_recog_course/Visual%20Recognition%20Spring%202014.htm

Introduction

The course is based on UT-Austin course: Special Topics in Computer Vision,
by Kristen Grauman.

Course overview

- Graduate course in computer vision.
- We will survey and discuss vision papers relating to object and activity recognition and scene understanding.
- The goal: understand classical and modern approaches to some important problems, analyzing their strengths and weaknesses, and identifying interesting open questions.

Requirements

- Paper review each week
- Participating in discussions
- Presenting one topic in a class (in pairs)
- Programming project (in pairs)
- Presentations are due one week before the slot your presentation is scheduled.
  - you will need to read the papers, create slides one week before the date you are signed up
  - **requires my approval for your presentation**.

Paper review

- Submit review for one of the assigned papers (with *) each week, except for the week of your presentation.
- **Reviews are due by 9 PM on the night before** class (Sunday).
- Email reviews to me, pasting the text directly into your mail (no attachments). Include [4738] in the subject header.

The review should address:

- Give a summary of the paper in your own words (very brief, 2-3 sentences)
- What is the main contribution of the paper?
- What are the primary strengths and weaknesses of the paper?
- How convincing are the experiments? If something specific is lacking, what should have been tested?
- Describe one specific way in which the work could be extended (bonus).
- Additional comments, including unclear points, connections you see between the papers.

Paper presentation

- Each team (of 2 students) will give a presentation in class covering 2 papers on a topic selected from the course syllabus list.
- The talk should be well-organized and polished, sticking to about 40 minutes (20 min. each student).
- Run through it beforehand and check the time (a good rule of thumb: a minute per slide).

Presentation should cover

- Clear statement of the problem
- Why the problem is interesting, important, difficult?
- Key technical ideas, how they work, main contributions, strengths and weaknesses
- Evaluation, summary of key experiments and data
- Open issues raised in the papers, likely extensions

Presentation guidelines

- Try to use applications to motivate the work when possible, and look for visual elements (images, videos) to put in the presentation.
- Check out the webpages linked on the class webpage, and also look at authors webpages for supplementary materials.
- It's ok to grab a few slides from conference talks etc. when available, but be sure to **clearly cite the source on each slide** that is not your own.

Project

- A project could be built around one of the topics of the syllabus
- It should be done with a partner.
- Experimental evaluation should be done on a benchmark data set (provided in the course page)
- You can use papers provided as an additional reading to choose your project.
- Initial project proposals will be due before the **middle of the term**.

Suggested Projects

- Object recognition using bag-of-words representation and discriminative learning.
- Face detection
- Pedestrian detection
- Face recognition
- Saliency in images
- Action recognition
- Image retrieval

Grading

- 30% participation, includes
  - attendance,
  - in-class discussions,
  - paper reviews.
- 40% presentations, includes
  - drafts submitted one week prior,
  - in-class presentation.
- 30% final project, includes
  - implementation,
  - presentation,
  - final report.

Syllabus

A. Recognizing specific objects

Global Features:

- Linear Subspaces
- Detection as a binary decision

Local Features:

- Local features, matching for object instances
- Visual Vocabularies and Bag of Words

Region-based Features:

- Mid-Level Representations

B. Beyond Single objects (using additional information)

- Saliency
- Attributes
- Context

C. Scalability problems

- Scaling with the large number of categories
- Large-scale search

D. Action recognition in video and images

Syllabus

A. Recognizing specific objects

Global Features:

- Linear Subspaces
- Detection as a binary decision

Local Features:

- Local features, matching for object instances
- Visual Vocabularies and Bag of Words

Region-based Features:

- ~~Mid-Level Representations~~

B. Beyond Single objects (using additional information)

- ~~Saliency~~
- Attributes
- Context

C. Scalability problems

- Scaling with the large number of categories
- Large-scale search

D. Action recognition in video and images

Two of three

Object Recognition

So what does object recognition involve?

Verification: is that a bus?

Detection: locate the cars in the image

Identification: is that a picture of Mao?

Object categorization

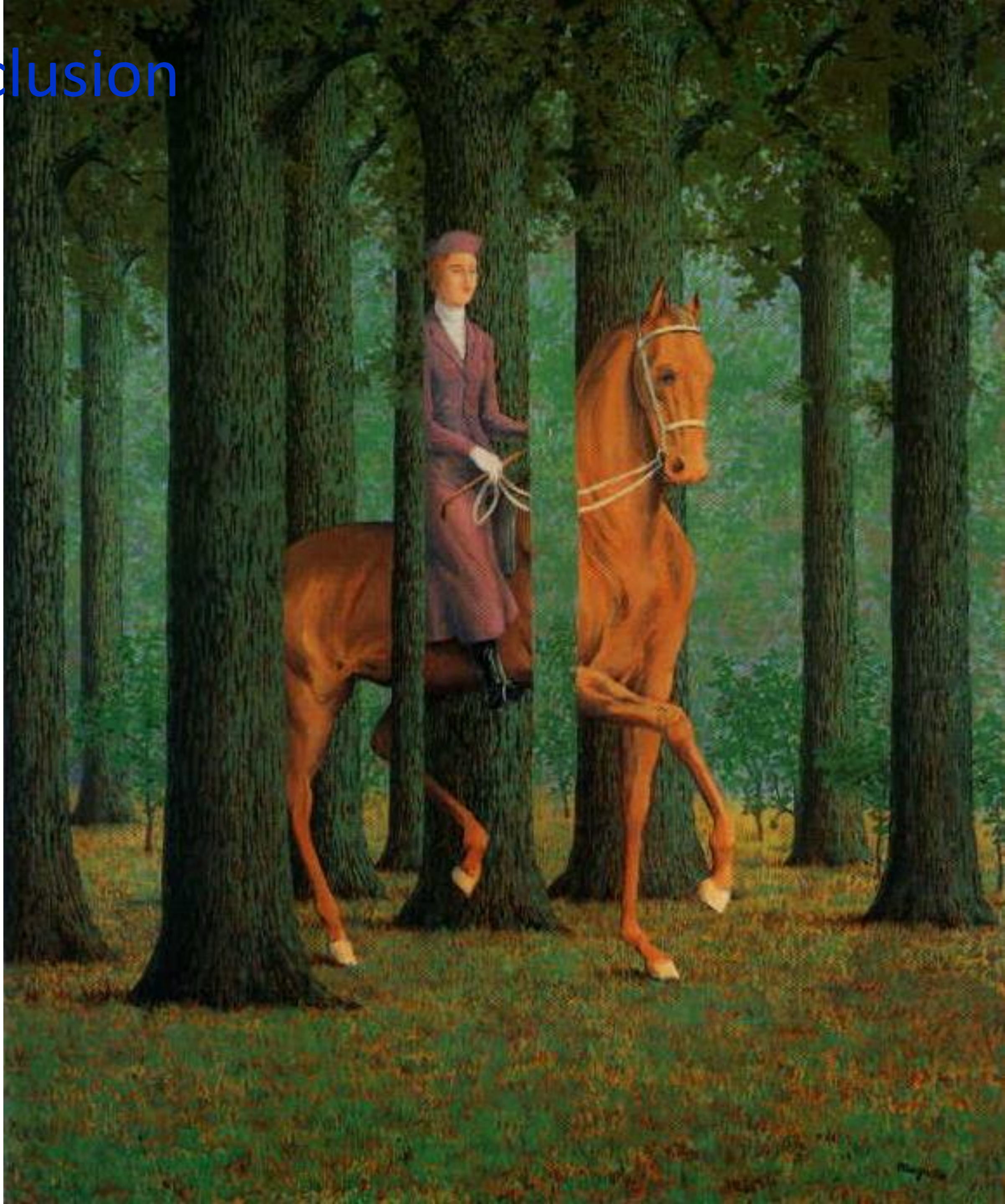
Challenges 1: view point variation

Michelangelo 1475-1564

Challenges 2: illumination

slide credit: S. Ullman

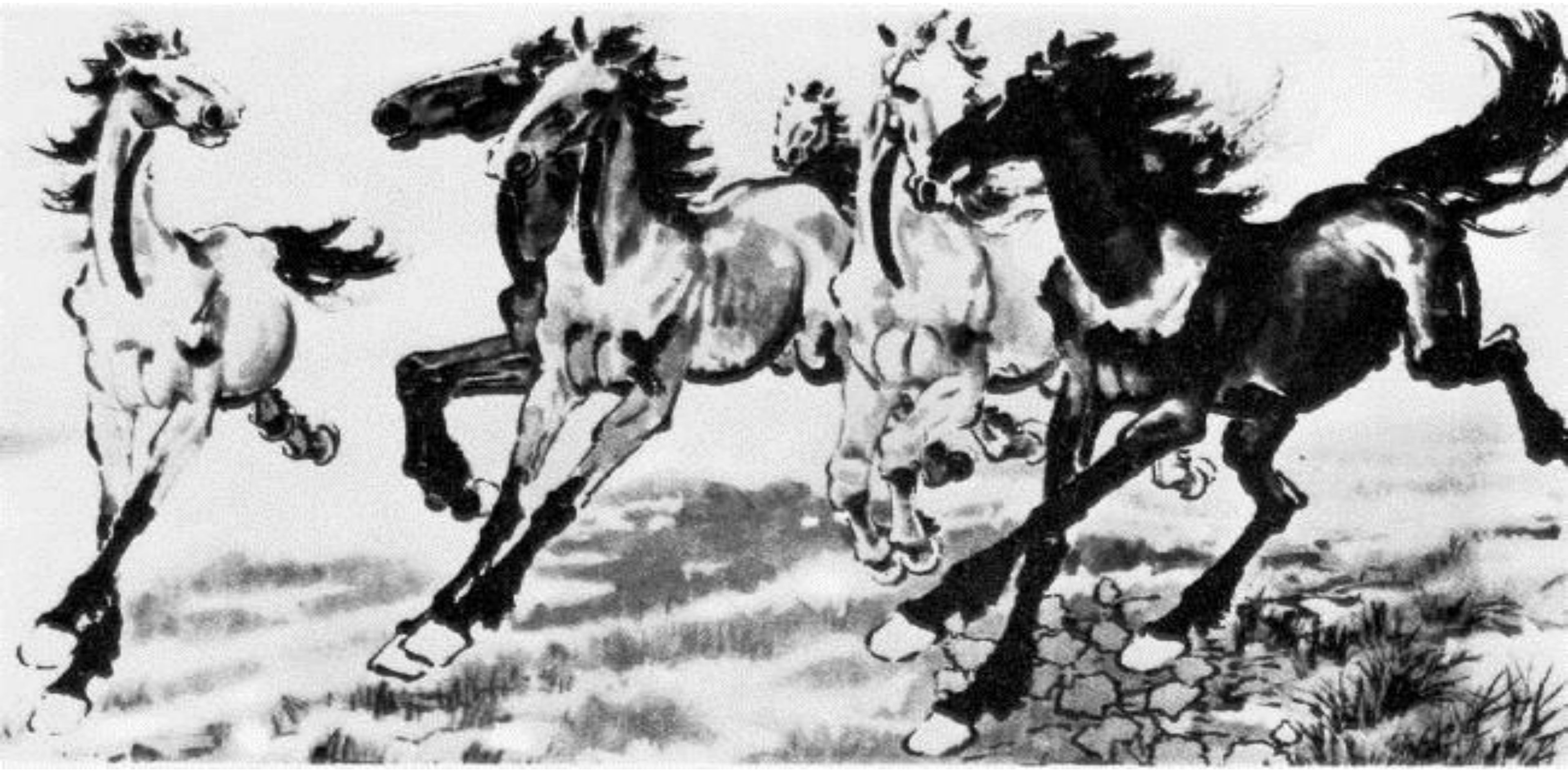
Challenges 3: occlusion



Magritte, 1957

Challenges 4: scale

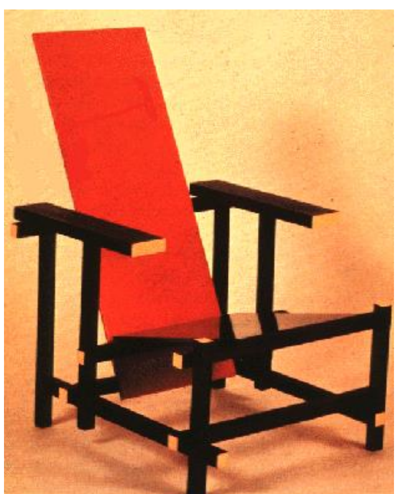
Challenges 5: deformation



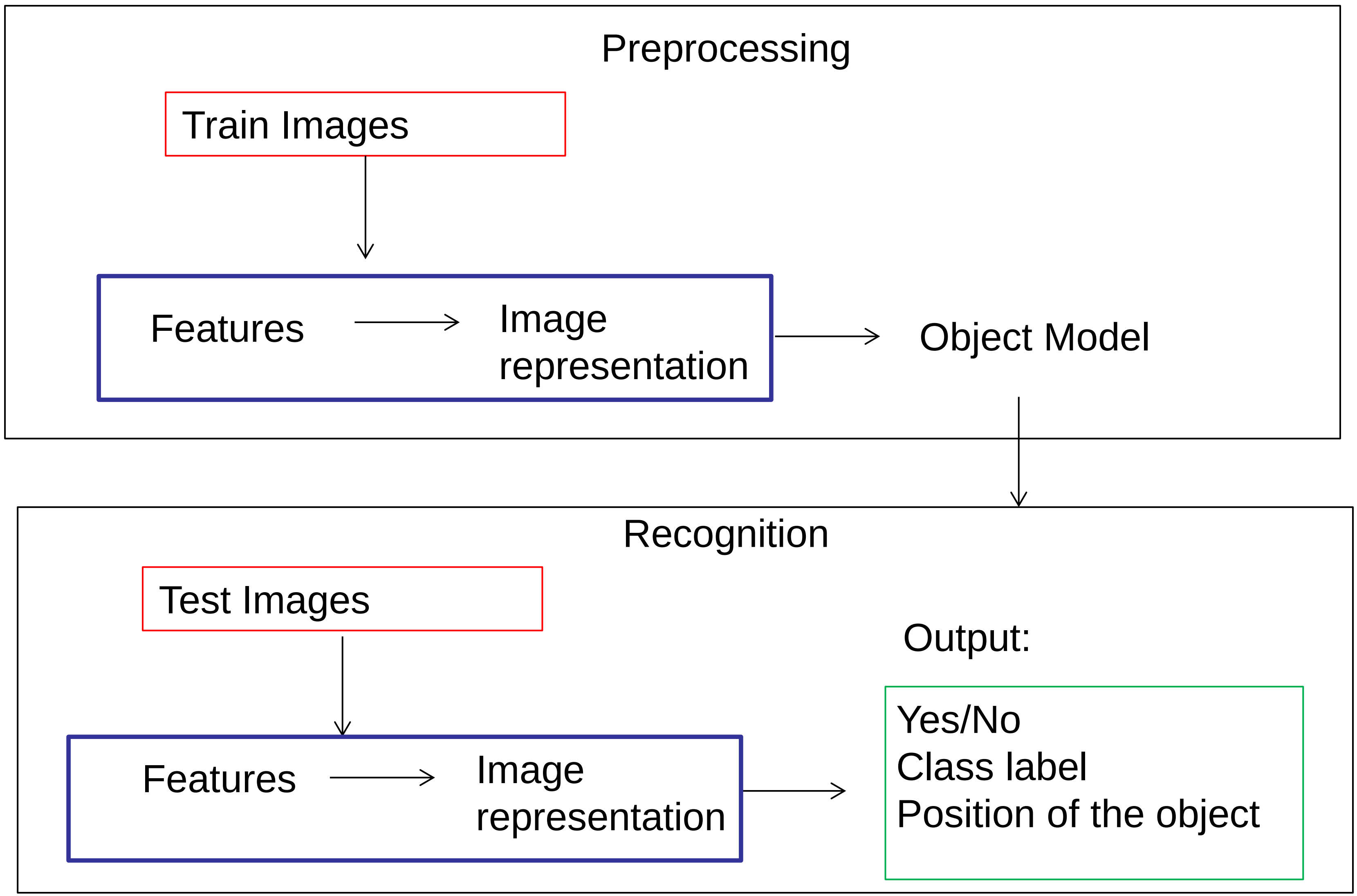
Xu, Beihong 1943

Challenges 7: intra-class variation



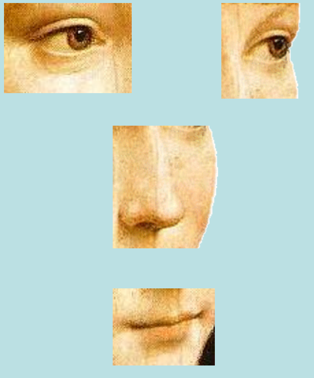
Recognition Steps



Features

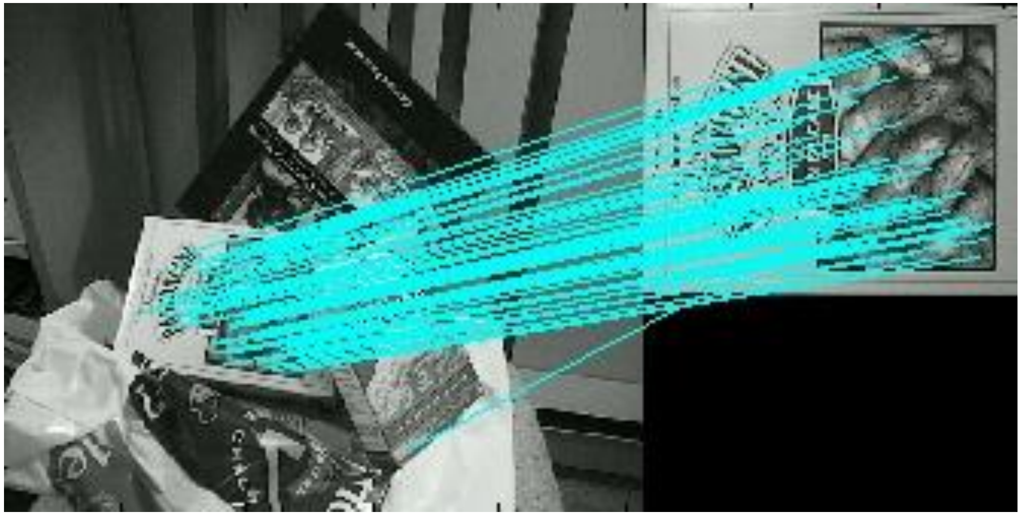
Global

Part-based



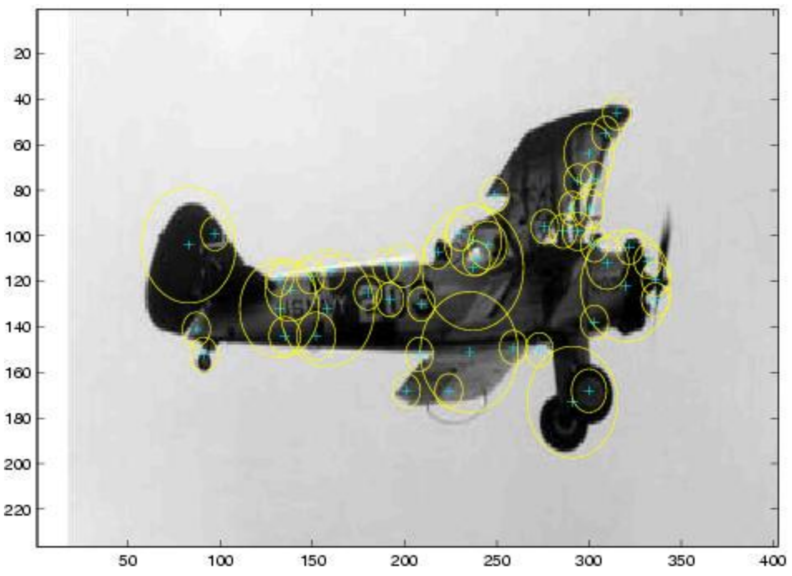
Recognition using local features

1. Find invariant local features
2. Match them with the model features
3. Vote on global geometric transformation



Interest Point Detectors

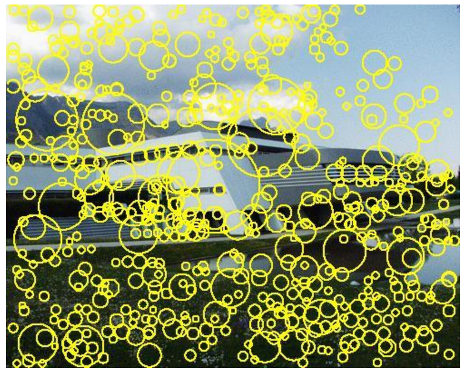
- Basic requirements:
  - Sparse
  - Informative
  - Repeatable
- Invariance
  - Rotation
  - Scale (Similarity)
  - Affine



Popular Detectors

Scale Invariant

Harris-Laplace



Difference of Gaussians

Laplace of Gaussians

Scale Saliency (Kadir-Braidy)

Affine Invariant

Harris-Laplace Affine

Difference of Gaussians Affine

Laplace of Gaussians Affine

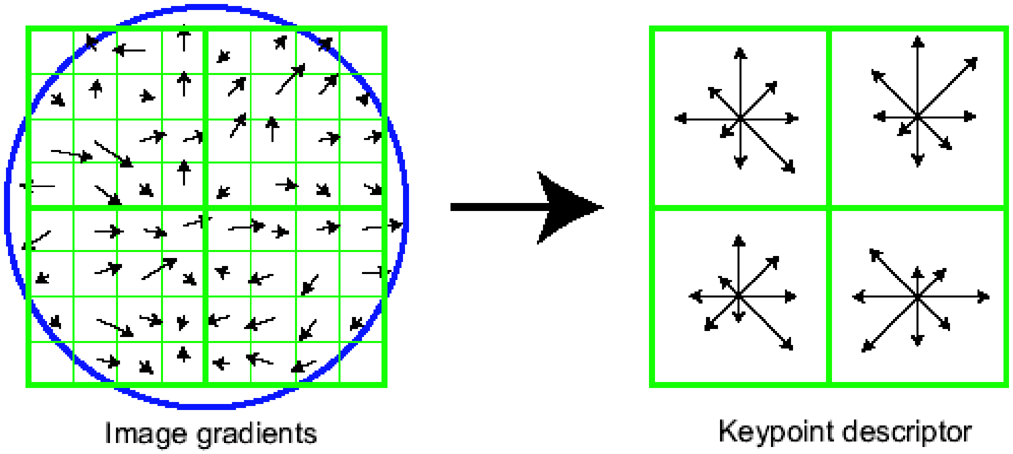
Affine Saliency (Kadir-Braidy)

Representation of appearance: Local Descriptors

- Invariance
  - Rotation
  - Scale
  - Affine
- Insensitive to small deformations
- Illumination invariance
  - Normalize out

SIFT – Scale Invariant Feature Transform

- Descriptor overview:
  - Determine scale (by maximizing DoG in scale and in space), local orientation (as the dominant gradient direction). Use this scale and orientation to make all further computations invariant to scale and rotation.
  - Compute gradient orientation histograms of several small windows (128 values for each point)
  - Normalize the descriptor to make it invariant to intensity change



David G. Lowe, **"Distinctive image features from scale-invariant keypoints,“** *International Journal of Computer Vision,* 60, 2 (2004), pp. 91-110.

