Computer Vision I

Categorization – 12.06.2013

TECHNISCHE UNIVERSITÄT DARMSTADT

vi visual inference

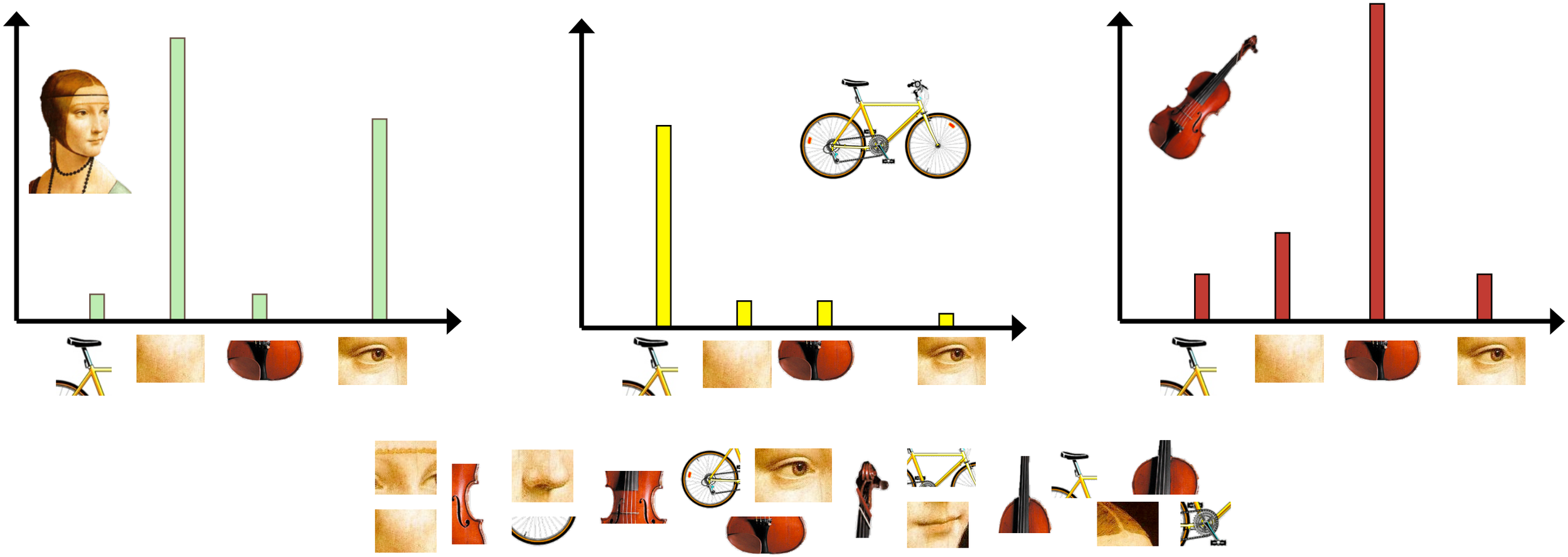
Bag-of-Words Model (BoW)