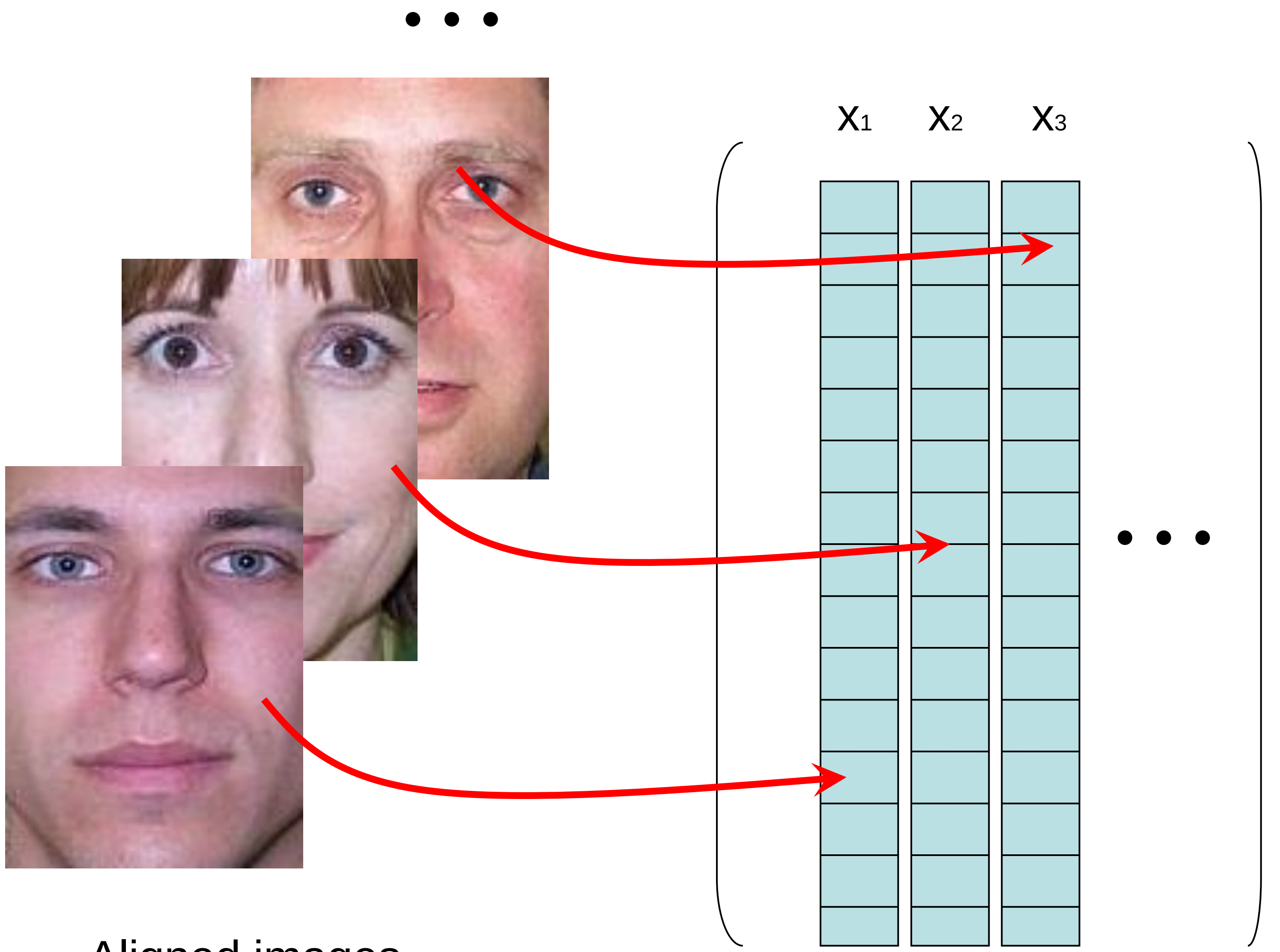
Recognizing Specific Objects

Learned models of local features, and got object outline from

Objects may then be found under occlusion and 3D rotation

Images from: David Lowe, **Object Recognition from Local Scale-Invariant Features** Proc. of the International Conference on Computer Vision,Corfu (Sept. 1999)

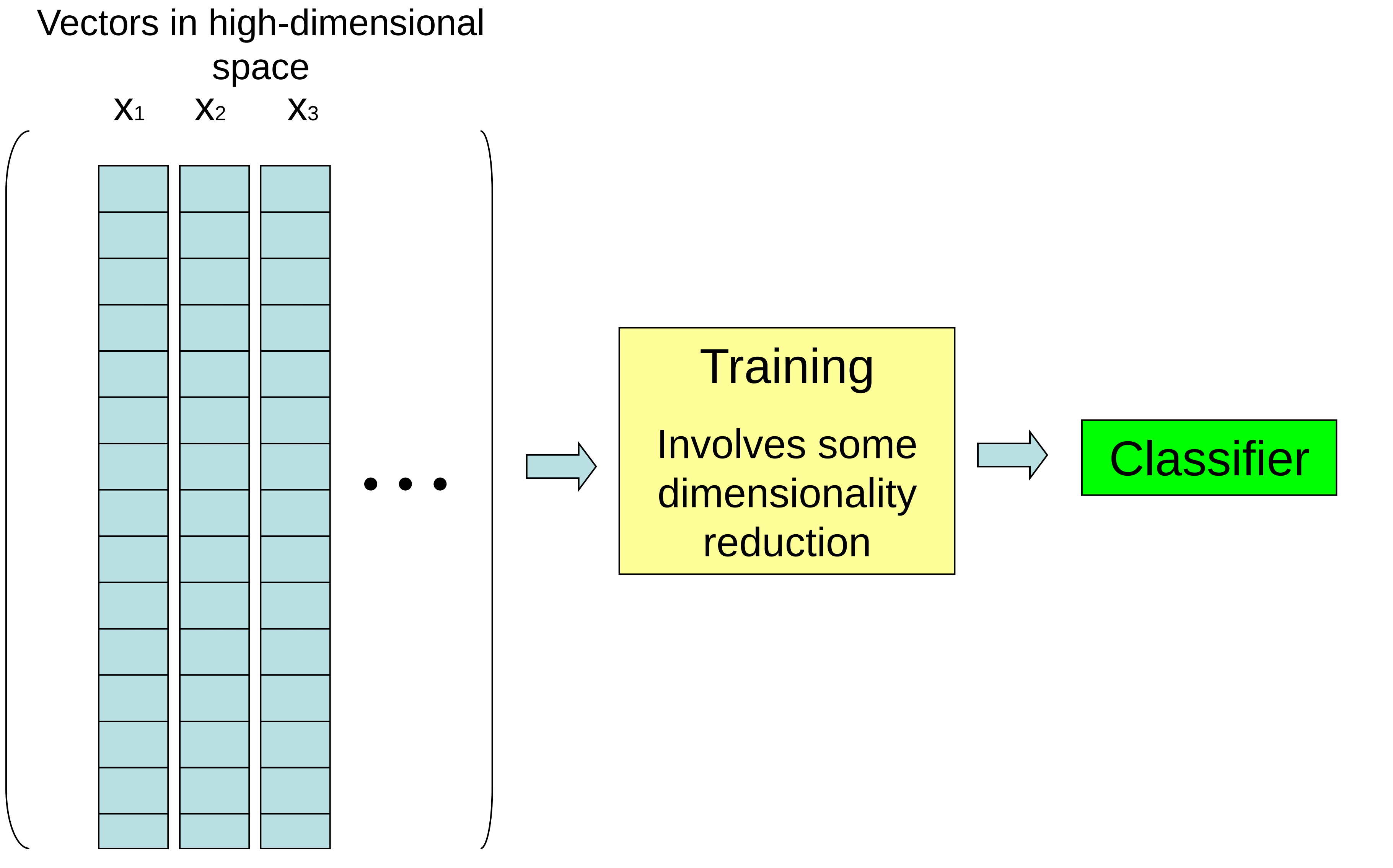
Recognition using global representations



Aligned images

Vectors in high-dimensional space

Global Approaches



Dimensionality Reduction

- Inputs (**high dimensional**)

  $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ points in R^d

- Outputs **(low dimensional)**

  $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$ points in R^k ($k \ll d$)

- Goal:

  combine old features $\boldsymbol{x}$ to create new features $\boldsymbol{y}$

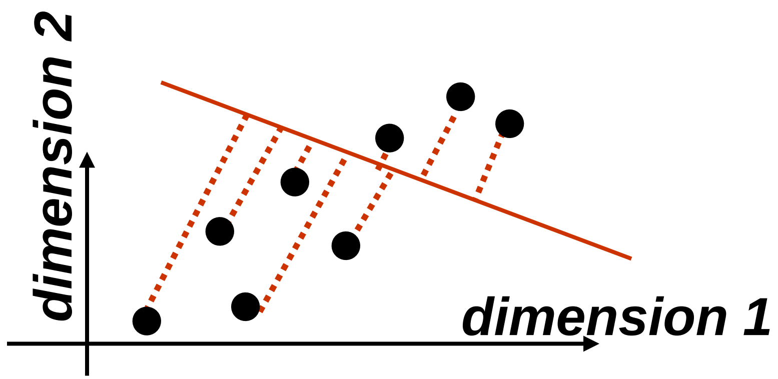
Linear Methods

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \Rightarrow W \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} w_{11} & \cdots & w_{1d} \\ \vdots & & \vdots \\ w_{k1} & \cdots & w_{kd} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_k \end{bmatrix} \quad \textbf{\textit{with } } k < d$$

- Principle component analysis PCA
- Fisher Linear Discriminant (FLD)

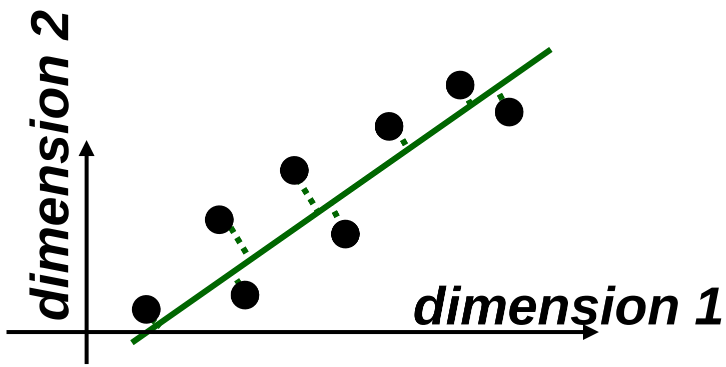
Principle Component Analysis

Good
representation



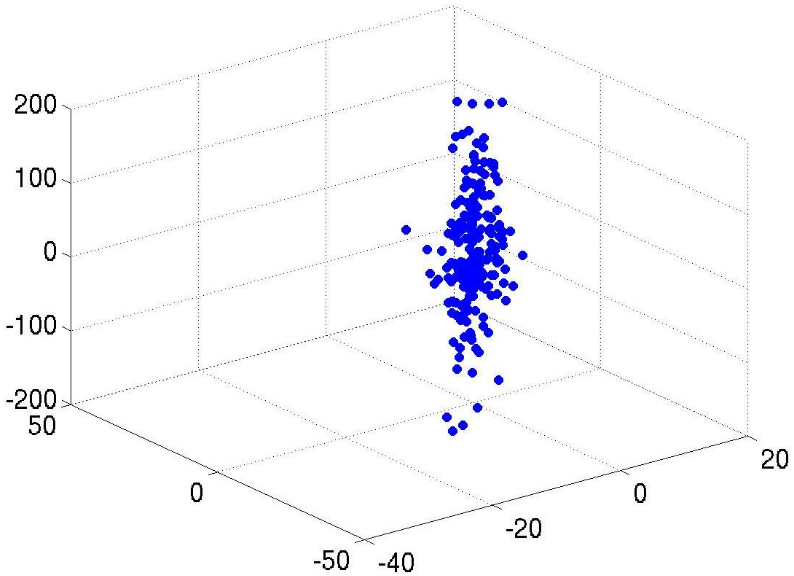
the projected data has a fairly large variance, and the points tend to be far from zero.

Poor
representation



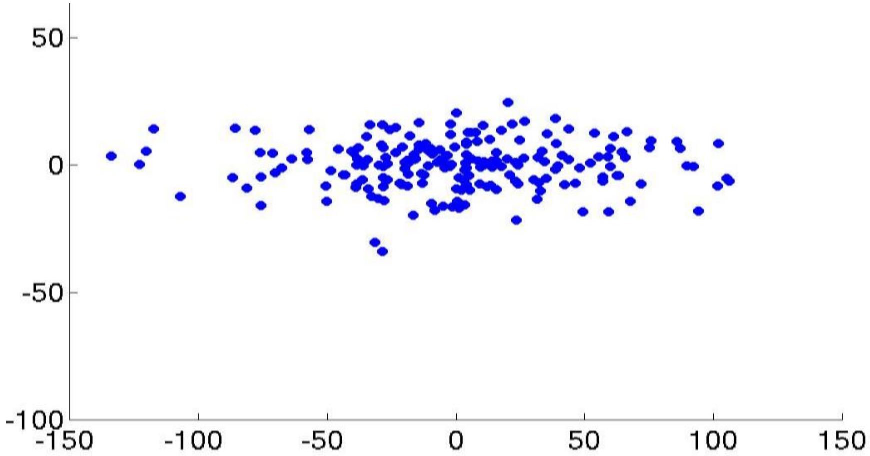
the projections have a significantly smaller variance, and are much closer to the origin.

PCA

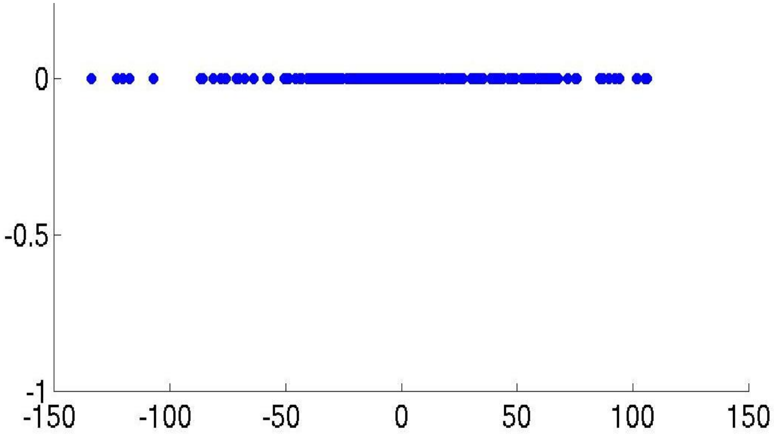


- Seek most accurate data representation in a lower dimensional space.
- The good direction/subspace to use for projection lies in the direction of largest variance.

best 2D approximation



best 1D approximation



Maximum Variance Subspace

- Assume inputs are centered: $\sum_{i=1}^{n} x_i = 0$
- Given a unit vector u and a point x, the length of the projection of x onto u is given by $x^T u$
- Maximize projected variance:

$$\mathrm{var}(y) = \frac{1}{n}\sum_i \left(x_i^T u\right)^2 = \frac{1}{n}\sum_i u^T x_i x_i^T u$$

$$= u^T \left(\frac{1}{n} x_i x_i^T\right) u$$

1D Subspace

- Maximizing $u^T C u$ subject to $\|u\| = 1$ where $C = \frac{1}{n}\sum_i x_i x_i^T$ is the empirical covariance matrix of the data, gives the principle eigenvector of *C*.

d-dimensional subspace

- to project the data into a d-dimensional subspace (k <<d), we should choose

  $u_1,\ldots,u_k$ to be the top k eigenvectors of *C*.
- $u_1,\ldots,u_k$ now form a new, orthogonal basis for the data.
- The low dimensional representation of x is given by

$$y = \begin{bmatrix} u_1^T x \\ u_2^T x \\ \vdots \\ u_k^T x \end{bmatrix}$$

Interpreting PCA

- Eigenvectors:

  principal axes of maximum variance subspace.

- Eigenvalues:

  variance of projected inputs along principle axes.

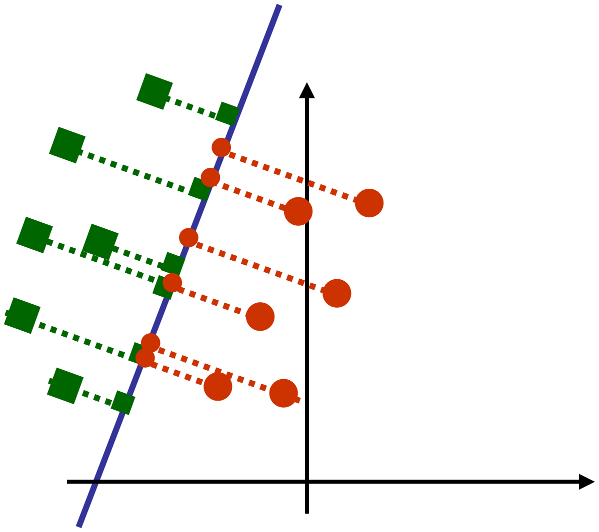
- Estimated dimensionality:

  number of significant (nonnegative) eigenvalues.

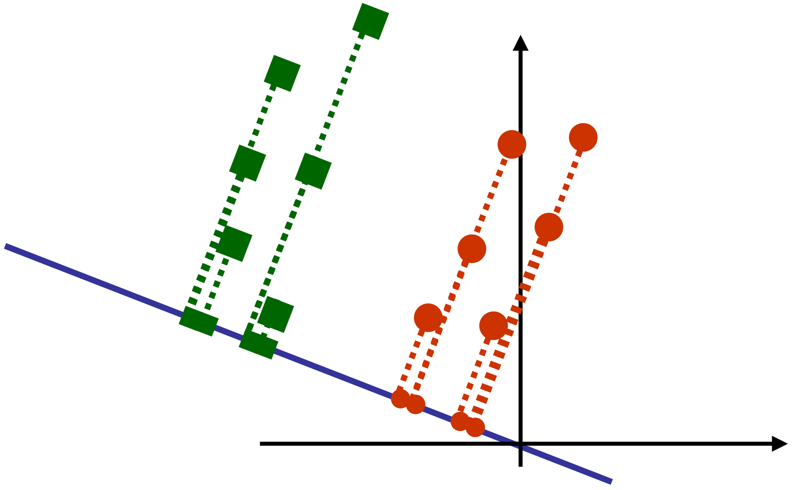
Fisher Linear Discriminant

- Main idea: find projection to a line s.t. samples from different classes are well separated.