Object → **Bag of 'words'**

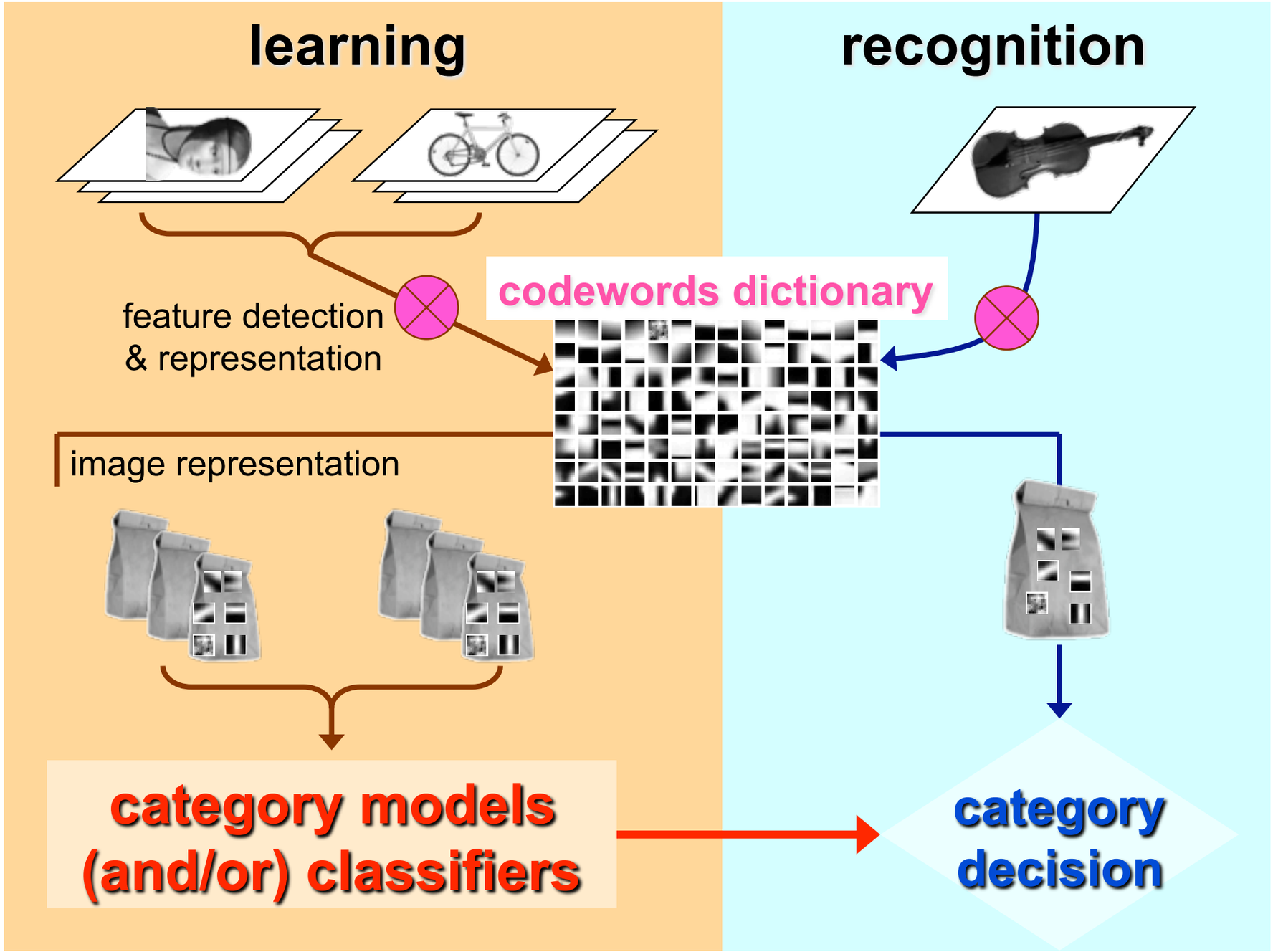
Bag of Words

TECHNISCHE UNIVERSITÄT DARMSTADT

- Independent features
- Histogram representation

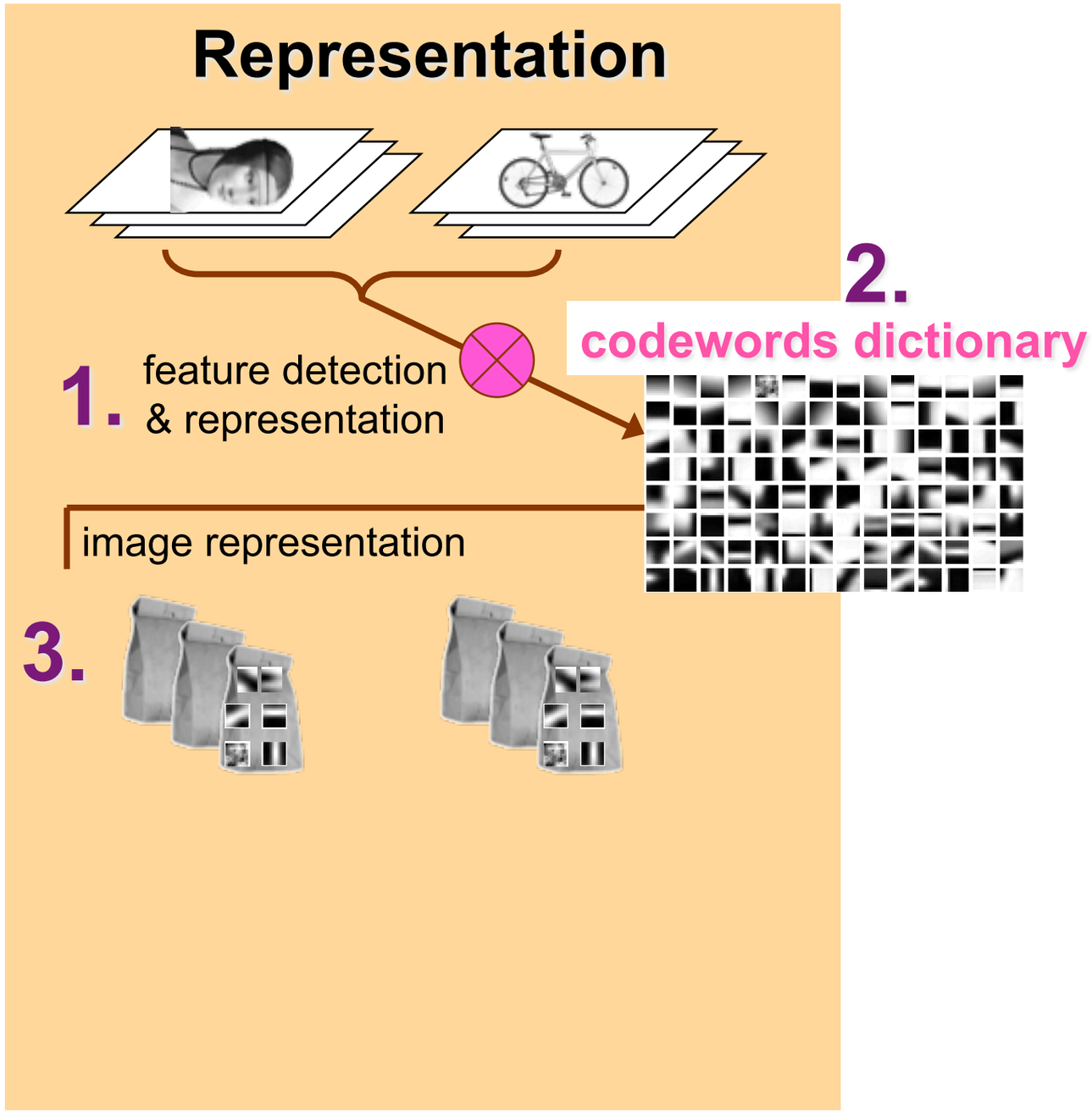


Bag-of-Words Model: Overview



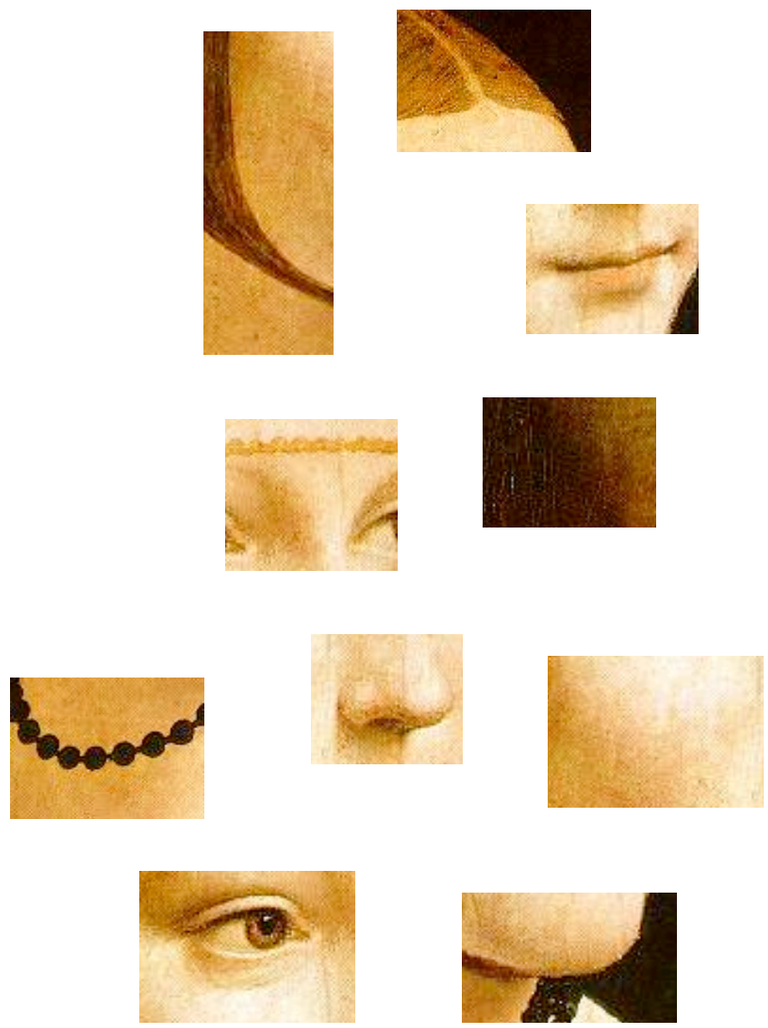
Bag-of-Words Model: Object Representation & Learning