Example in 2D



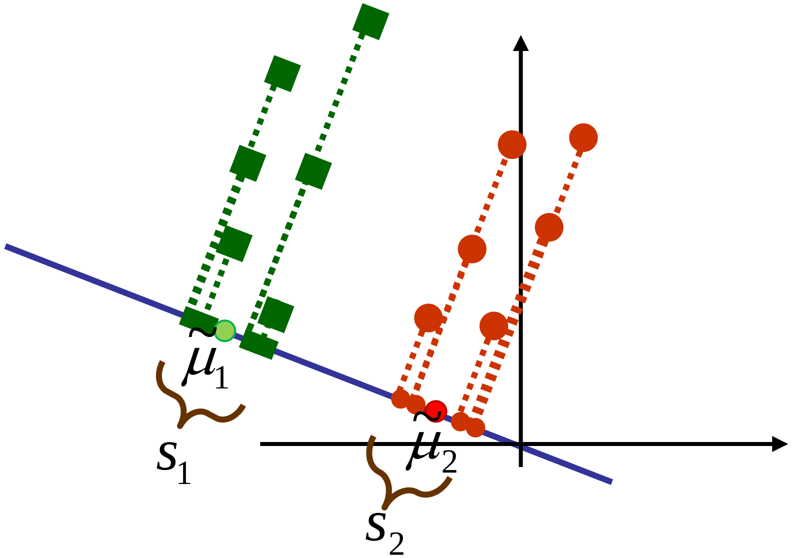
bad line to project to, classes are mixed up



good line to project to, classes are well separated

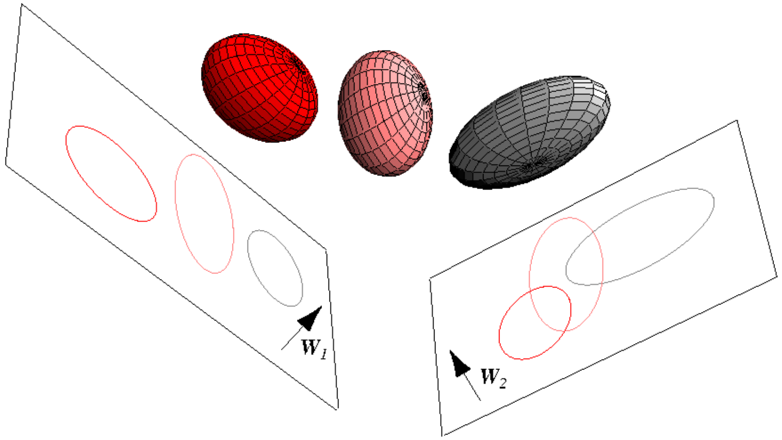
FLD

- Find projection that
  1. Maximizes the distance between class means $|\tilde{\mu}_1 - \tilde{\mu}_2|$
  2. Minimizes the scatter of the classes s_1, s_2



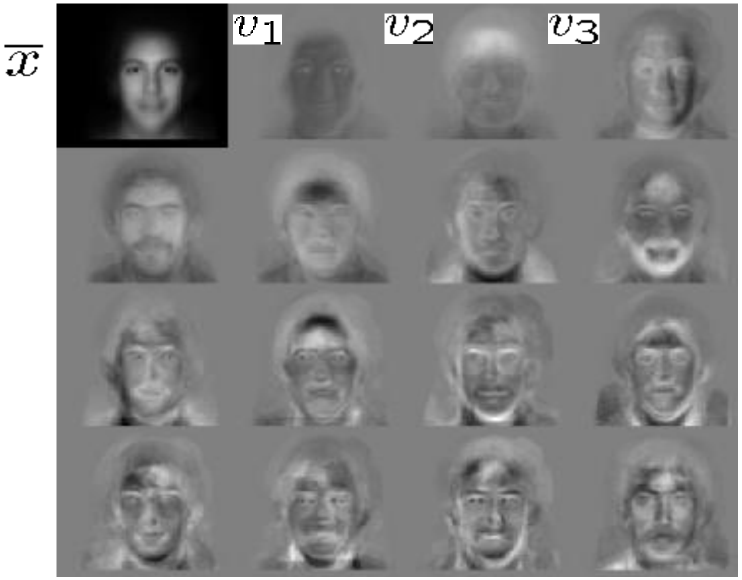
Multiple Discriminant Analysis

- Can generalize FLD to multiple classes
- In case of ***c*** classes, can reduce dimensionality to ***c***-1 dimensions or less.



Linear Subspaces

Eigenfaces (PCA)



top left image is linear combination of the rest.

Fisherfaces (FLD)

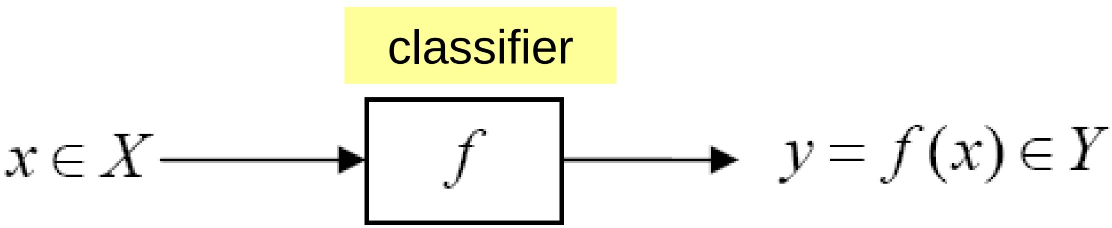
first four Fisherfaces from a set of 100 classes

Classification in Machine Learning

- A classifier is a function or an algorithm that maps every possible input (from a legal set of inputs) to a finite set of categories.
- X – input space, $\mathrm{x} \in \mathrm{X}$ sample from an input space.
- A typical input space is high-dimensional, for example $\mathrm{x} = \{x_1, \ldots, x_d\} \in R^d, \quad d > 1$. We also call x a feature vector.
- Ω is a finite set of categories to which the input samples belong: $\Omega = \{1, 2, \ldots, \mathrm{C}\}$.
- $w_i \in \Omega$ are called labels.

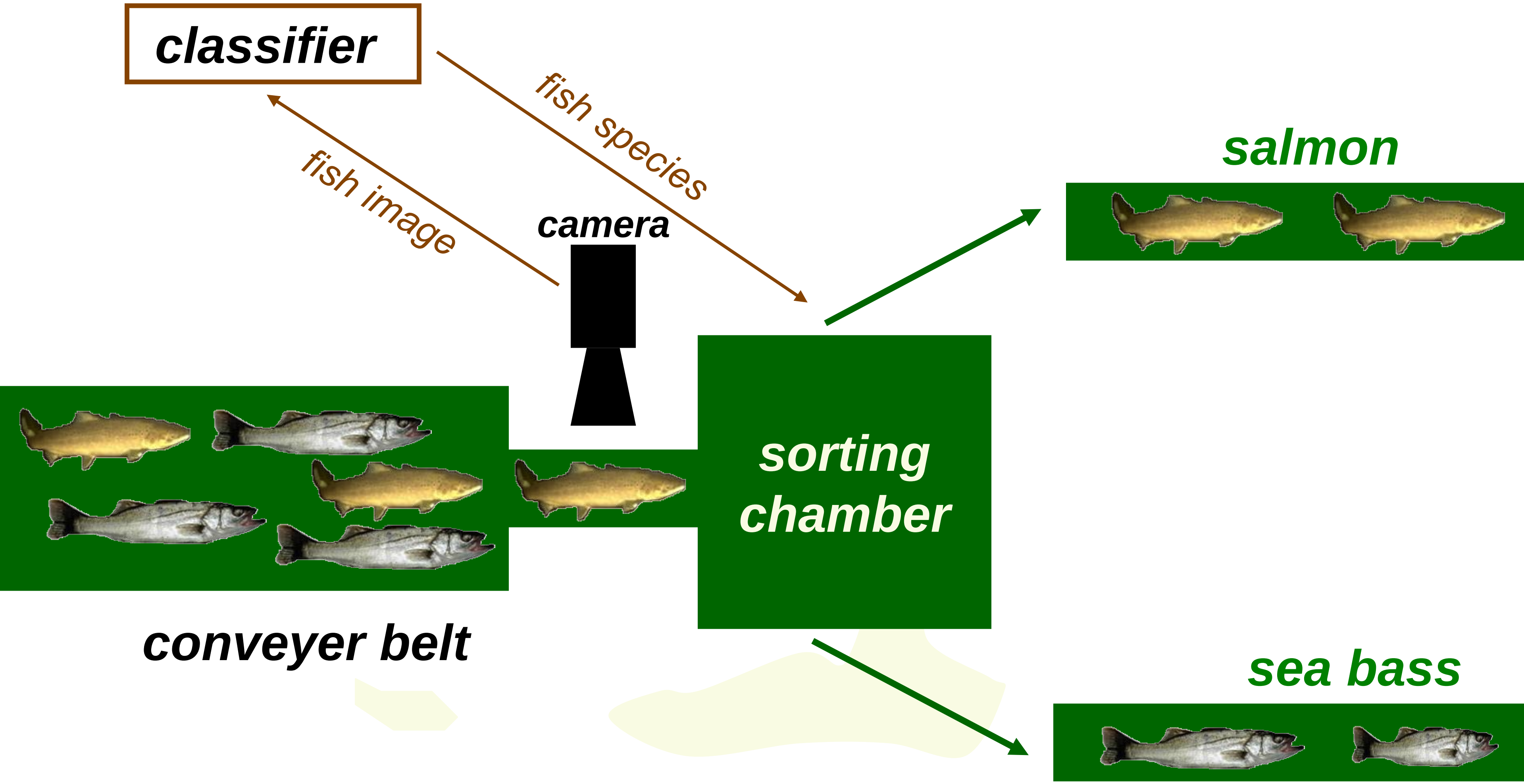
Definition of Classification

- Y is a finite set of decisions – the output set of the classifier.
- A classifier is a function $f:\mathrm{X}\rightarrow\mathrm{Y}$



- Classification is also called Pattern Recognition.

Toy Application: fish sorting



How to design a PR system?

- Collect data and classify by hand

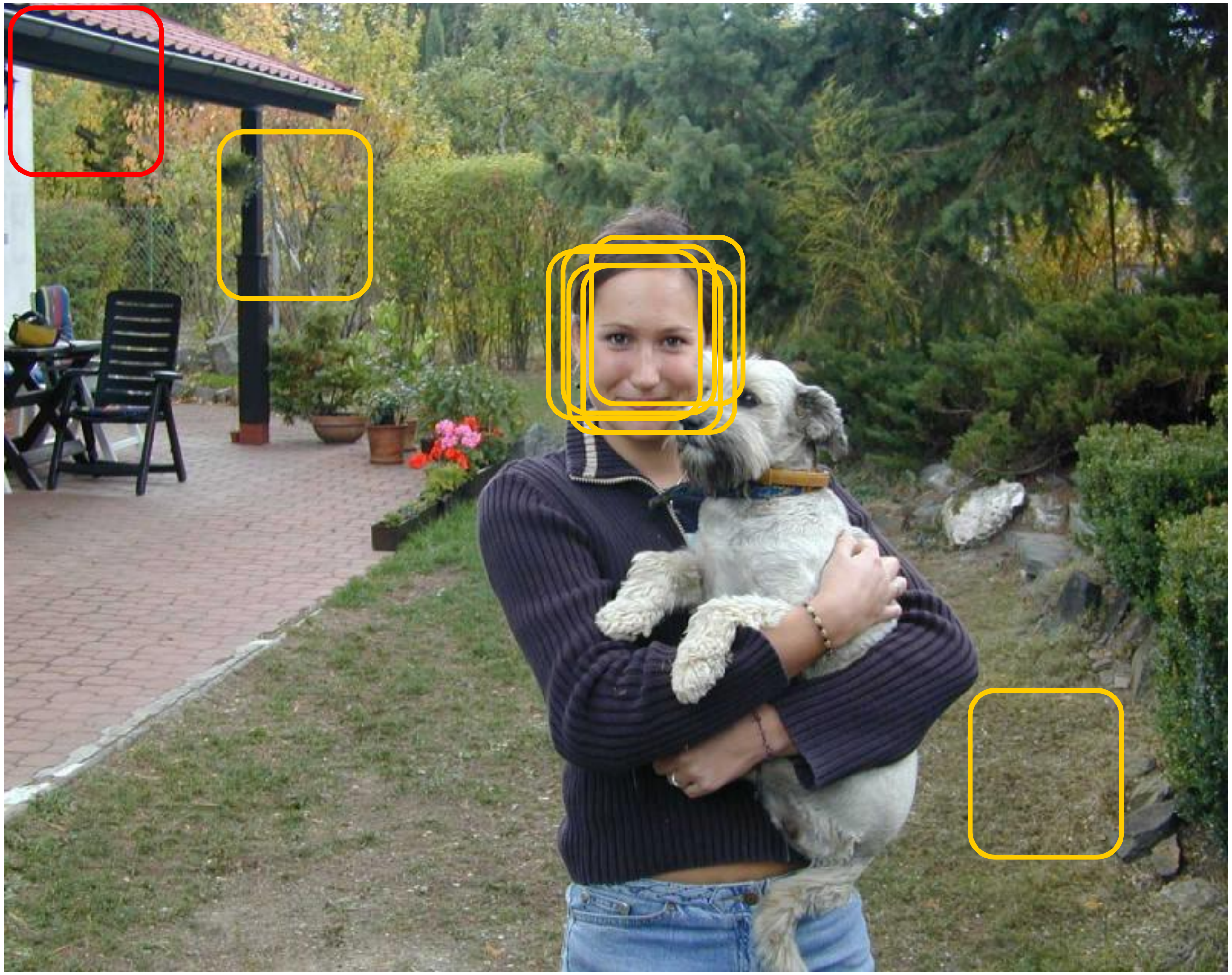
salmon *sea bass* *salmon* *salmon* *sea bass* *sea bass*

- Preprocess by segmenting fish from background

- Extract possibly discriminating features
  - length, lightness,width,number of fins,etc.
- Classifier design
  - Choose model
  - Train classifier on part of collected data (training data)
- Test classifier on the rest of collected data (test data) i.e. the data not used for training
  - Should classify new data (new fish images) well

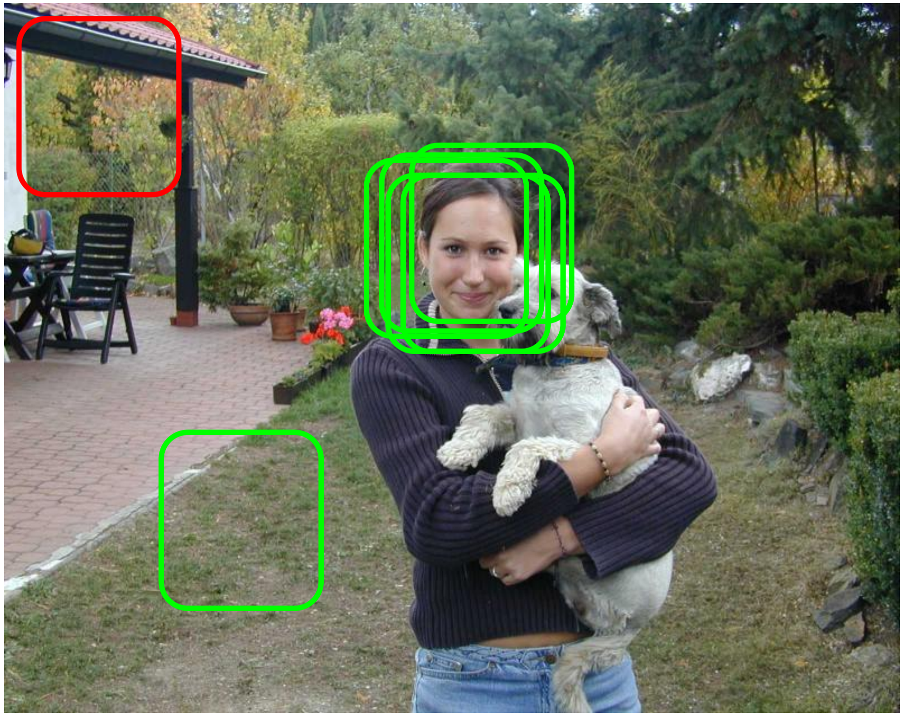
Detection

– Scale / position range to search over



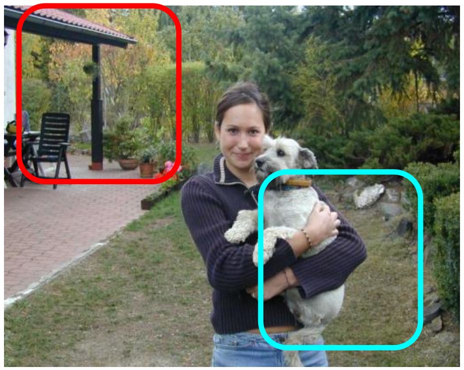
Detection

– Scale / position range to search over



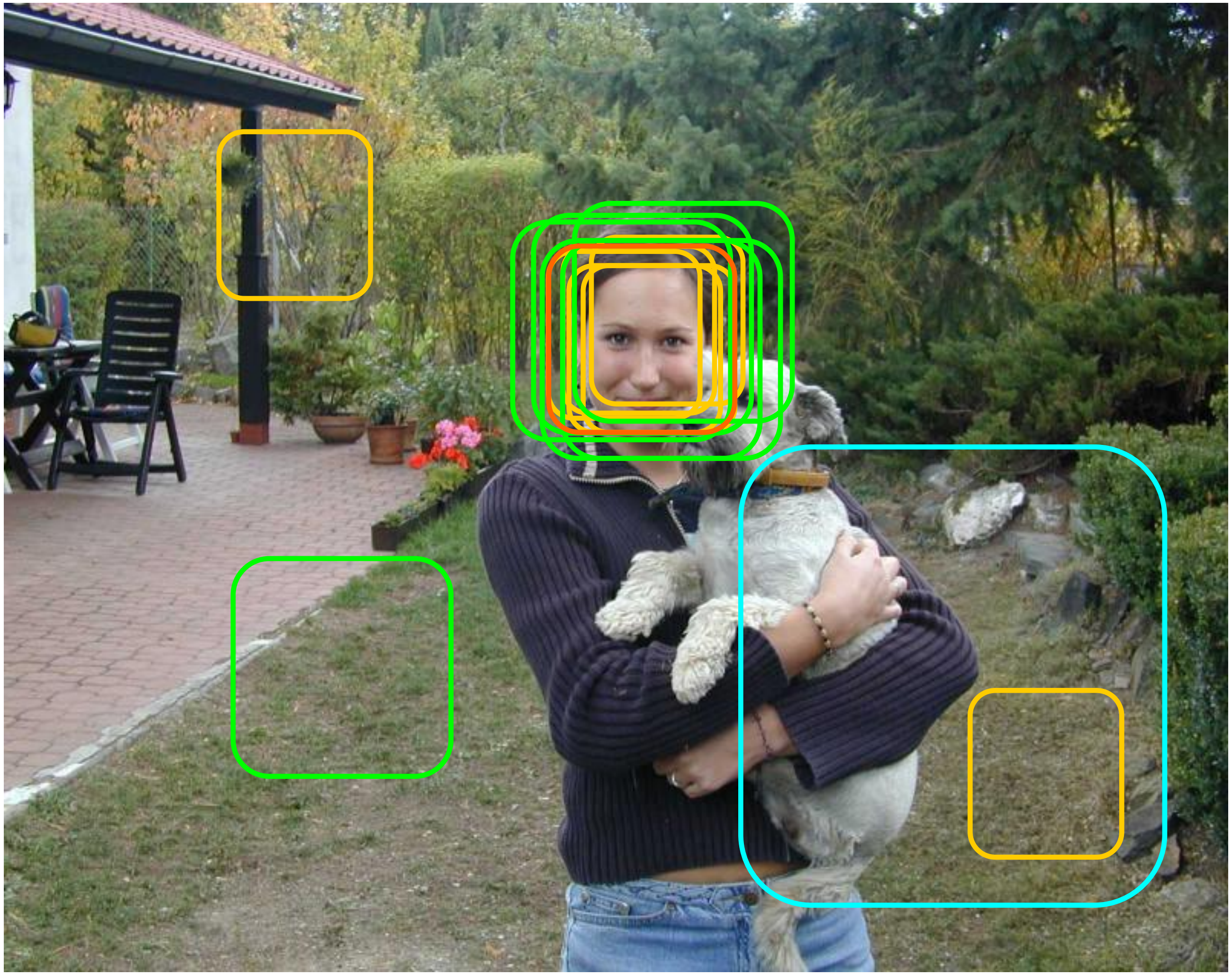
Detection

– Scale / position range to search over



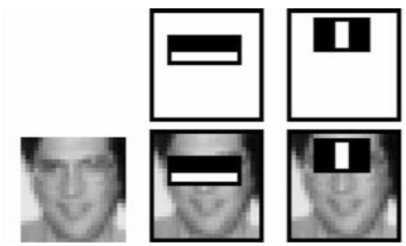
Detection

– Combine detection over space and scale.



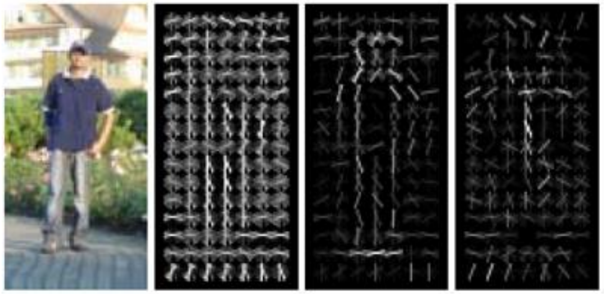
Detection as a binary decision

- Sliding window detection, detection as a binary decision problem.