TECHNISCHE UNIVERSITÄT DARMSTADT

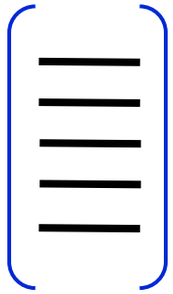


BoW-1. Feature detection and representation

TECHNISCHE UNIVERSITÄT DARMSTADT

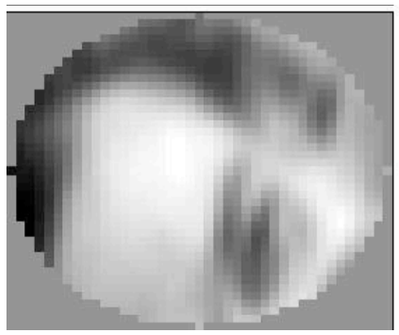


BoW-1. Feature detection and representation

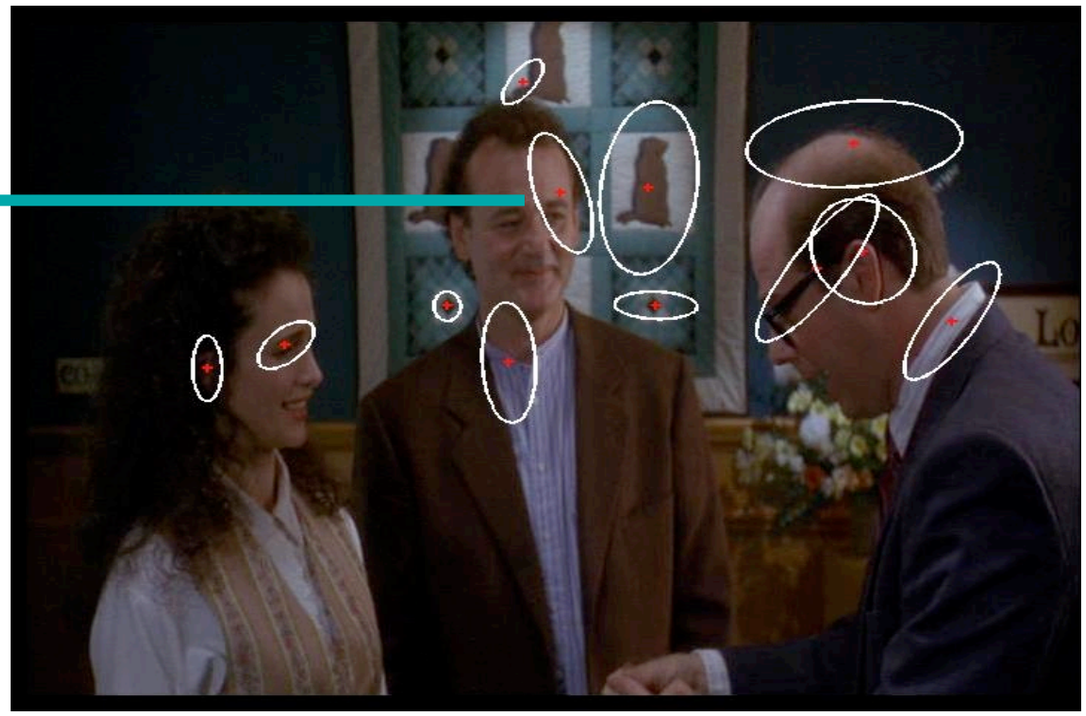


Compute descriptor

e.g. SIFT, shape context, etc.



Normalize patch

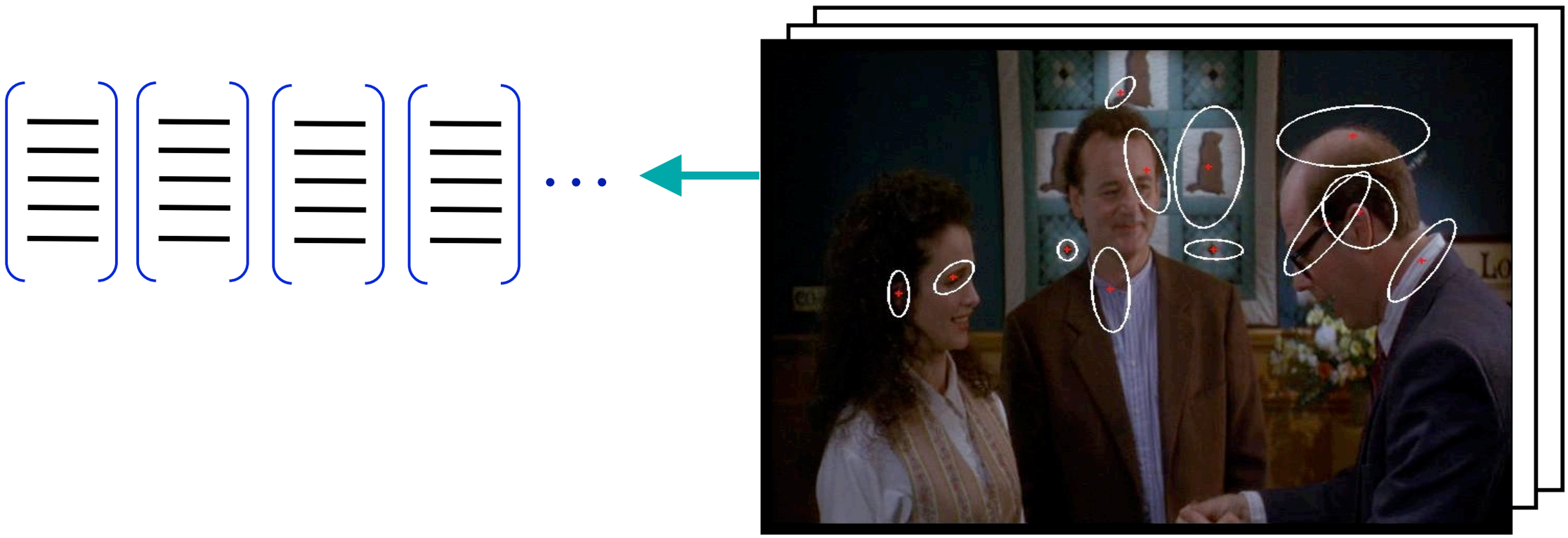


Detect patches

Local interest operator
(e.g. Harris-Laplace)
or regular grid

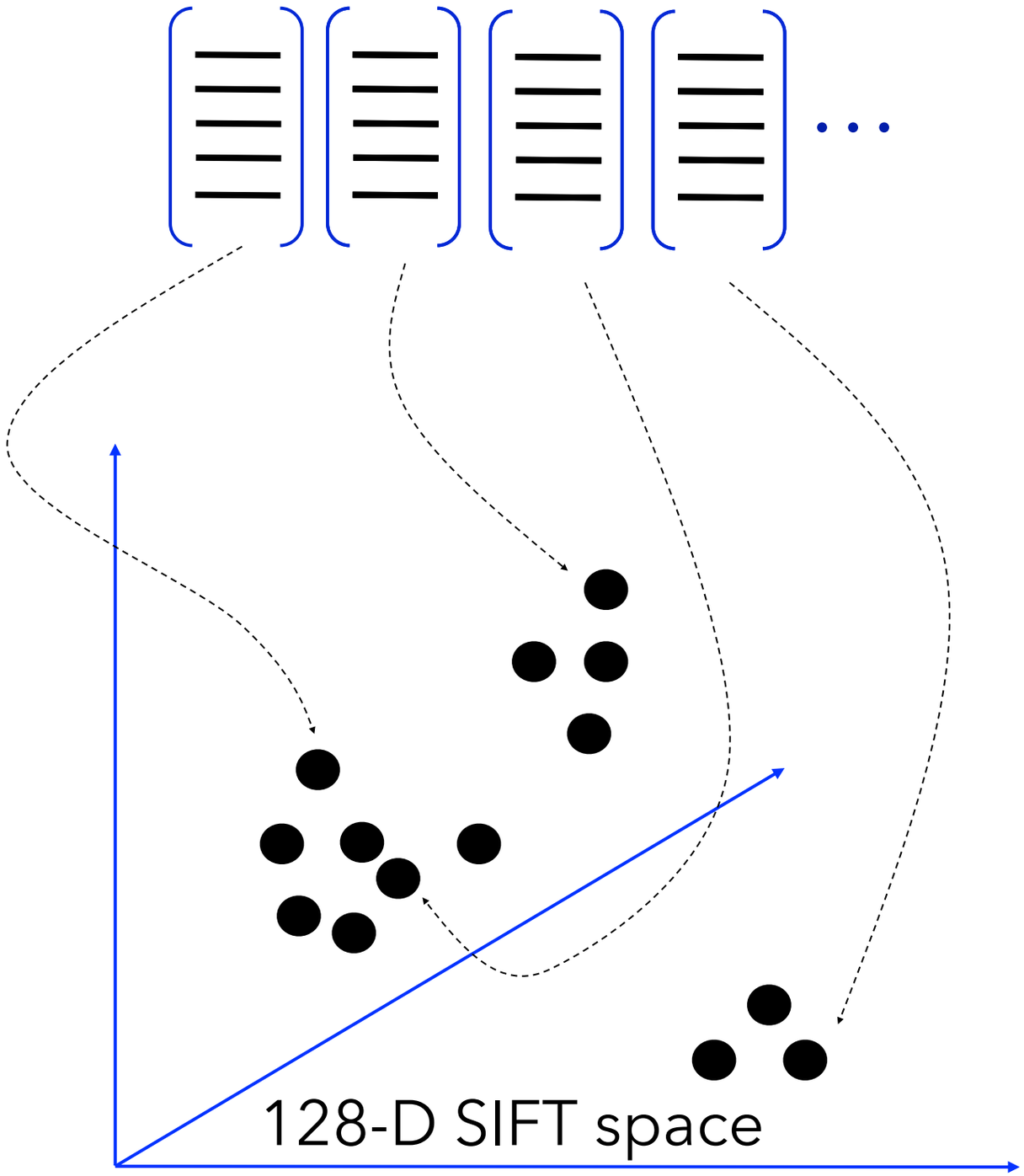
BoW-1. Feature detection and representation