Boosted Cascade of Simple Features

Adaboost



Histograms of Oriented Gradients for Human Detection

SVM

Boosting, motivation

- It is usually hard to design an accurate classifier which generalizes well
- However it is usually easy to find many “rule of thumb” *weak* classifiers
  - A classifier is weak if it is only slightly better than random guessing
- Can we combine several weak classifiers to produce an accurate classifier?
  - Question people have been working on since 1980’s

AdaBoost

- Let's assume we have 2-class classification problem, with $x_i \in R^n$, $y_i \in \{-1,1\}$
- AdaBoost will produce a discriminant function:

$$g(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

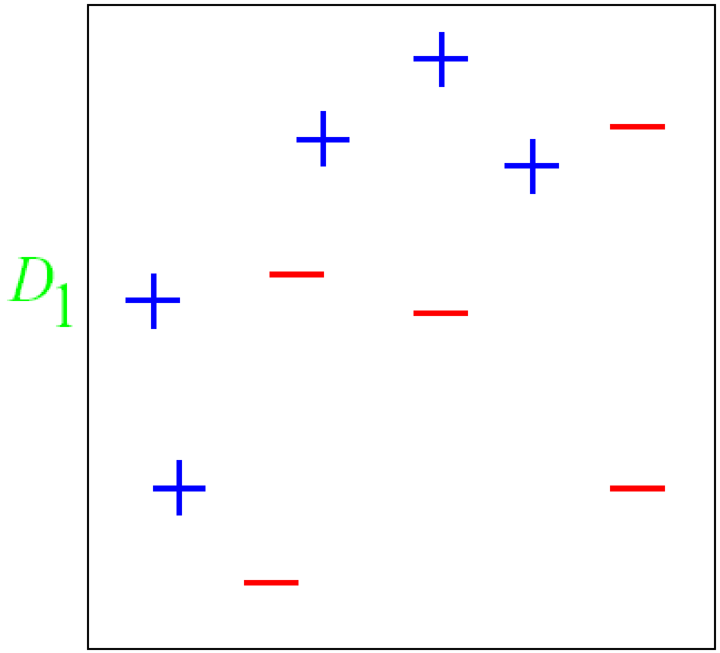
- where $h_t(x)$ is the "weak" classifier
- The final classifier is the sign of the discriminant function, that is $f_{final}(x) = \text{sign}[g(x)]$

AdaBoost

- $d(x)$ is the distribution of weights over the N training points $\sum d(x_i)=1$
- Initially assign uniform weights $d_0(x_i) = 1/N$ for all x_i
- At each iteration t :
  - Find best weak classifier $h_t(x)$ using weights $d_t(x)$
  - Compute the error rate ε_t as
    $\varepsilon_t = \sum_{i=1\ldots N} d_t(x_i) \cdot I[y_i \neq h_t(x_i)]$
  - assign weight α_t the classifier h_t's in the final hypothesis
    $$\alpha_t = \log((1 - \varepsilon_t)/\varepsilon_t)$$
  - For each x_i, $d_{t+1}(x_i) = d_t(x_i) \cdot \exp[\alpha_t \cdot I(y_i \neq h_t(x_i))]$
  - Normalize $d_{t+1}(x_i)$ so that $\sum_{i=1} d_{t+1}(x_i) = 1$
- $f_{FINAL}(x) = \text{sign}\left[\sum \alpha_t h_t(x)\right]$

AdaBoost Example

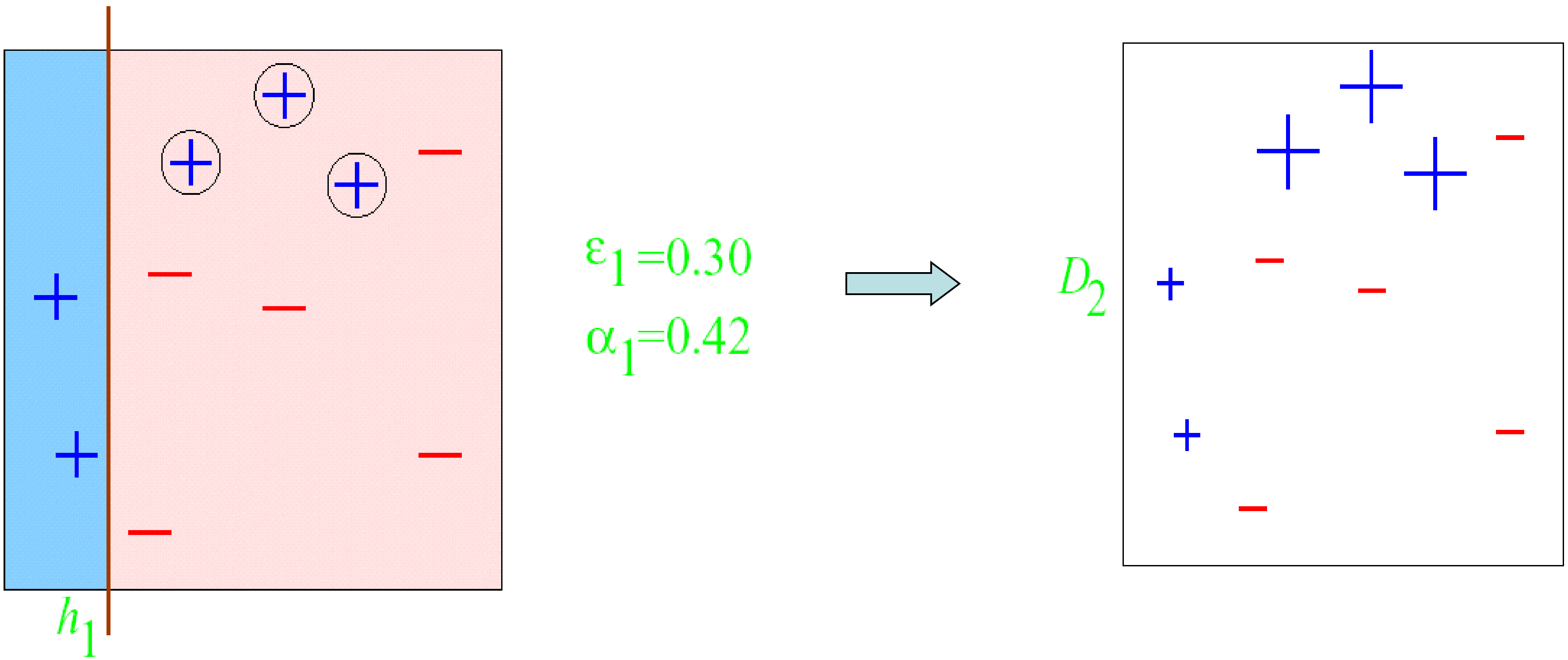
from "A Tutorial on Boosting" by Yoav Freund and Rob Schapire



Original Training set : equal weights to all training samples

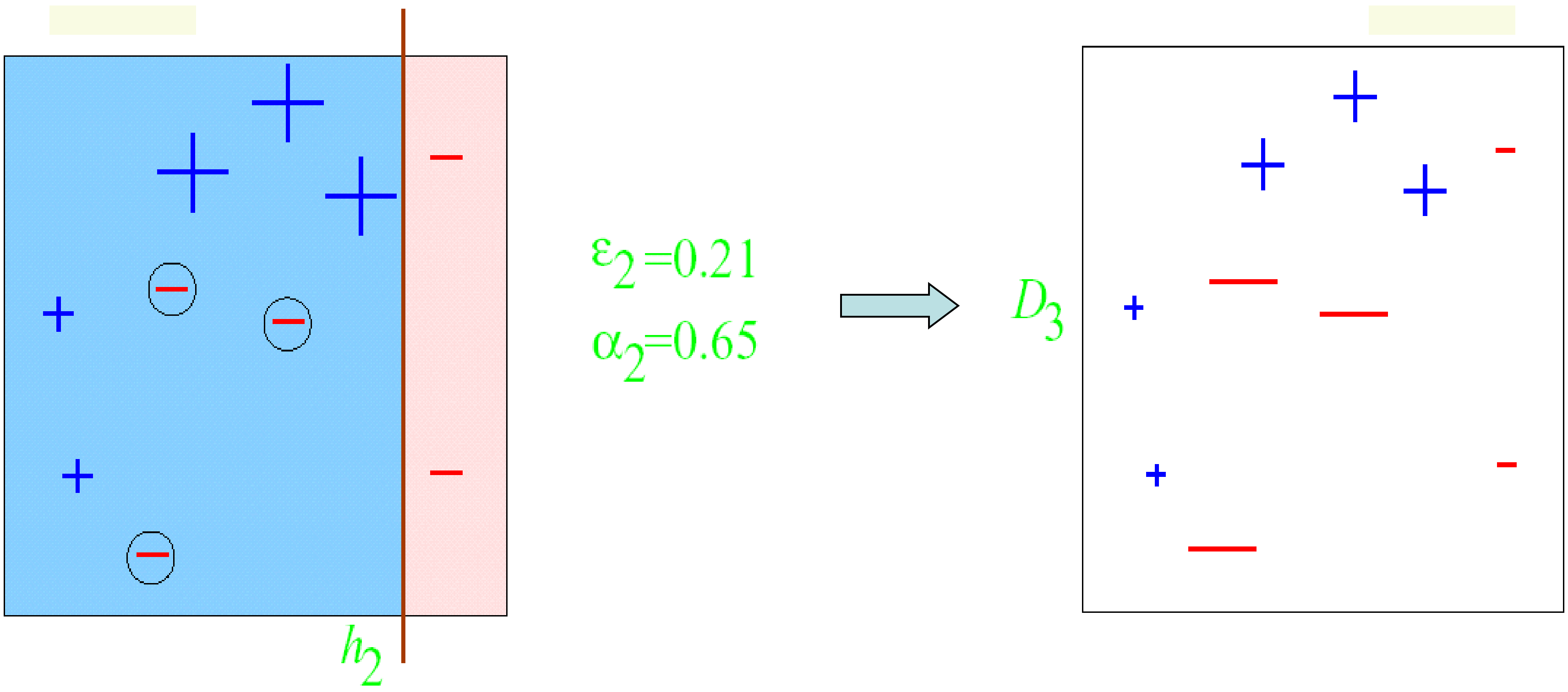
AdaBoost Example

ROUND 1



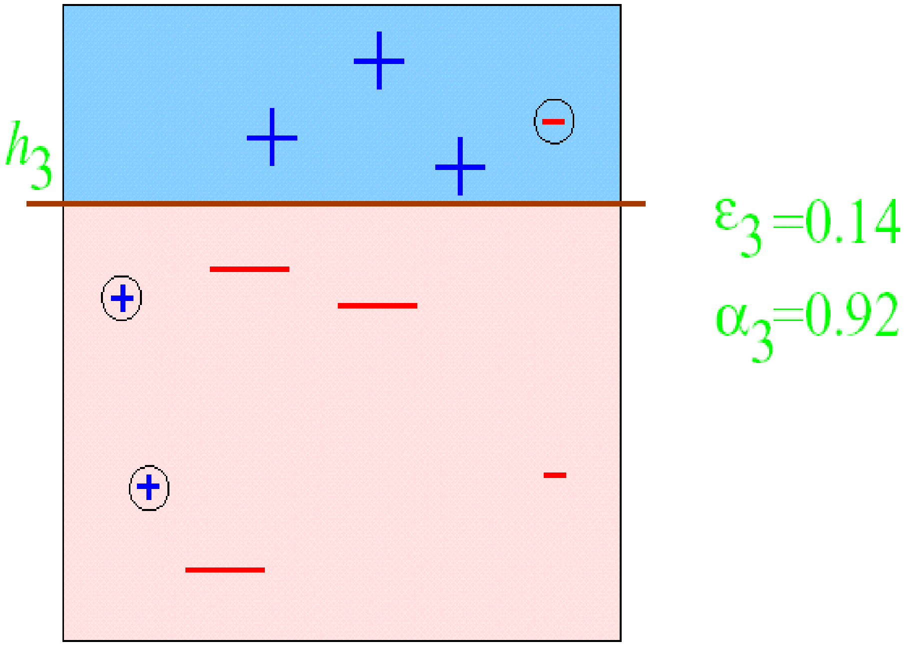
AdaBoost Example

ROUND 2

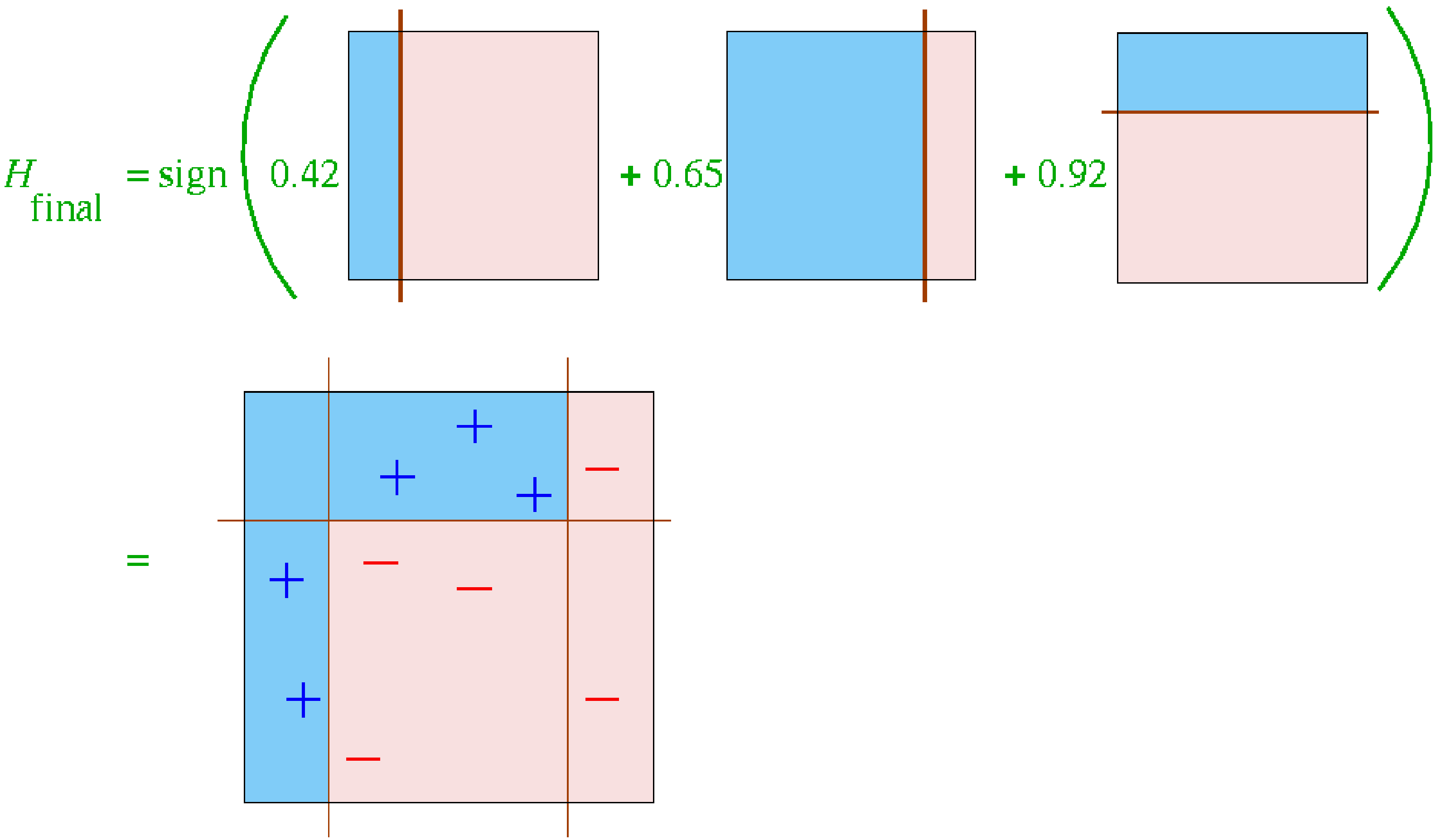


AdaBoost Example

ROUND 3



AdaBoost Example



SVM Problem Definition

Consider a training set of n iid samples

$$(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_n, \boldsymbol{y}_n)$$

where $\boldsymbol{x}_i$ is a vector of length m and

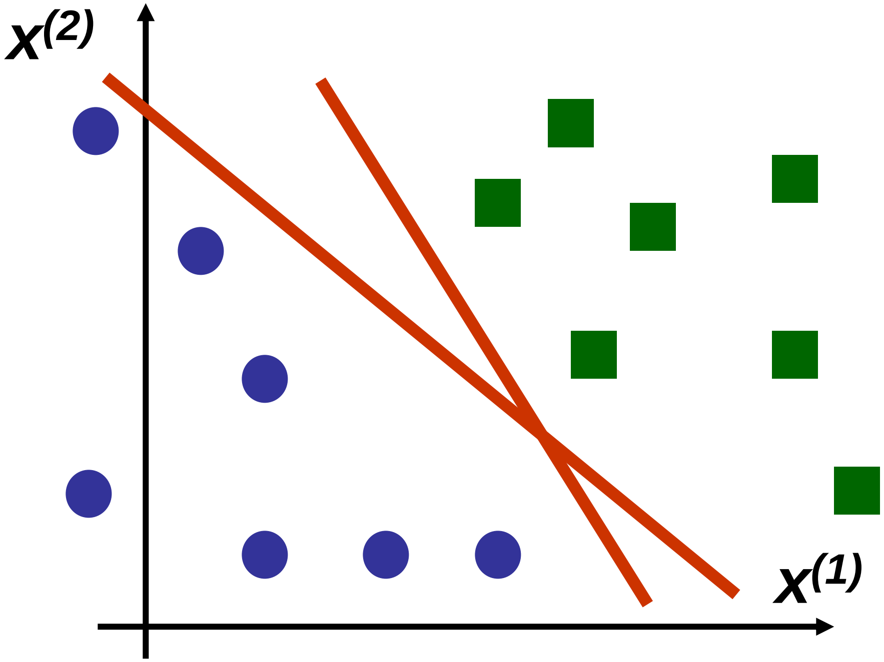
$\boldsymbol{y}_i \in \{+1, -1\}$ is the class label for data point $\boldsymbol{x}_i$.

Find a separating hyperplane $\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b} = \boldsymbol{0}$

corresponding to the decision function

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{sign}(\boldsymbol{w} \cdot \boldsymbol{x} + \boldsymbol{b})$$

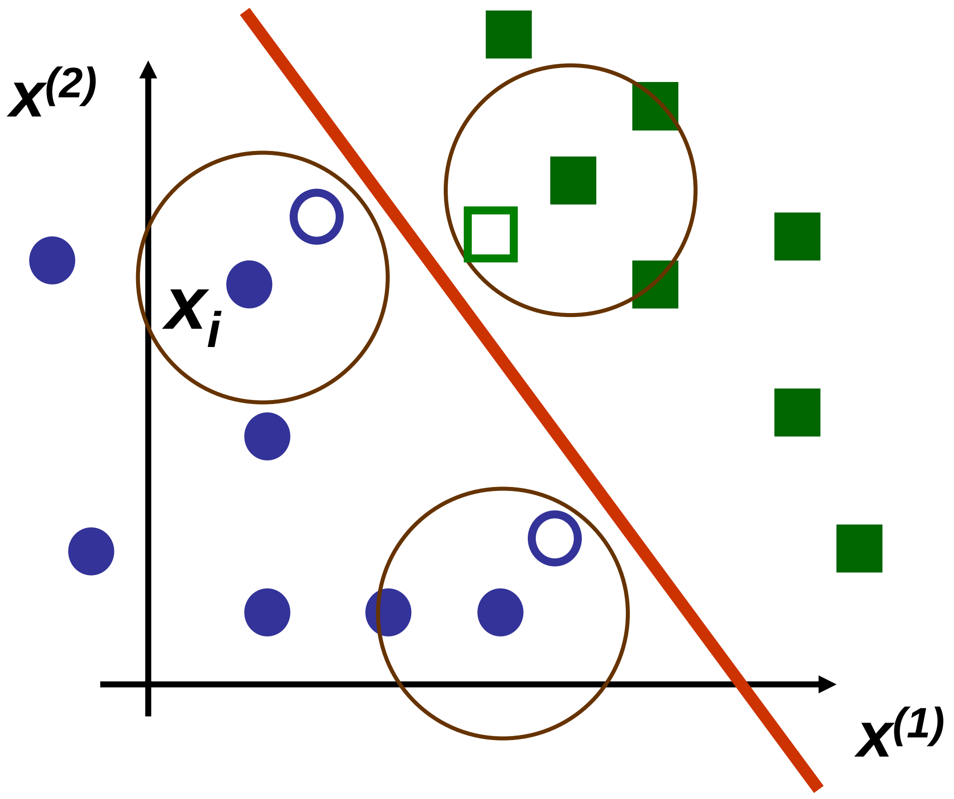
Separating Hyperplanes



- which separating hyperplane should we choose?