TECHNISCHE UNIVERSITÄT DARMSTADT



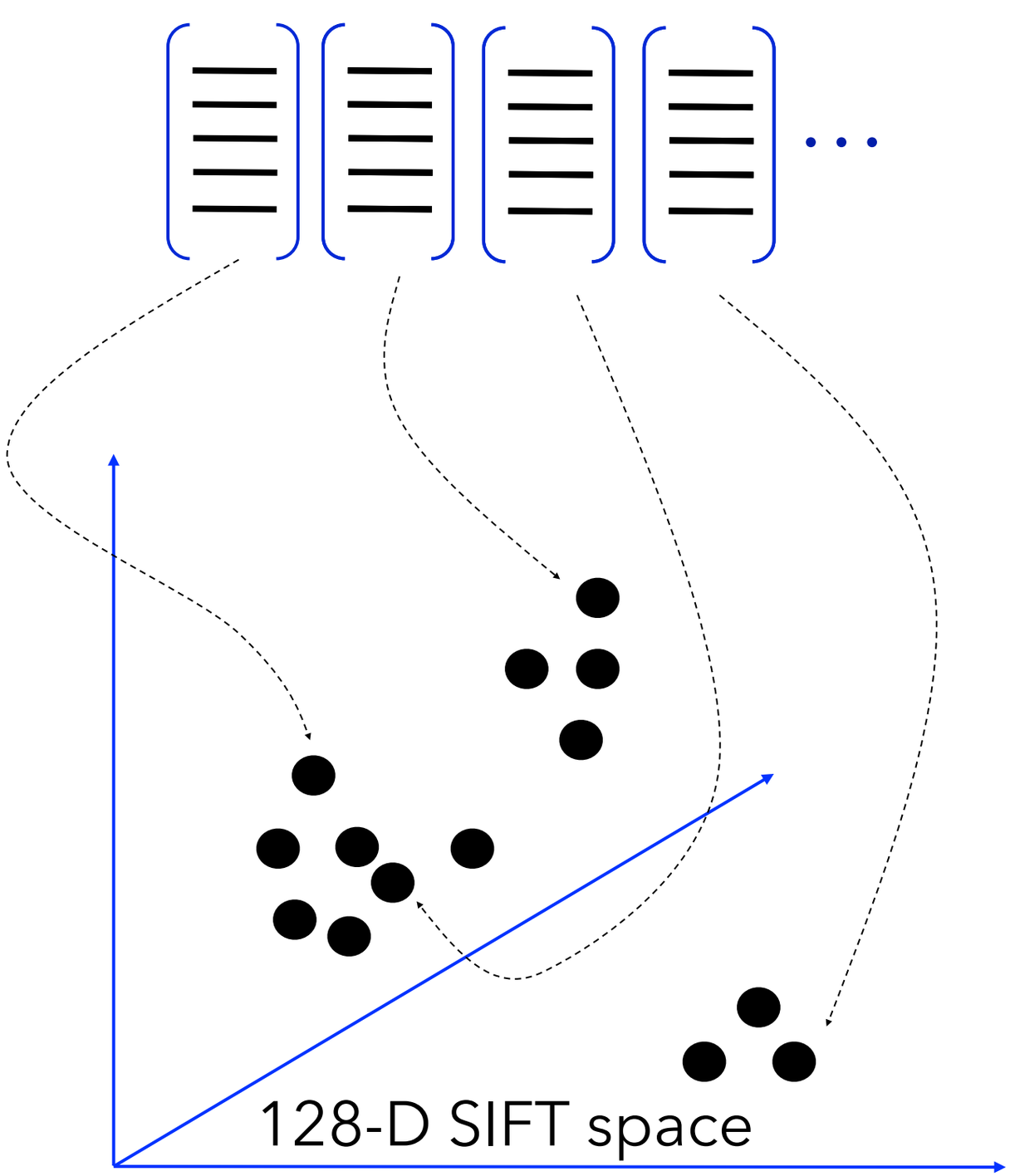
BoW-2. Codeword dictionary formation

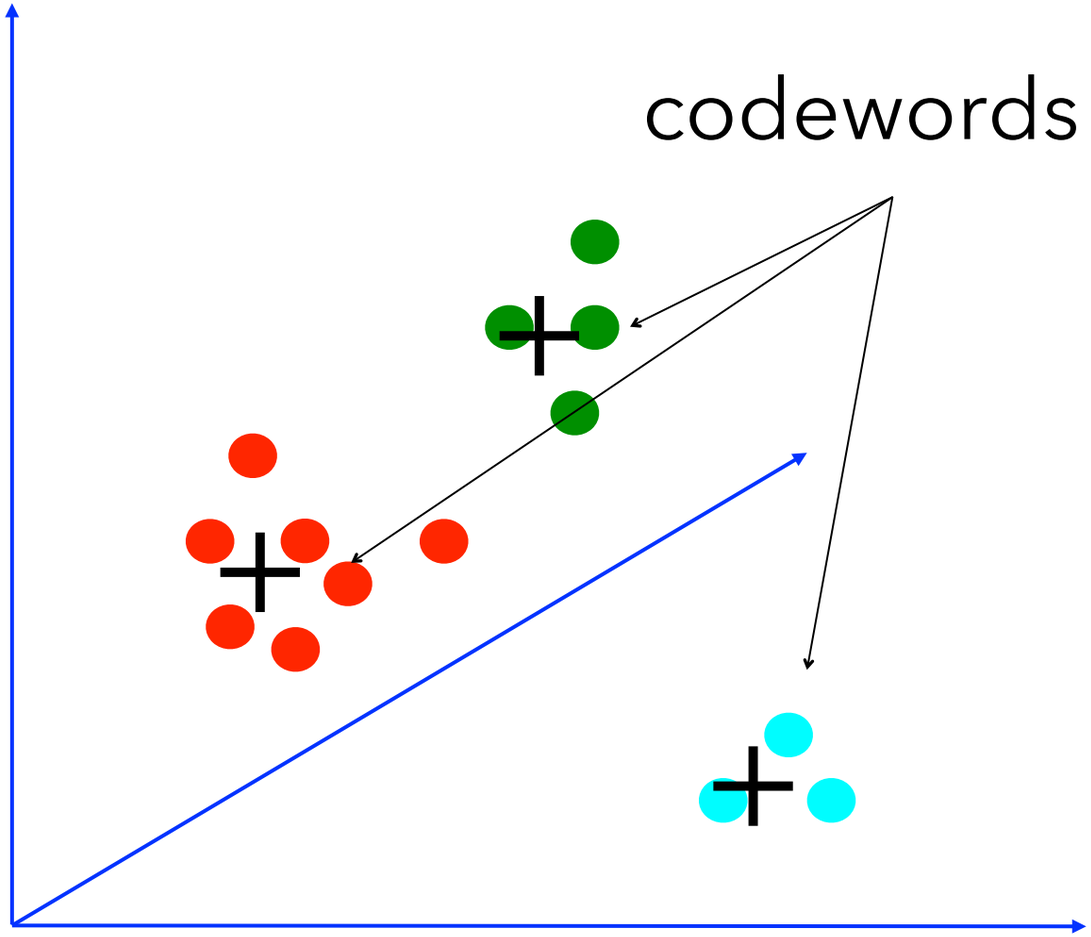
TECHNISCHE UNIVERSITÄT DARMSTADT