Separating Hyperplanes

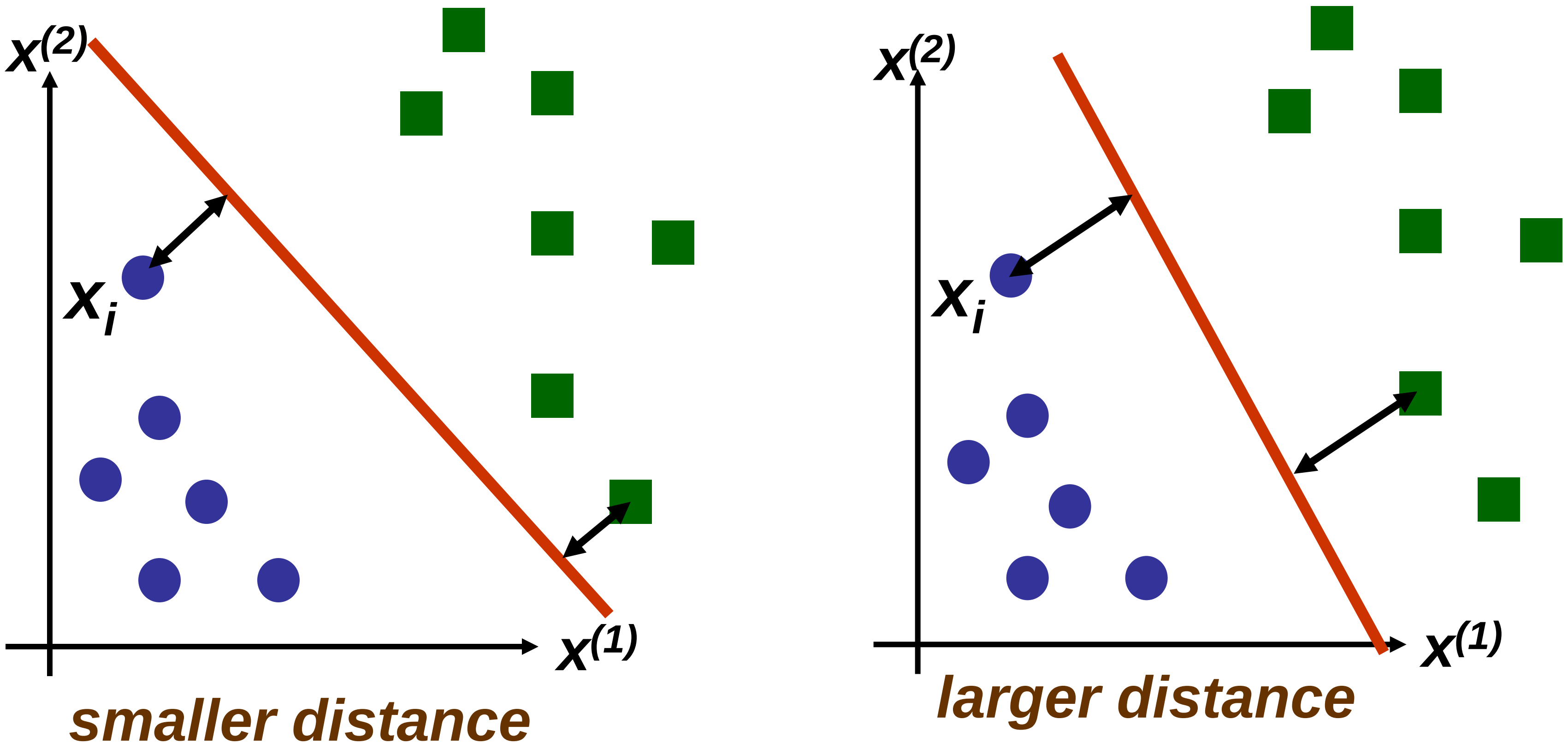
- Hyperplane as far as possible from any sample



- New samples close to the old samples will be classified correctly => Good generalization

SVM

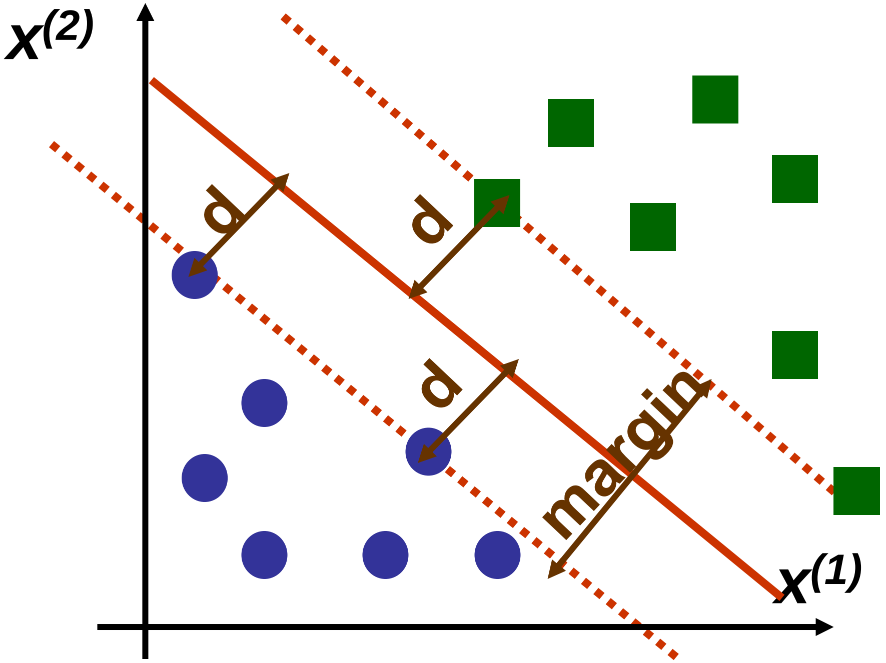
- Idea: maximize distance to the closest example



- For the optimal hyperplane
  - distance to the closest negative example = distance to the closest positive example

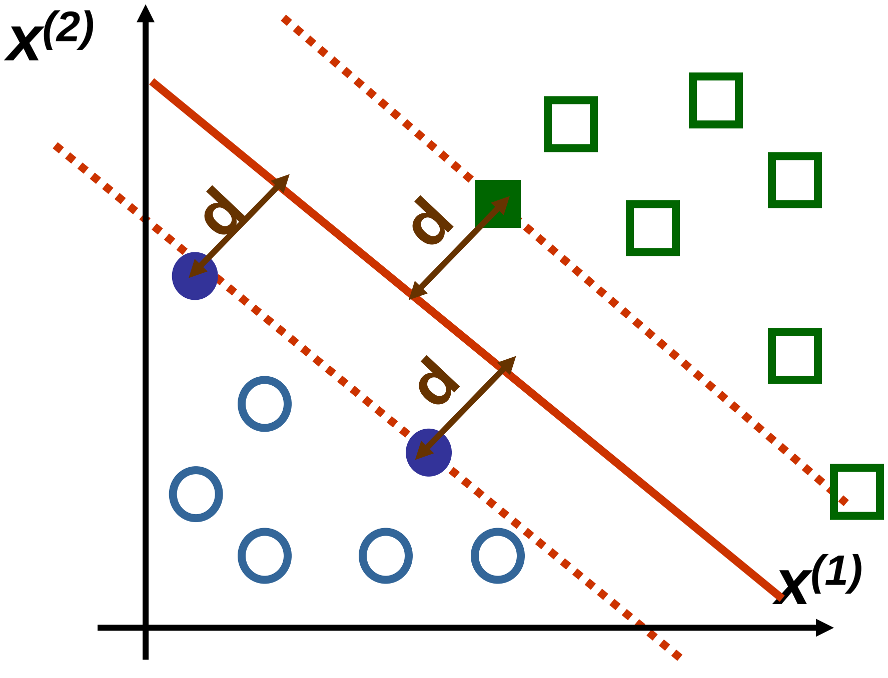
SVM: Linearly Separable Case

- SVM: maximize the ***margin***



- Better generalization in theory and practice

SVM: Linearly Separable Case



- ***Support vectors*** are the samples closest to the separating hyperplane

SVM: Optimal Hyperplane

- Maximize margin $m = \dfrac{2}{\|\mathbf{w}\|}$

  subject to constraints

$$\begin{cases} \mathbf{w}^t \mathbf{x}_i + \boldsymbol{b} \geq \mathbf{1} & \mathbf{y}_i = \mathbf{1} \\ \mathbf{w}^t \mathbf{x}_i + \boldsymbol{b} \leq -\mathbf{1} & \mathbf{y}_i = \mathbf{-1} \end{cases}$$

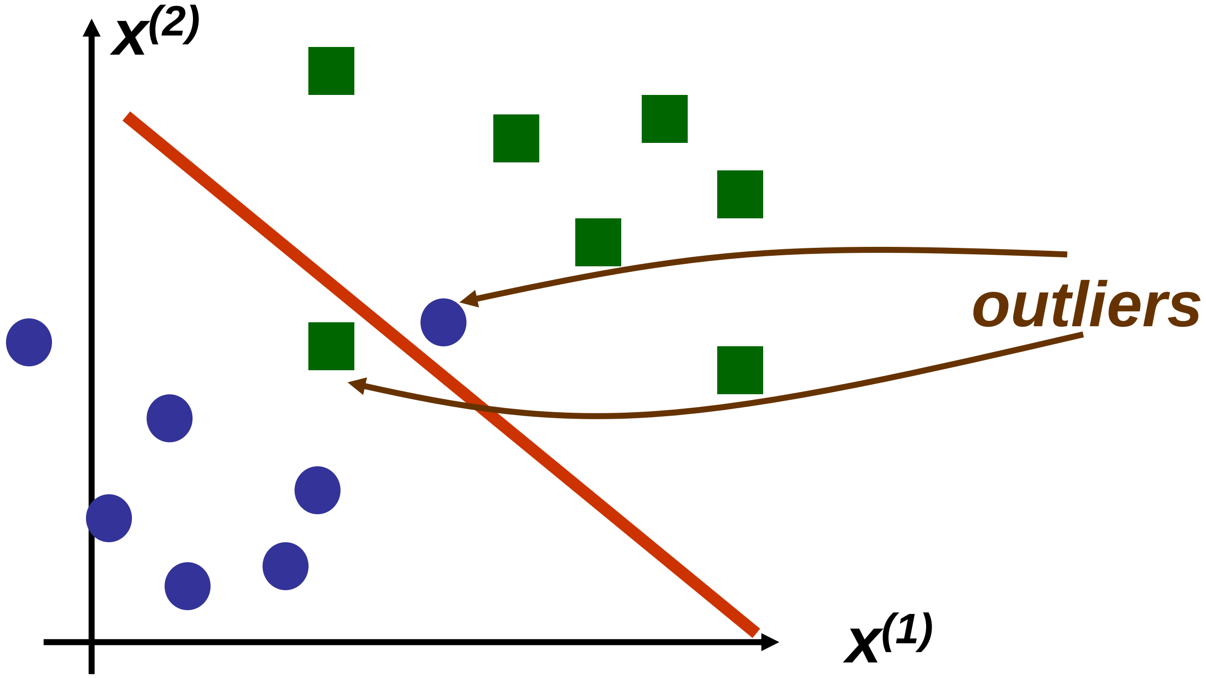
- Can convert our problem to

$$\boldsymbol{J}(\mathbf{w}) = \frac{\mathbf{1}}{\mathbf{2}} \|\mathbf{w}\|^2 \qquad \text{s.t} \quad \boldsymbol{y}_i (\mathbf{w} \cdot \mathbf{x}_i + \boldsymbol{b}) \geq \mathbf{1}$$

- ***J***(***w***) is a quadratic function, thus there is a single global minimum

SVM: Non Separable Case

- Data is most likely to be not linearly separable, but linear classifier may still be appropriate

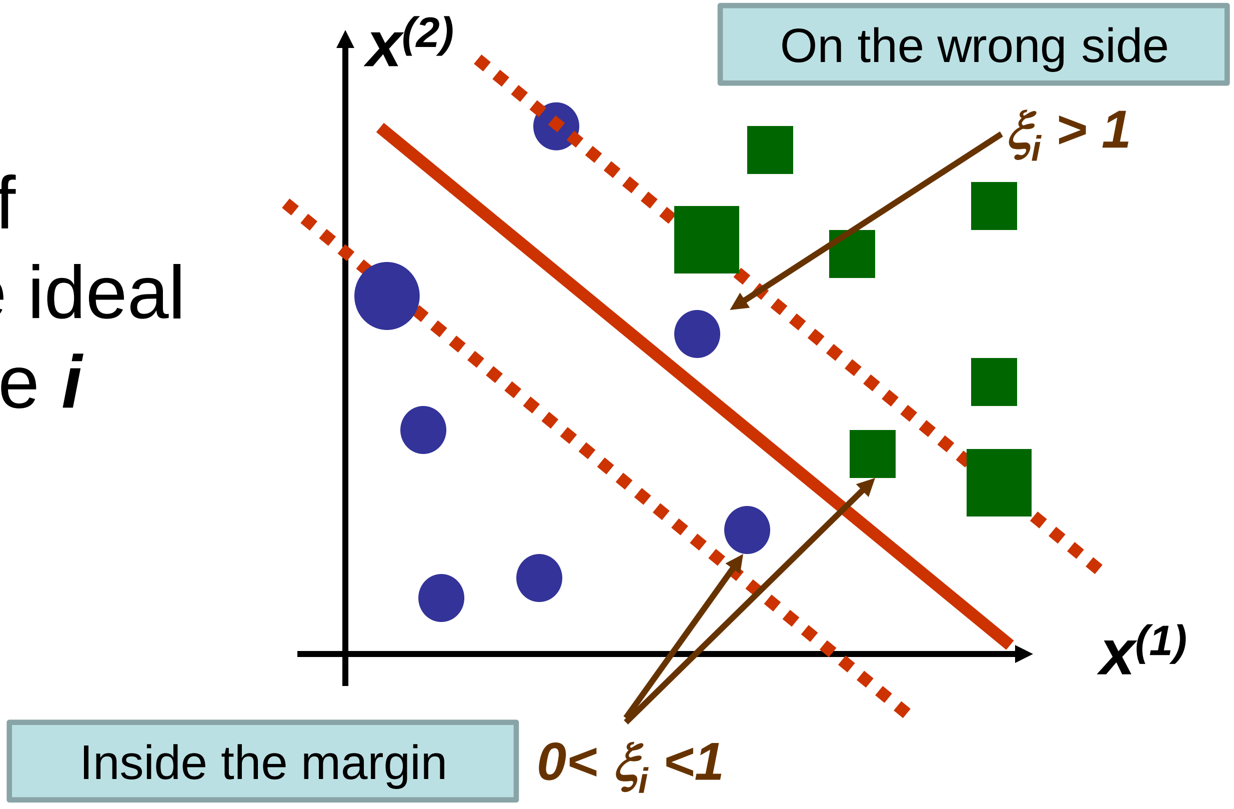


SVM with slacks

- Use nonnegative "slack" variables $\xi_1, \ldots, \xi_n$ (one for each sample)
- Change constraints from $y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 \quad \forall i$ to

$$y_i(\mathbf{w}^t\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$$

- ξ_i is a measure of deviation from the ideal position for sample i



SVM with slacks

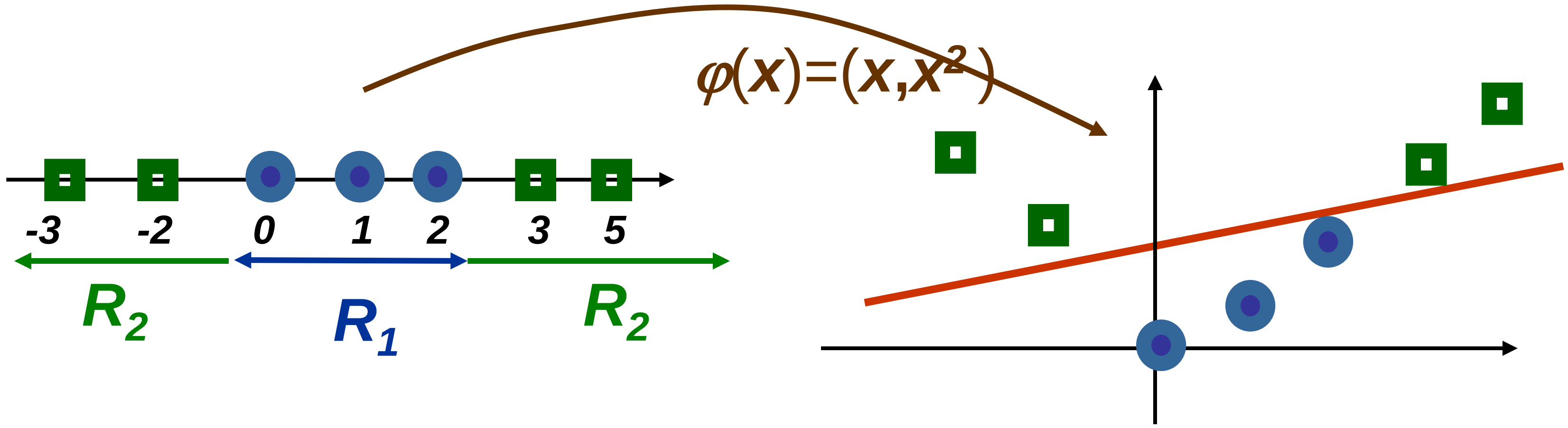
minimize $\quad J(w, \xi_1, \ldots, \xi_n) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$

constrained to $\; y_i\left(\mathbf{w}^t\mathbf{x}_i + b\right) \geq 1 - \xi_i \;$ and $\; \xi_i \geq 0 \quad \forall i$

- **C** $> \mathbf{0}$ is a constant which measures relative weight of the first and second terms
  - if C is small, we allow a lot of samples not in ideal position
  - if *C* is large, we want to have very few samples not in ideal position

Non Linear Mapping

- Solve a non linear classification problem with a linear classifier
  1. Project data $\boldsymbol{x}$ to high dimension using function $\varphi(\boldsymbol{x})$
  2. Find a linear discriminant function for transformed data $\varphi(\boldsymbol{x})$
  3. Final nonlinear discriminant function is $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{w}^t \varphi(\boldsymbol{x}) + \boldsymbol{w}_0$



- In 2D, discriminant function is linear

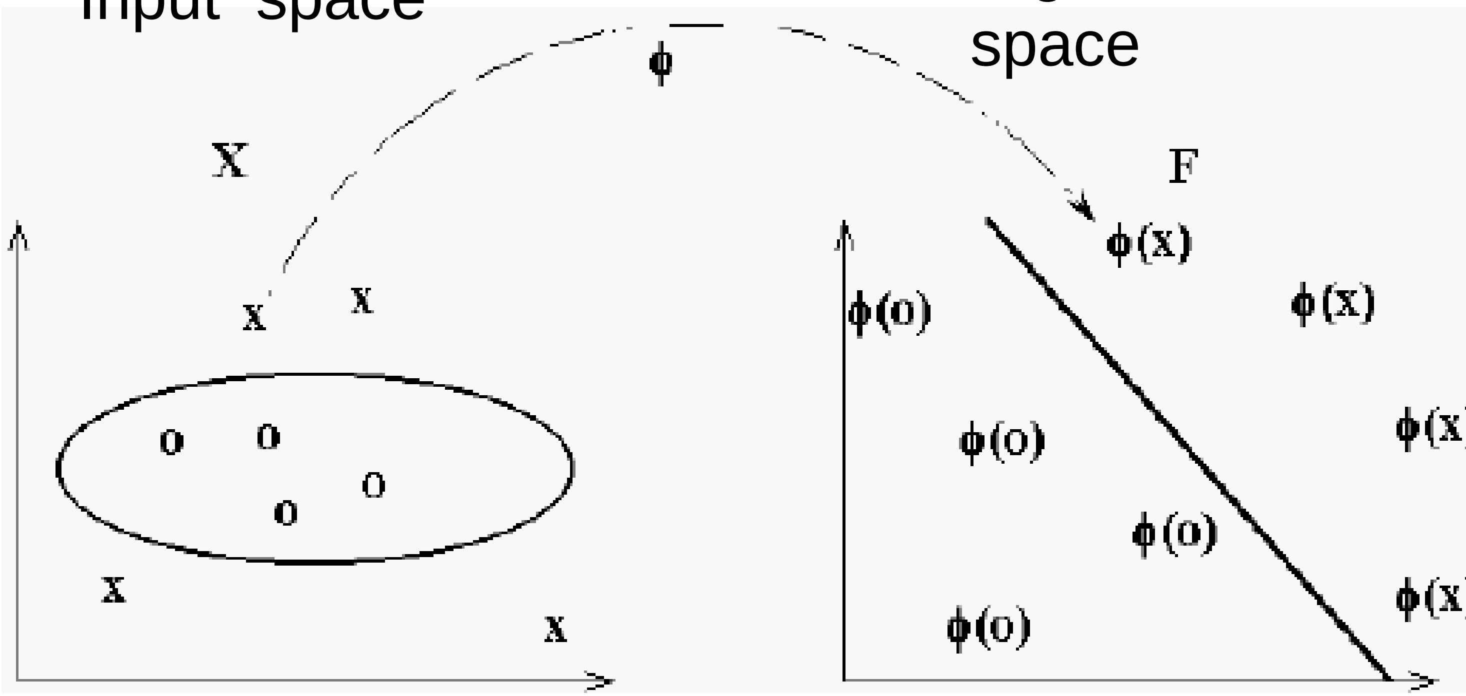
$$g\left(\begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix}\right) = \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ x^{(2)} \end{bmatrix} + w_0$$