BoW-2. Codeword dictionary formation

TECHNISCHE UNIVERSITÄT DARMSTADT

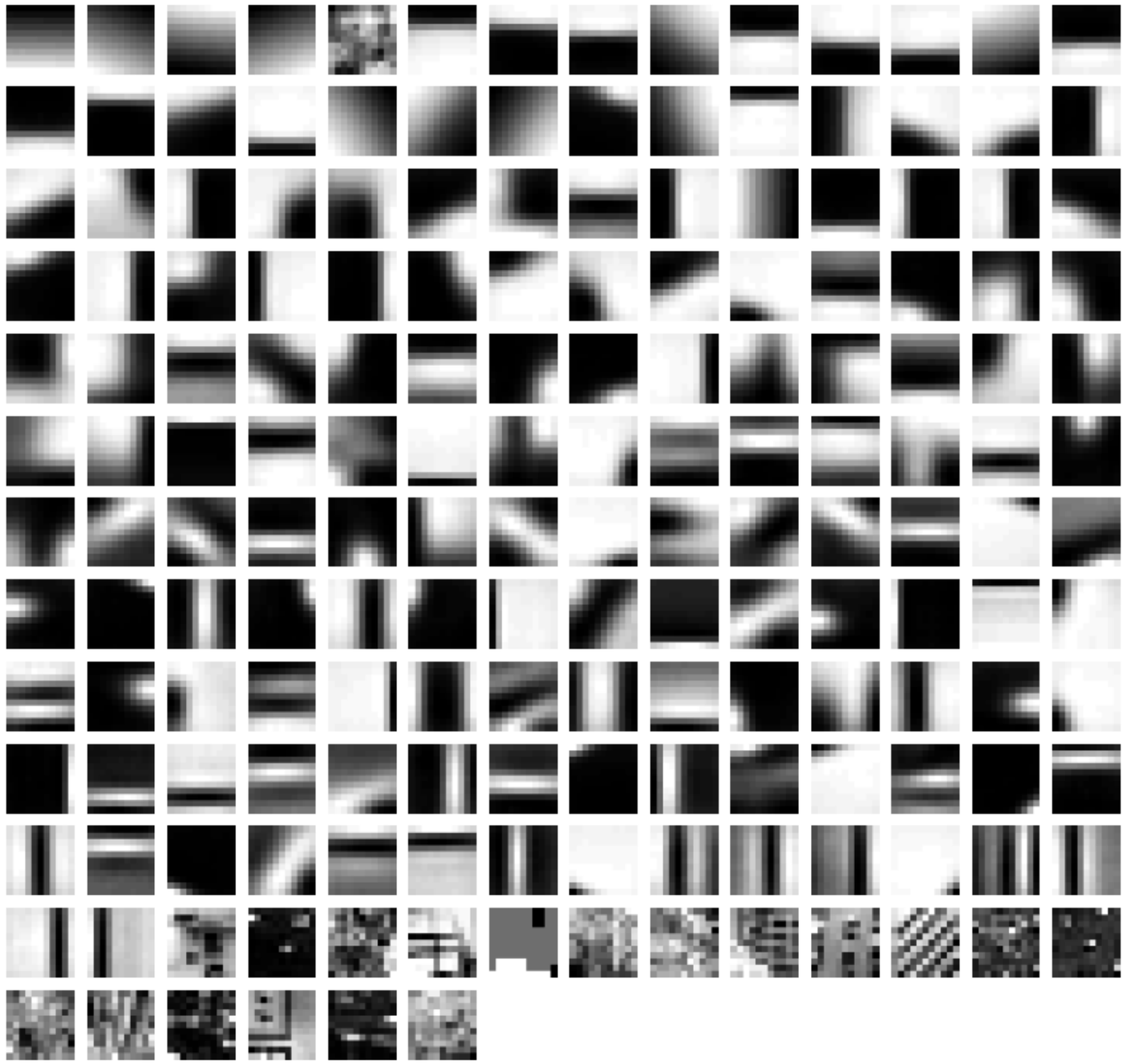




Vector quantization

e.g. k-means

BoW-2. Codeword dictionary formation

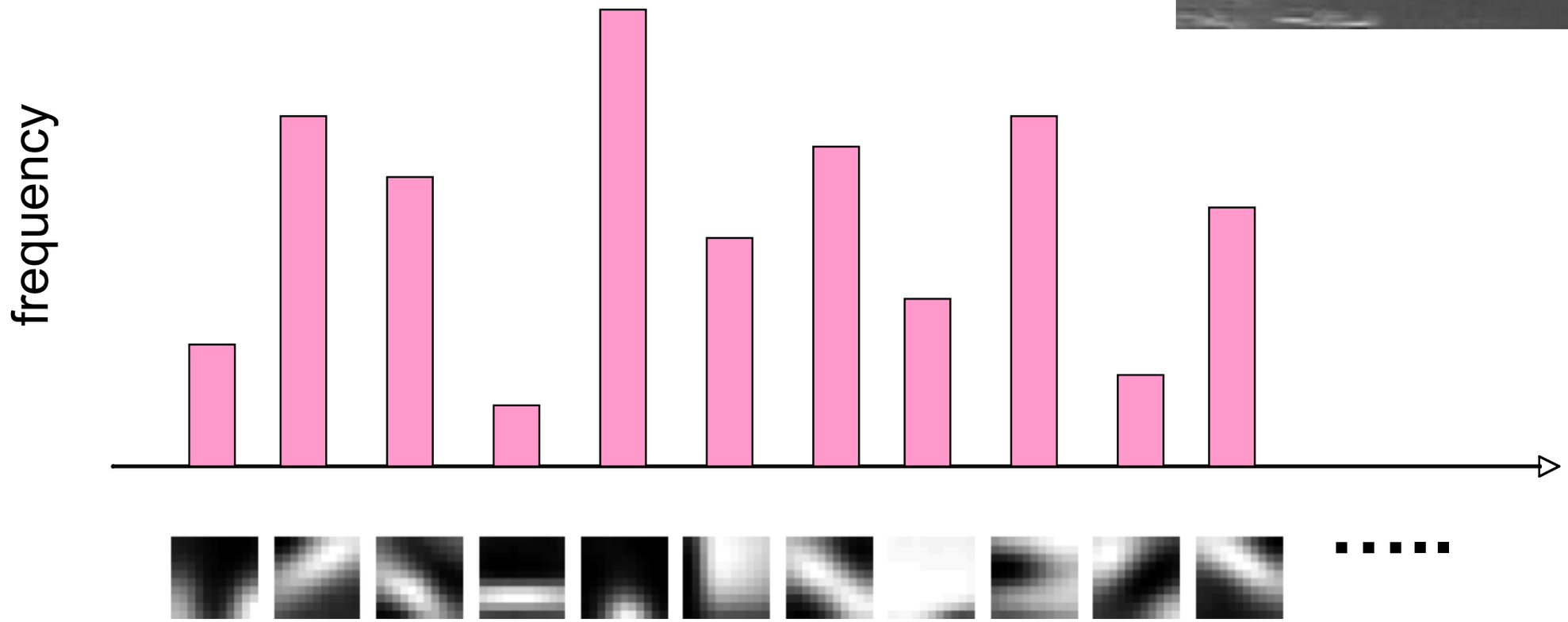