- In 1D, discriminant function is not linear $\quad g(x) = w_1 x + w_2 x^2 + w_0$

Kernel Trick

Input space

Higher dimensional space



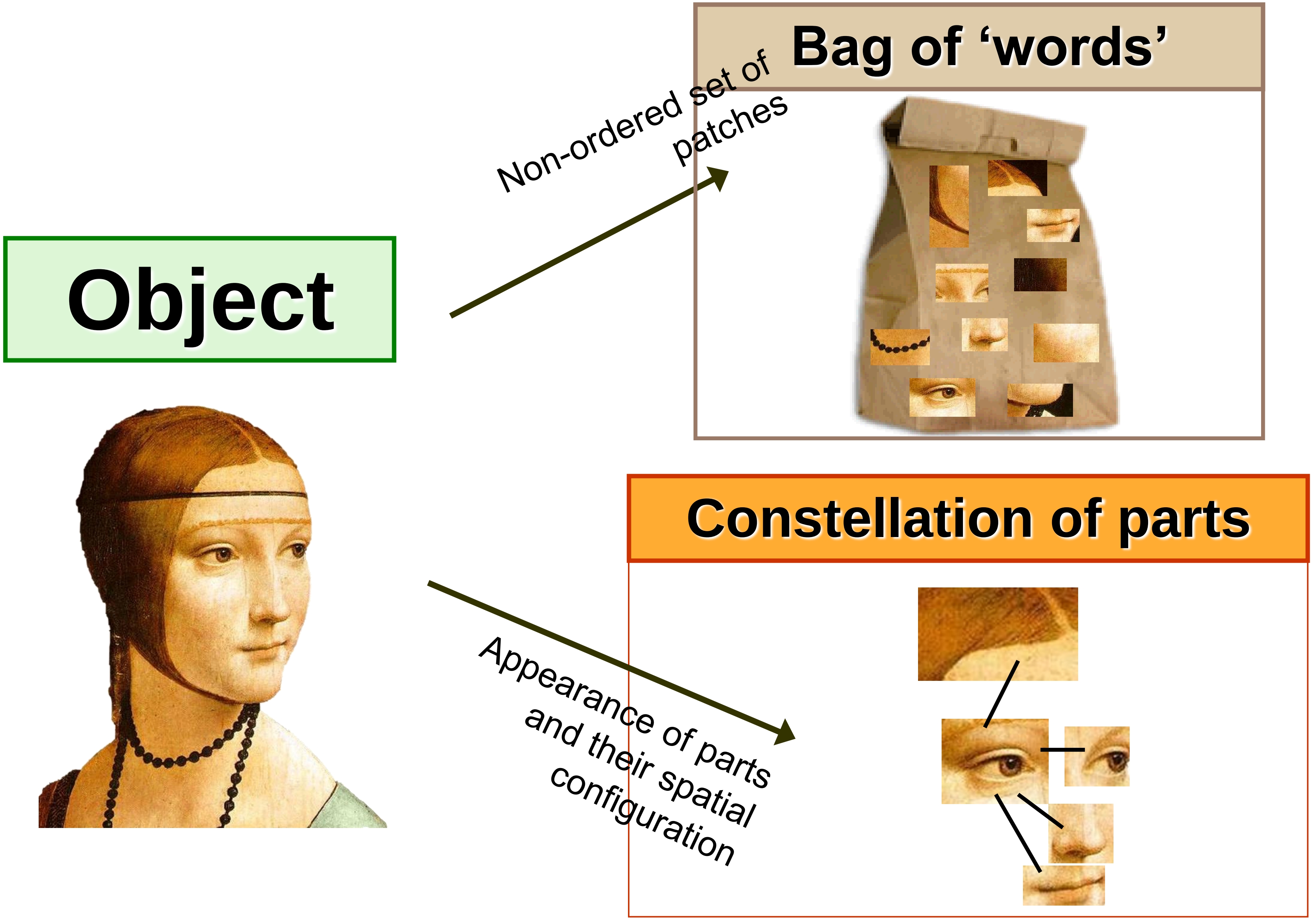
$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_1) \rangle$$

$$x_1, x_2 \in R^n \qquad \phi(x_1), \phi(x_2) \in F$$

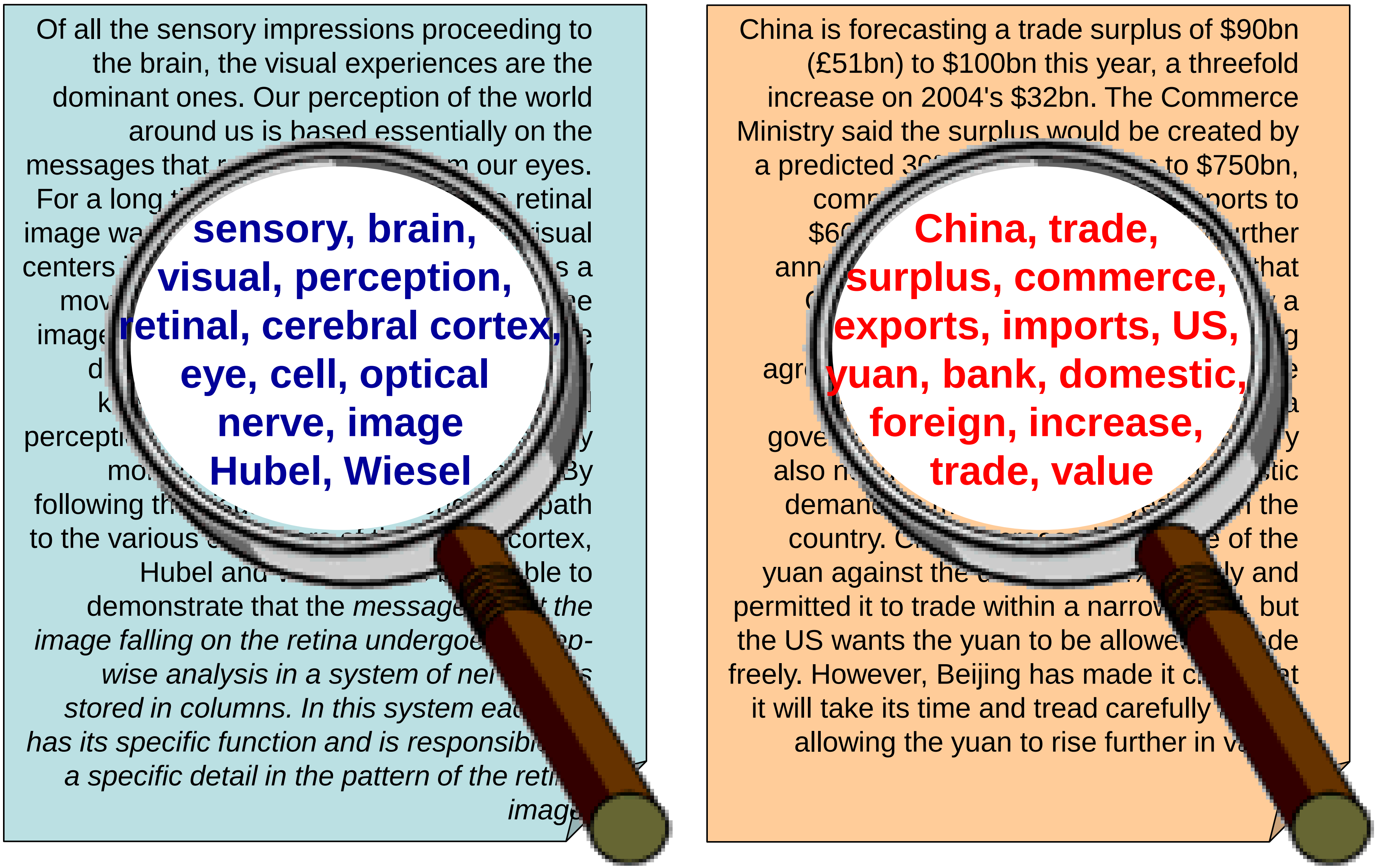
The Kernel Trick

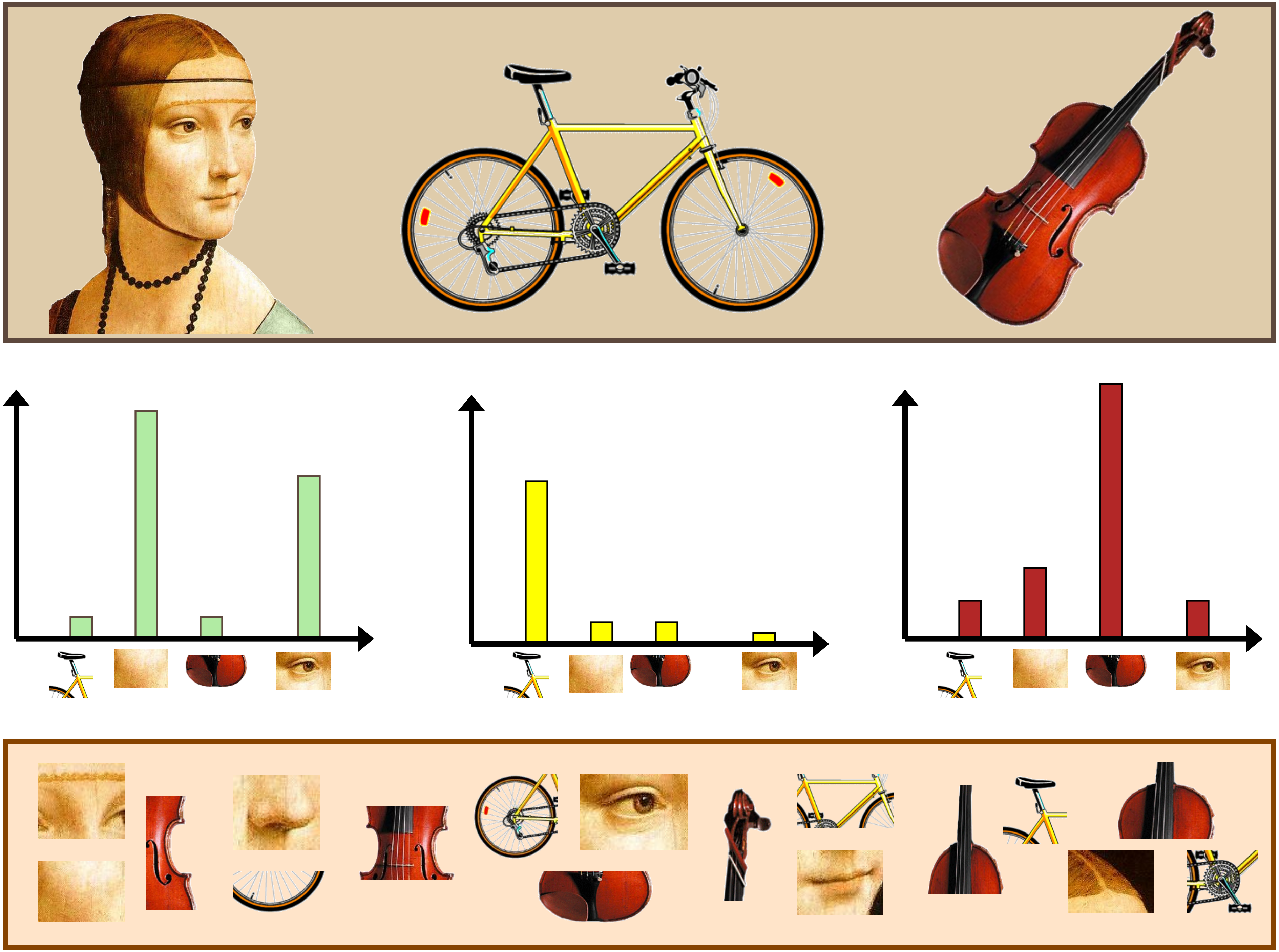
- Choose kernel $\boldsymbol{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ corresponding to some function $\varphi(\boldsymbol{x}_i)$ which takes sample $\boldsymbol{x}_i$ to a higher dimensional space (don't need to know $\varphi(\boldsymbol{x}_i)$)
- Replace dot products in the SVM formulation with kernel values.
- Need to compute the kernel matrix for the training data
- Need to compute $\boldsymbol{K(x_i, x)}$ for all SVs $\boldsymbol{x}_i$

Part-Based Approaches



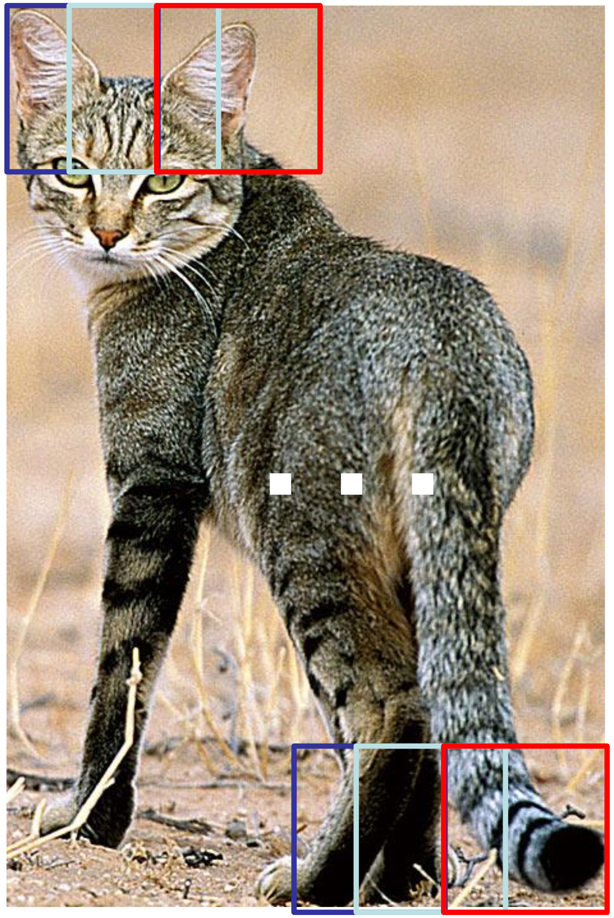
Bag of ‘words’ analogy to documents



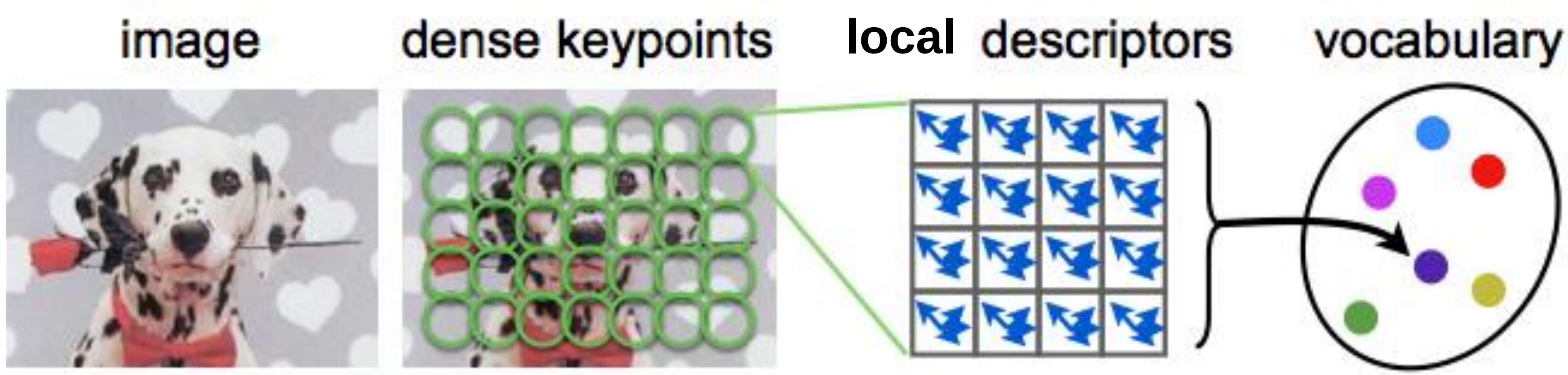


Regular dense grid over space and scale

Dense keypoints over space and scale



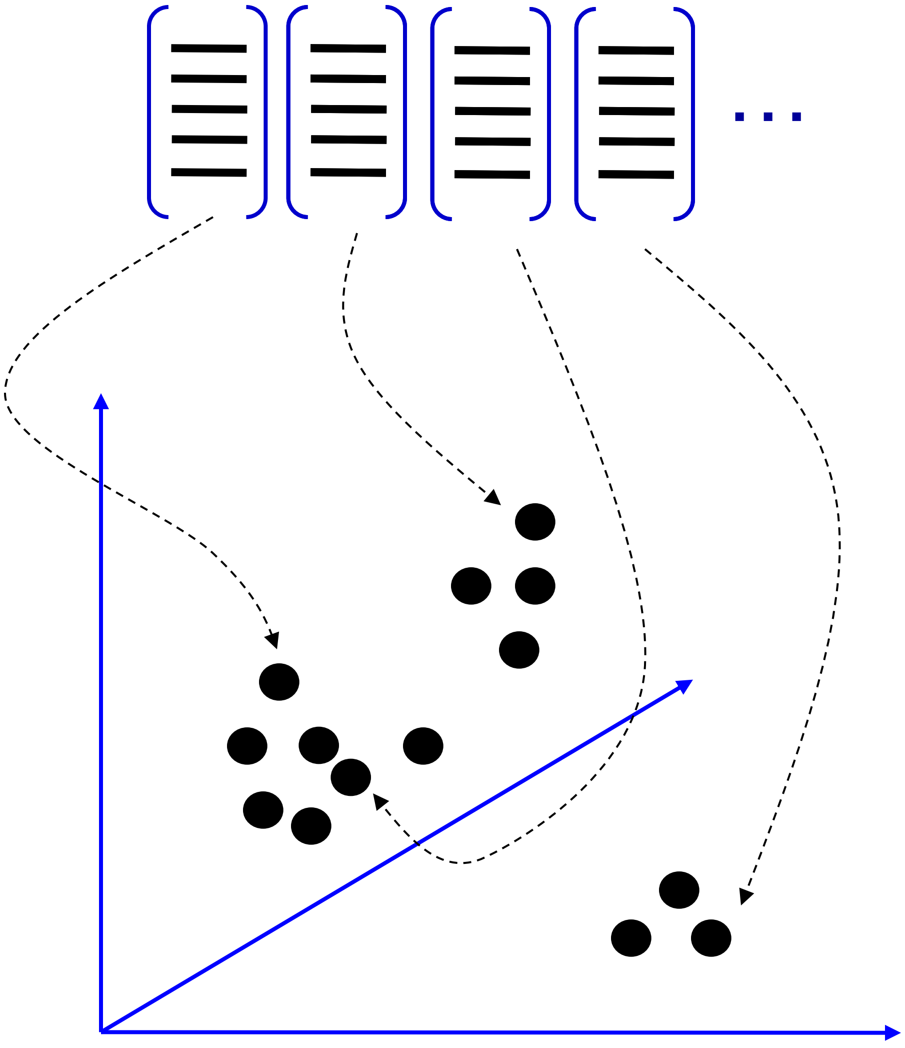
Representation Scheme



Slide credit: A. Vedaldi

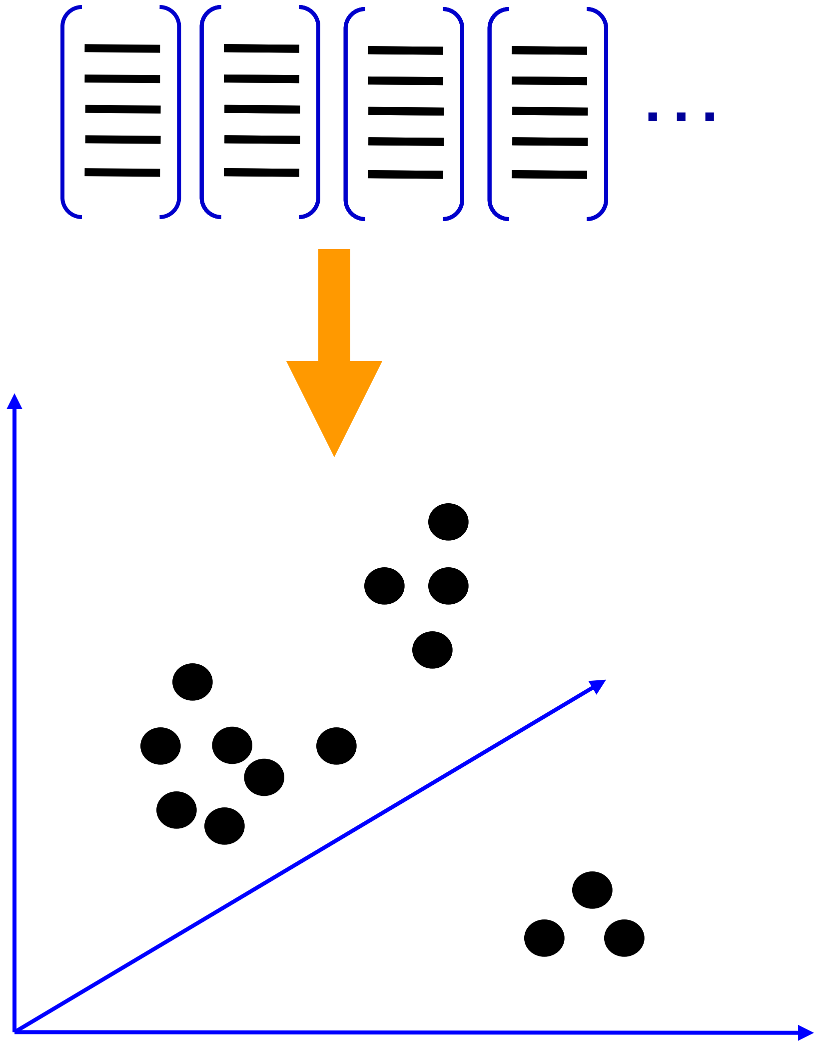
Vocabulary formation

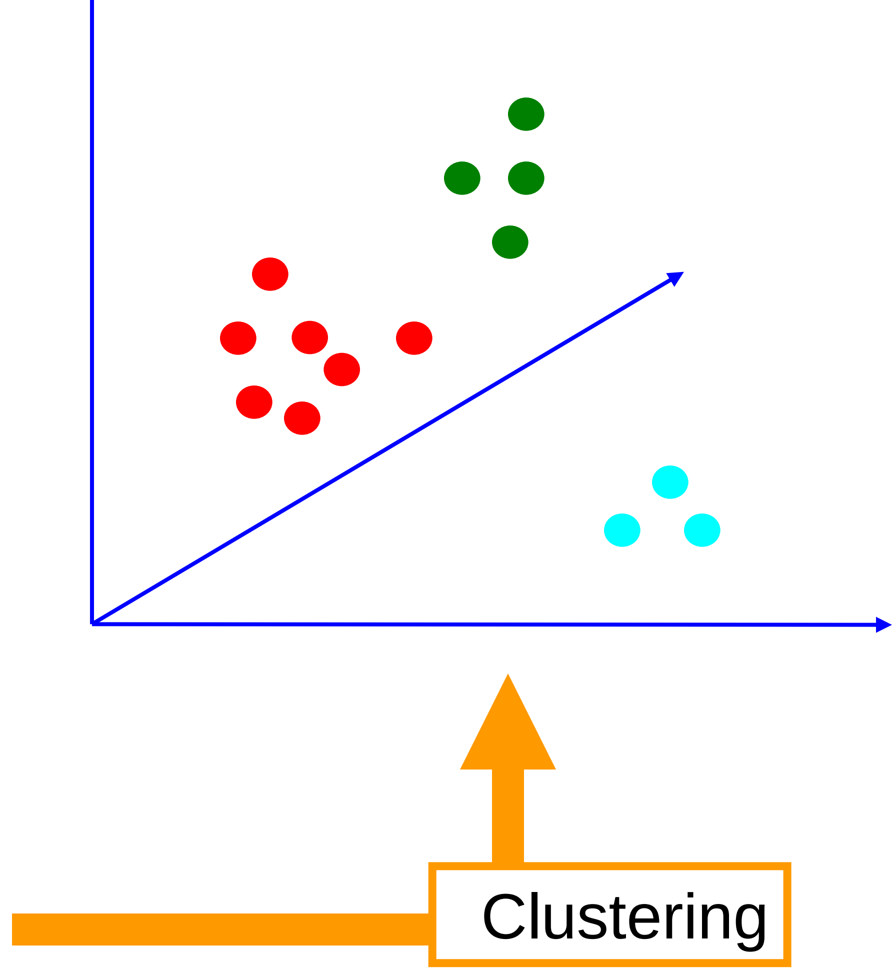
Local descriptors from all (subset of) images



Vocabulary formation

Local descriptors from all (subset of) images





Slide credit: Josef Sivic

Clustering

- Input:
  - Training samples $x_1,\ldots,x_n \in R^n$
  - No labels are given.
- Goal: group input samples into classes of similar objects – cohesive "clusters."
  - high intra-class similarity
  - low inter-class similarity
- Algorithms: many, most common k-means

k-means, definitions

- k – the number of clusters

    a parameter of the algorithm

- μ_i cluster centroids

    represent our current guesses for the positions of the centers of the clusters

- Initialization: pick k random training samples.

    Other initialization methods are also possible.

k-means, algorithm

1. Initialize cluster centroids: $\mu_1, \ldots, \mu_k \in R^n$
2. Repeat until convergence:

   For every i, set

   $$c_i = \arg\min_j \left\| x_i - \mu_j \right\|^2$$

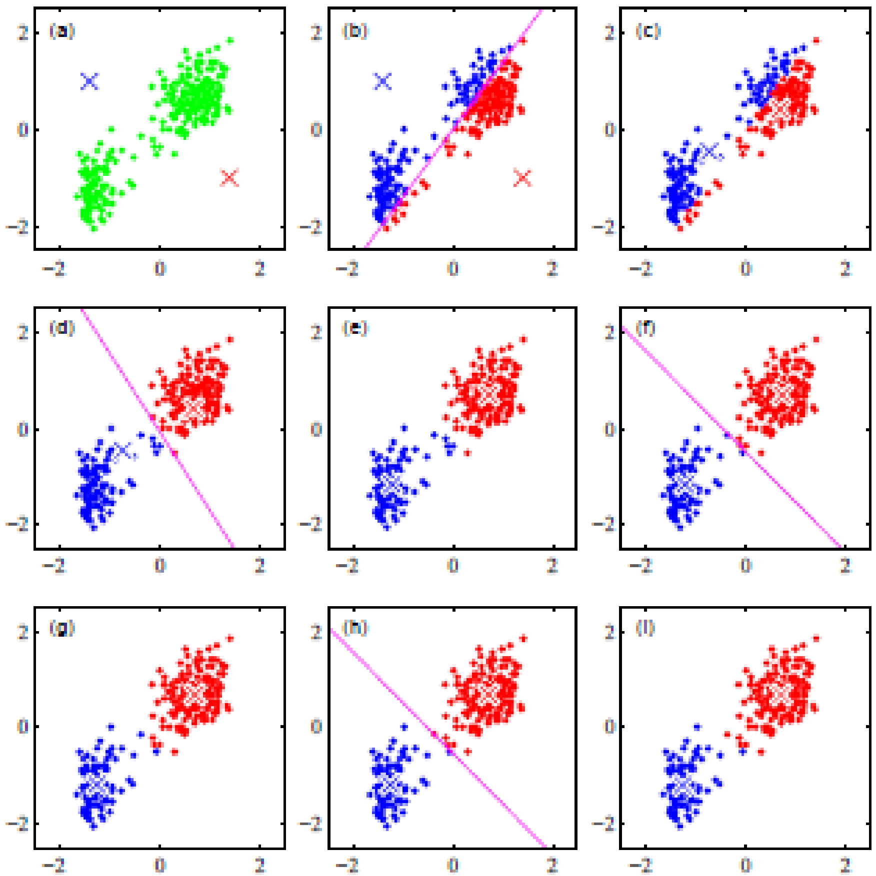
   Assign point to the closest cluster centroid

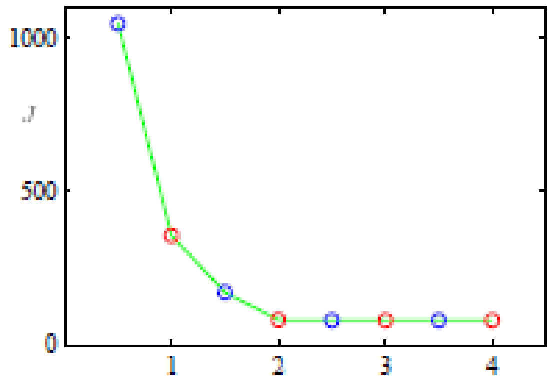
   For each j, set

   $$\mu_j = \frac{\sum_{i=1}^{n} 1\{c_i = j\} x_i}{\sum_{i=1}^{n} 1\{c_i = j\}}$$

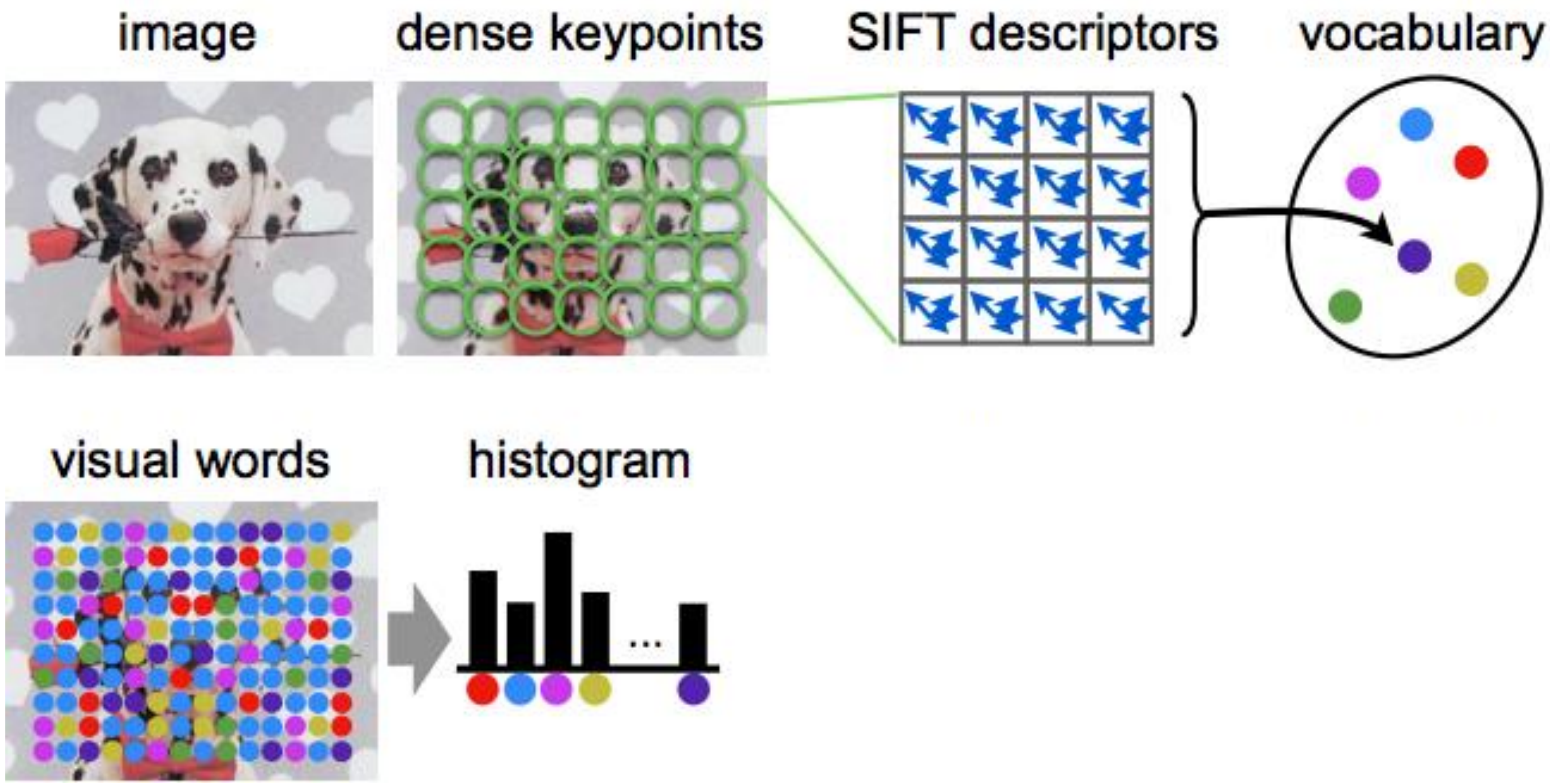
   Update centroids to be the mean of the points assigned to it

K-means, example



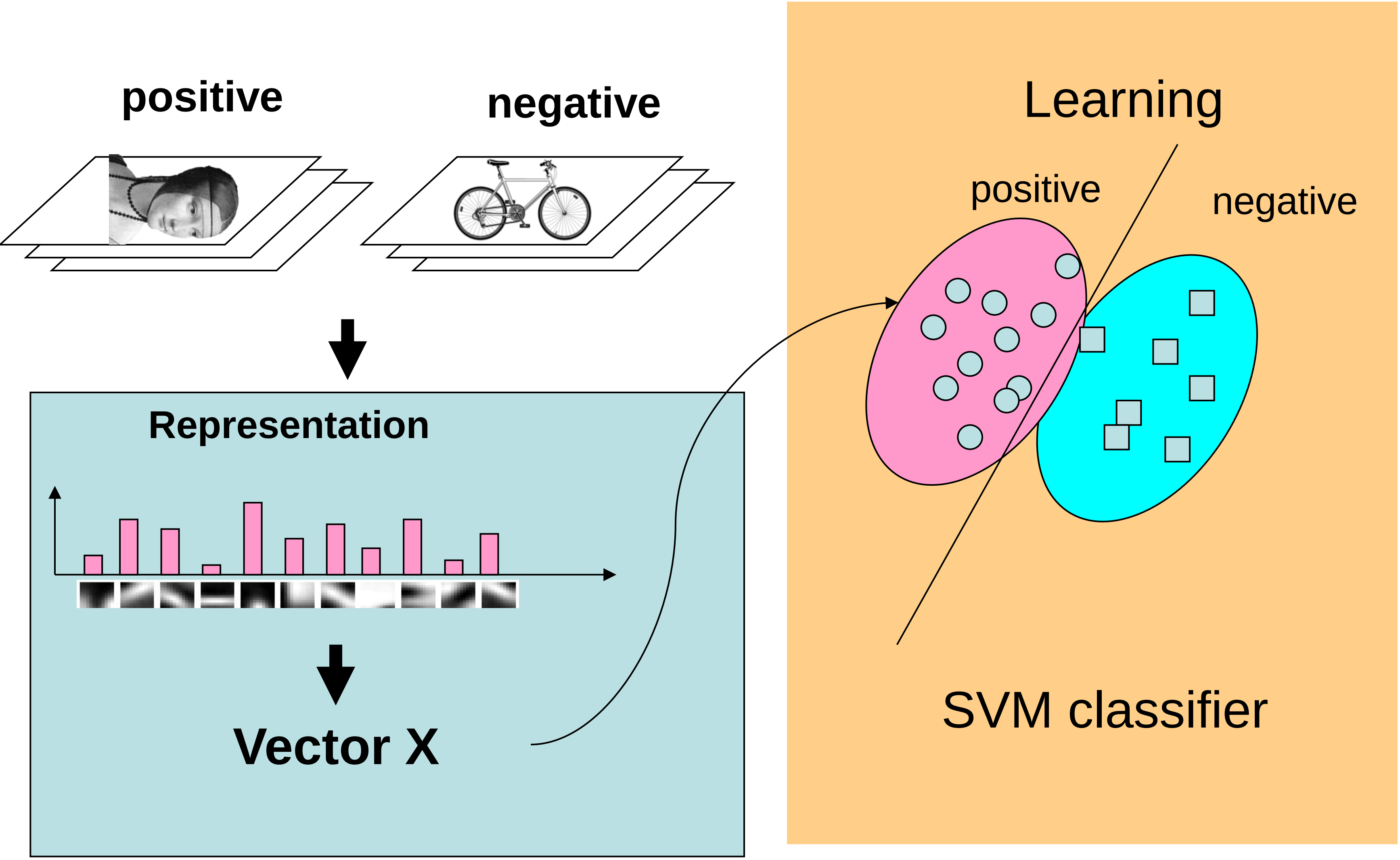


Representation Scheme

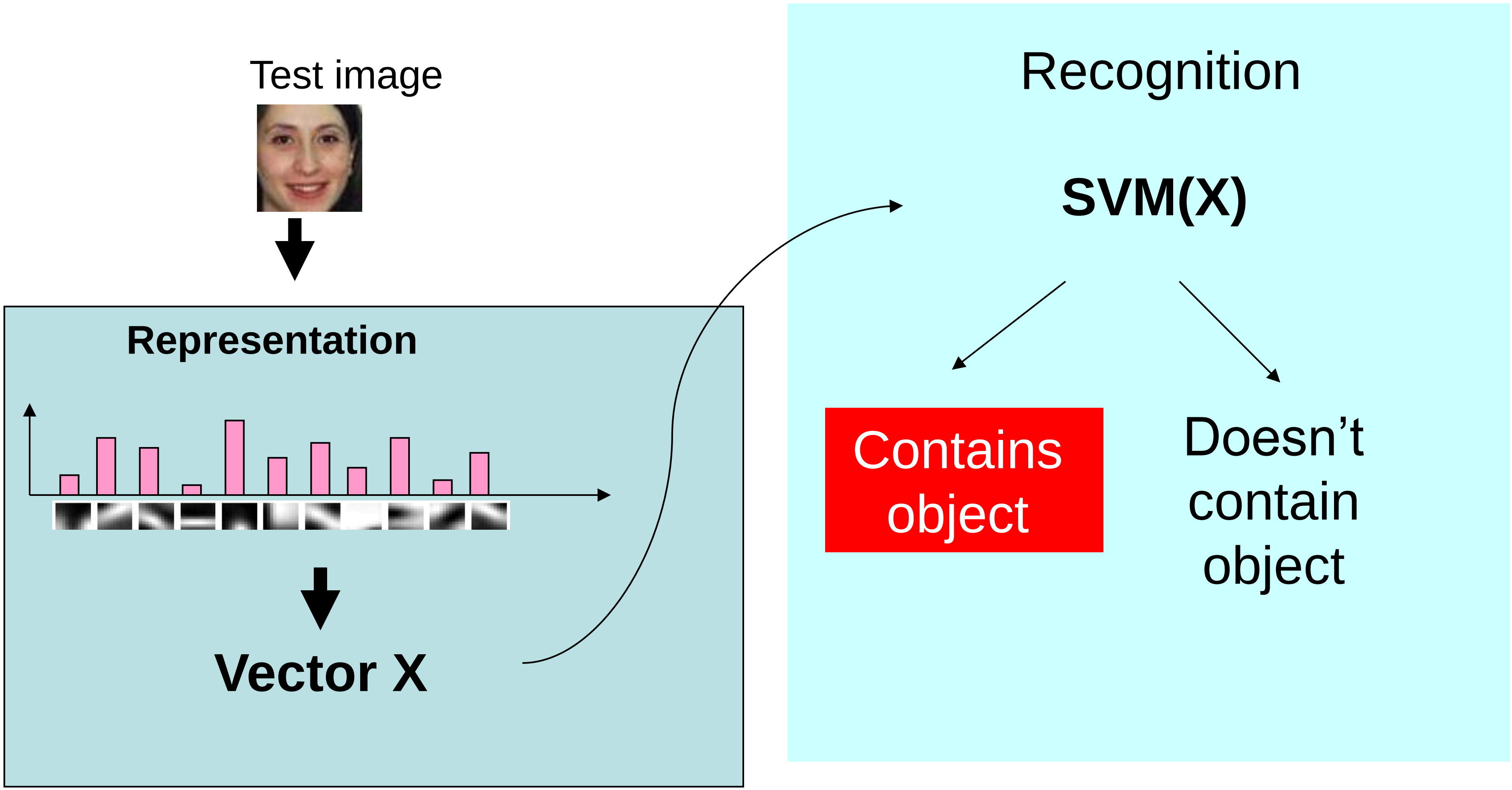


Slide credit: A. Vedaldi

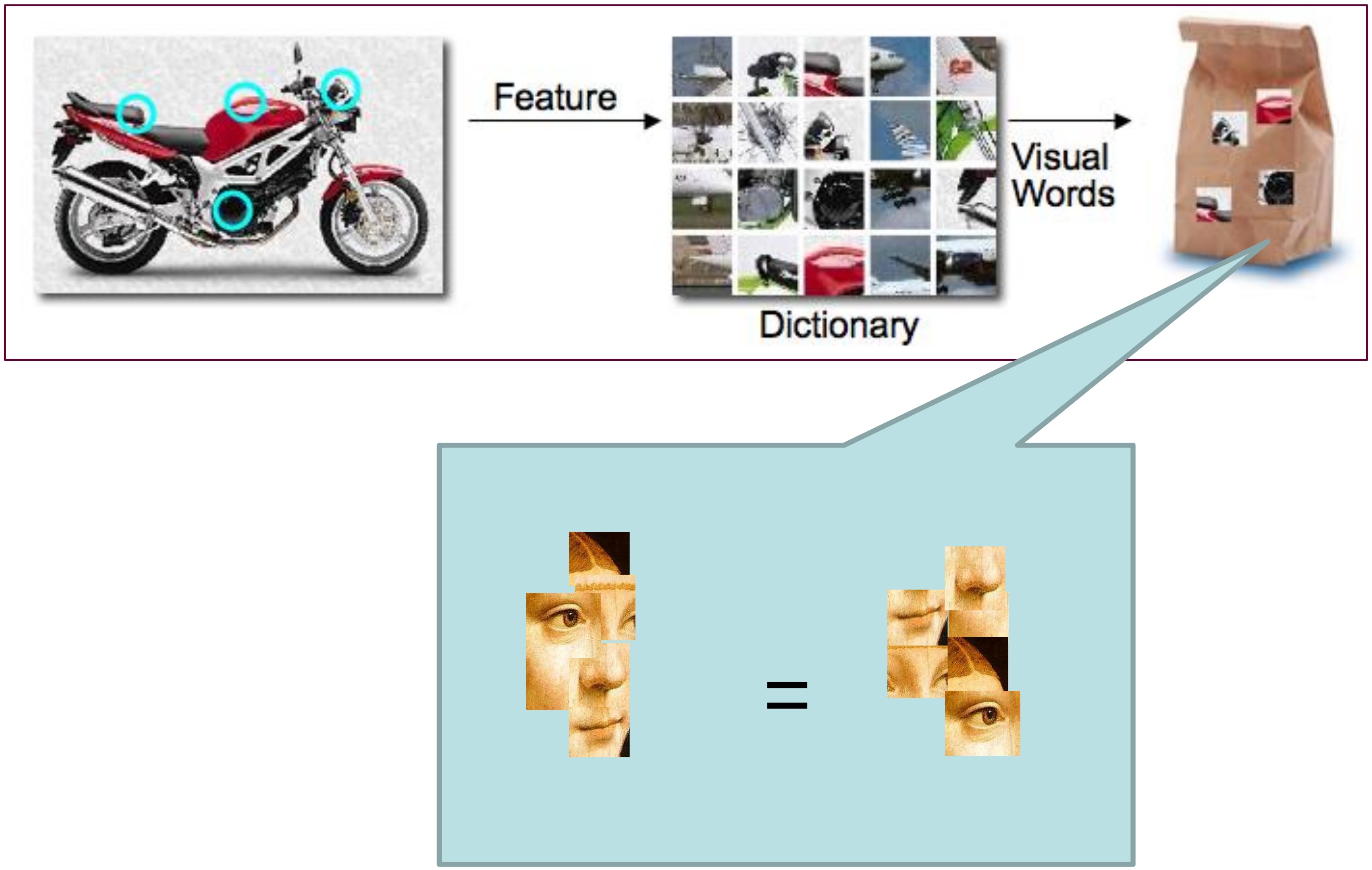
SVM classification



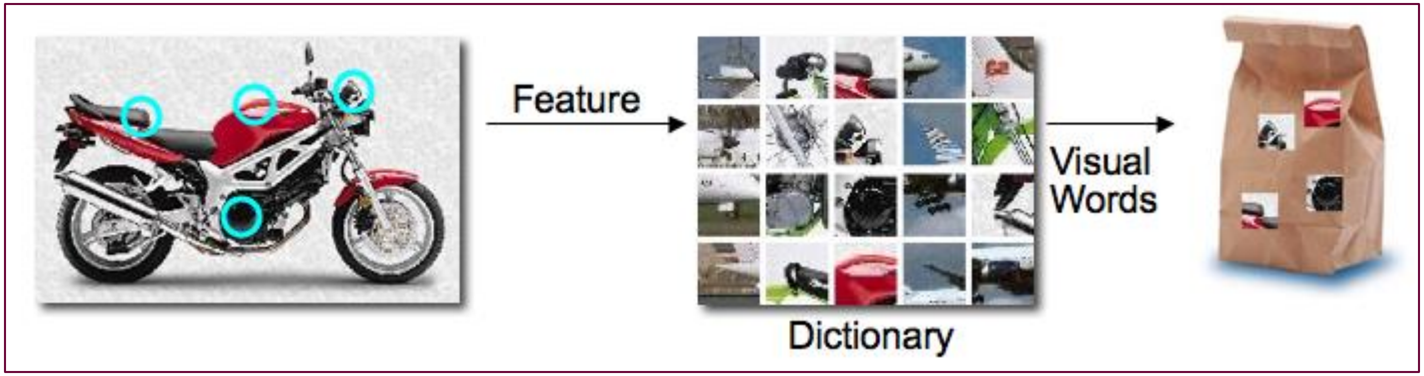
SVM classification



Bag of Features



Bag of Features



Pros: fairly flexible and computationally efficient

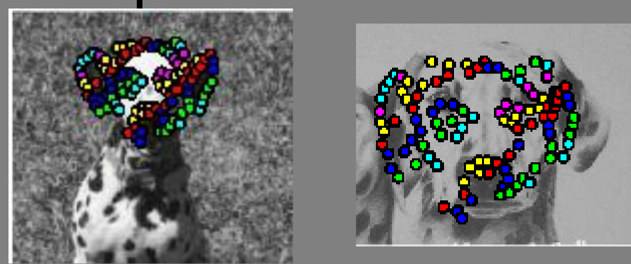
Cons: problems with large clutter

Different objects, but Similar representations;

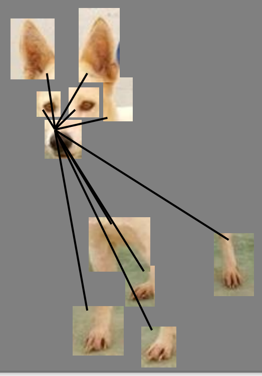
Similar objects, different representations;

Fusion of appearance and shape

Geometric correspondence search



Constellation model

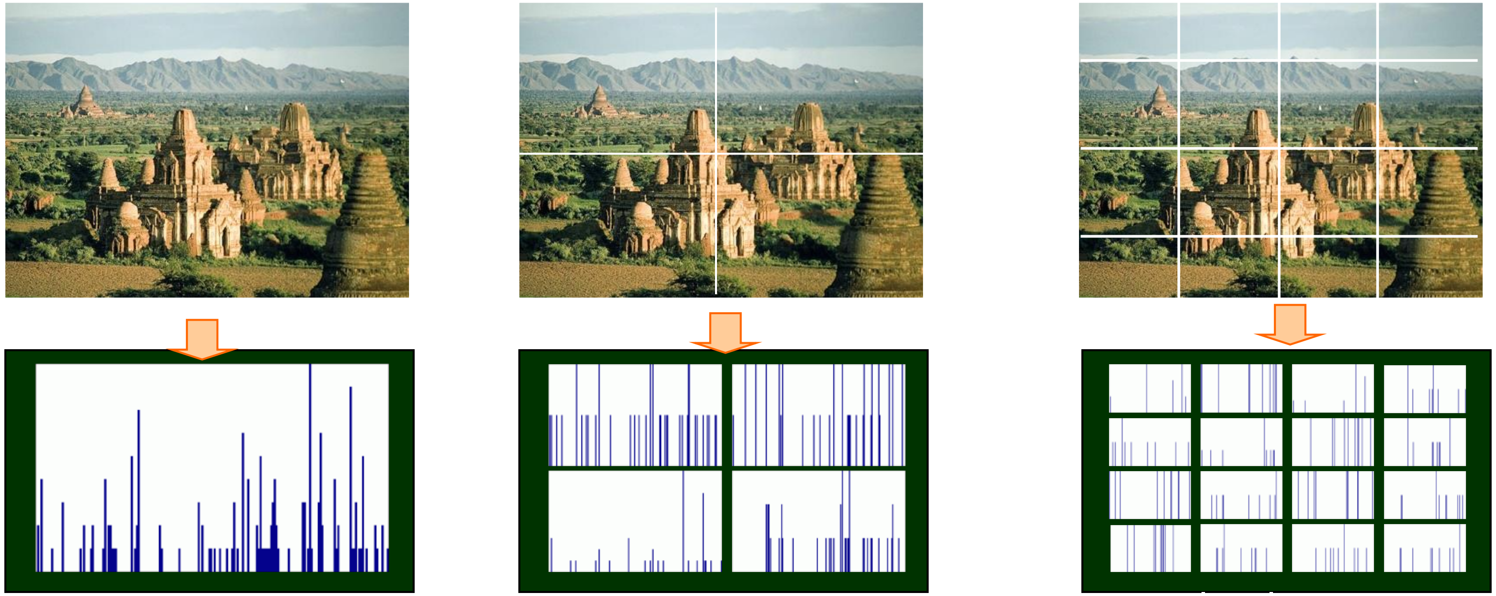


Pros: Structural information is important for recognition.

Cons: computationally expensive, restricted to small variation in shape within the category.

Beyond Bags of Features

- Computing bags of features on sub-windows of the whole image.



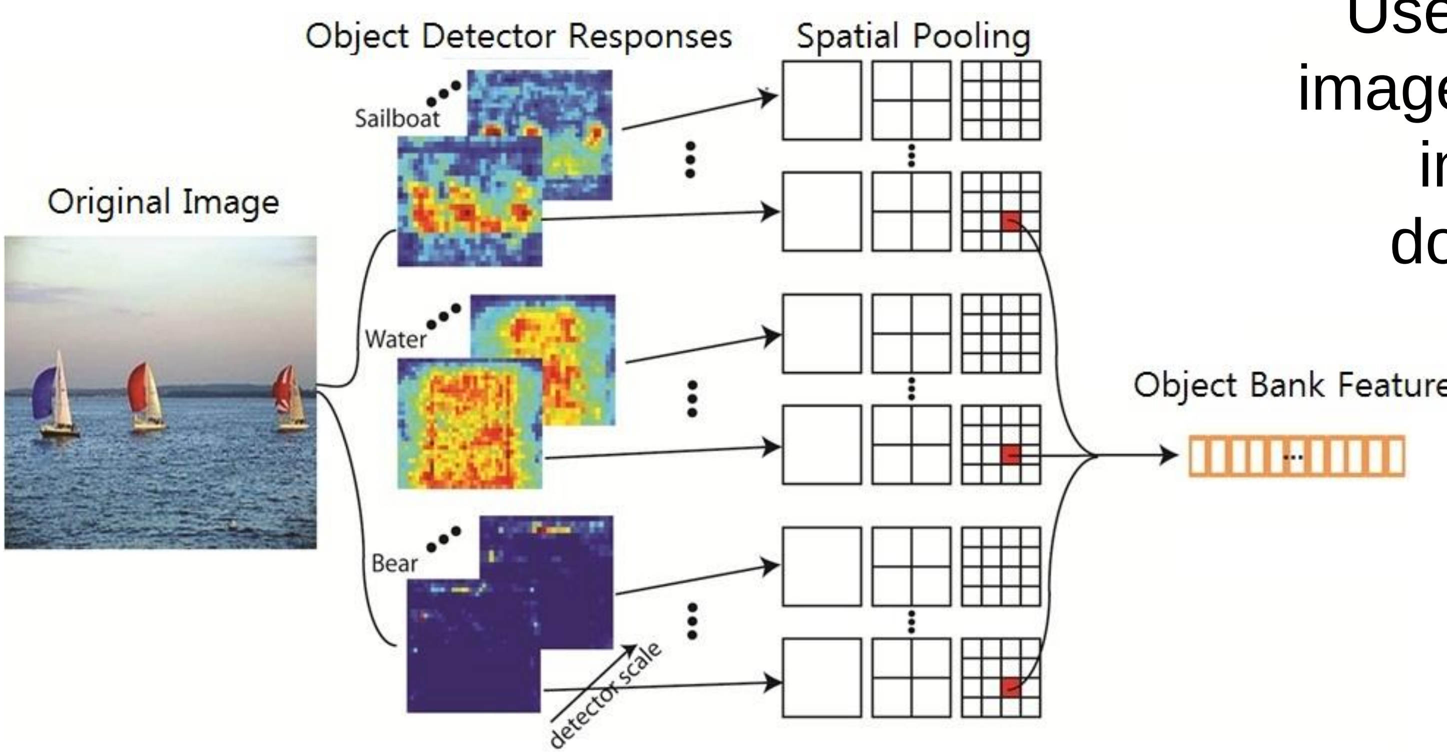
Lazebnik, Schmid & Ponce (CVPR 2006)

Context and Scenes

- What is behind the red rectangle?

Context is correlated with the object

Semantically Meaningful Feature

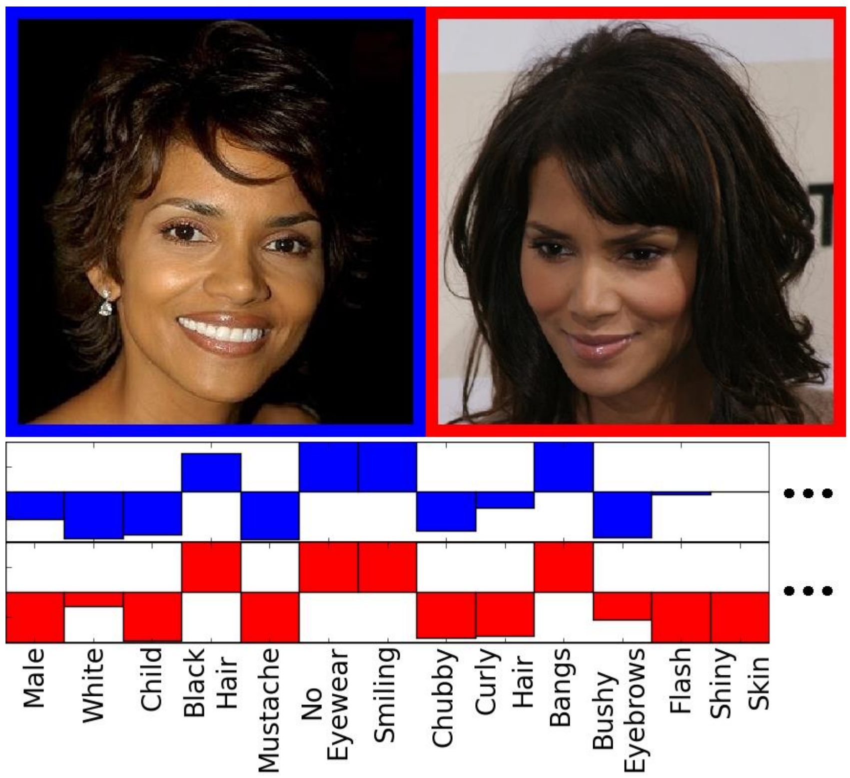


Use objects in images, as words in textual documents

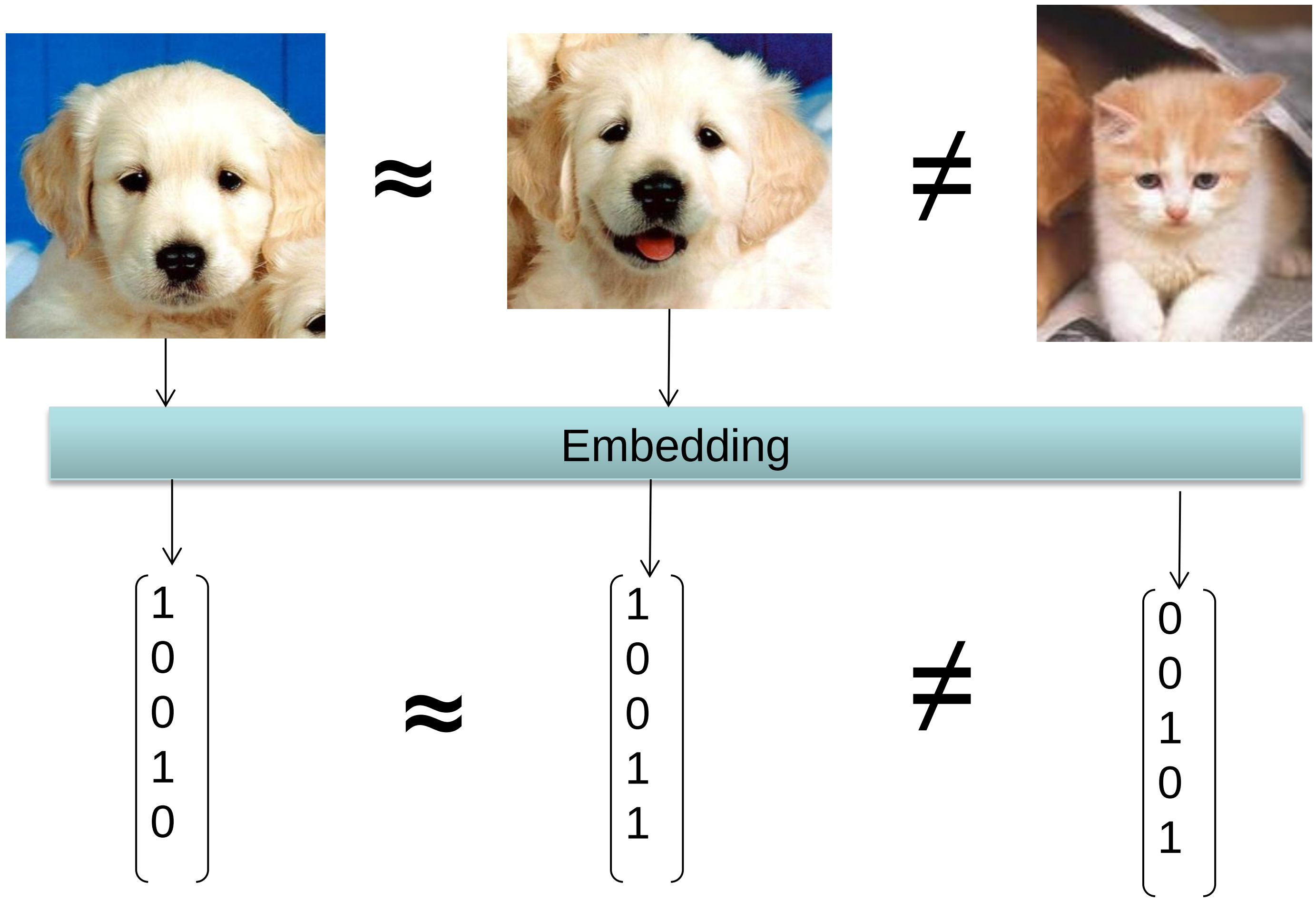
Li-Jia Li, Hao Su, Yongwhan Lim, Robert Cosgriff, Daniel Goodwin, and Li Fei-Fei Vision Lab, Stanford University

Describing Objects with Attributes

- Shift the goal of recognition from naming to describing



Large-scale image/object search



Large-scale image/object search

General form of linear projection-based hashing function:

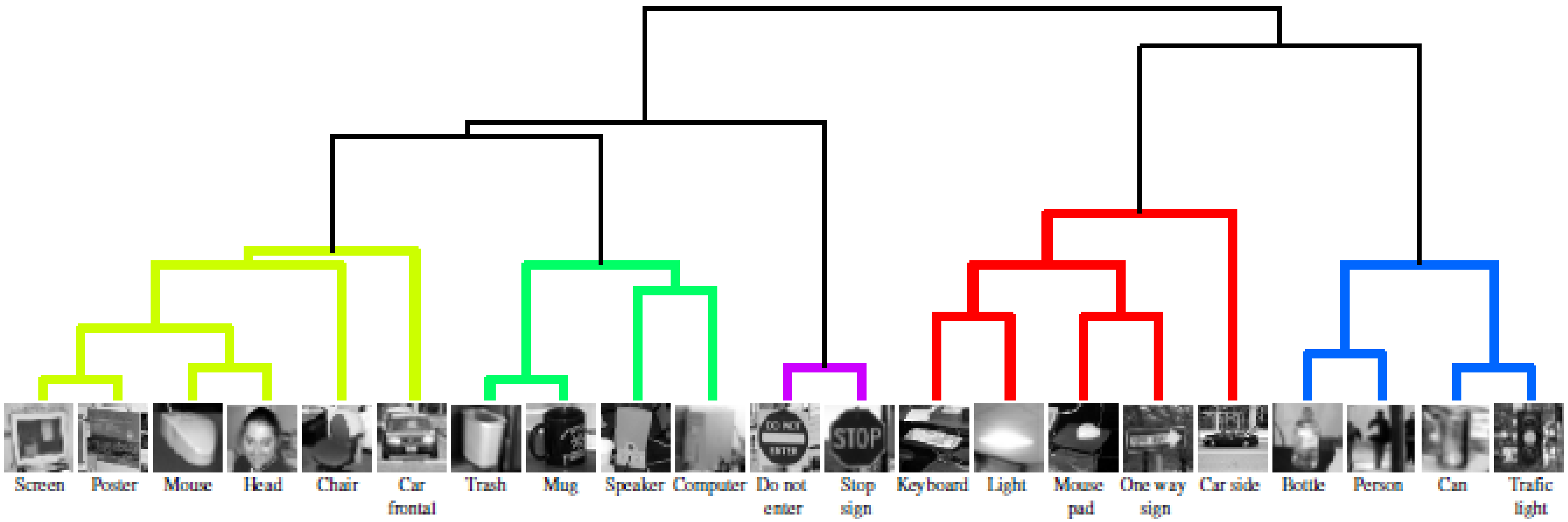
$$h_i(I) = sign\left(f\left(w_i^T I - b_i\right)\right)$$

Learn embedding $W = [w_1 \ldots w_k]$, such that:
If I and J are similar (for example $\|I - J\|^2 < \varepsilon$), then

$$d_{\text{Hamming}}(I, J) < k$$

Dealing with many categories

Sharing Features between classes



Dealing with many categories

Model Transfer

Learn Model on

...

Adjust Model to

Activity recognition