[Fei-Fei]

BoW-3. Image representation

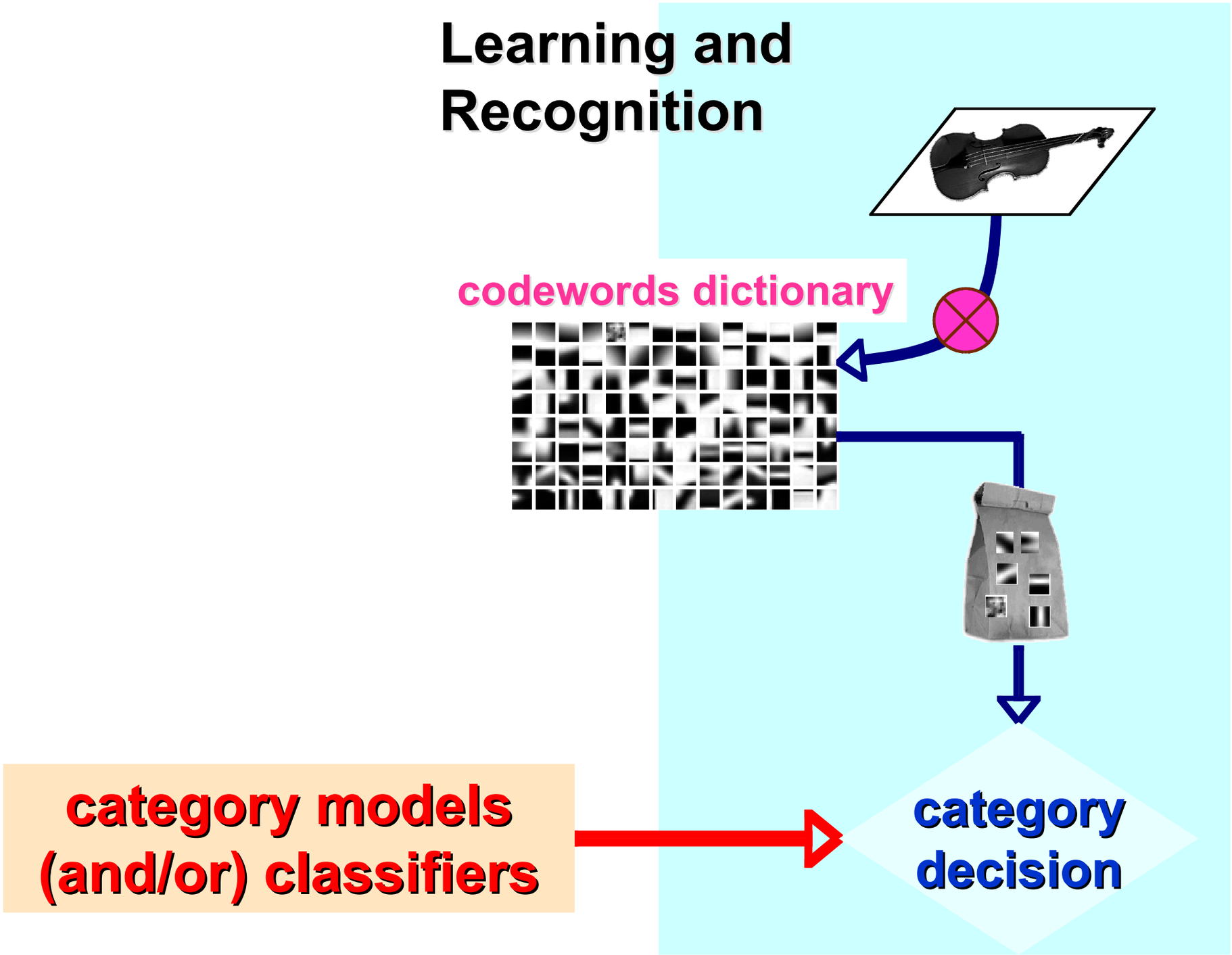
TECHNISCHE UNIVERSITÄT DARMSTADT

- Histogram of features assigned to each cluster



Next: Actual Recognition

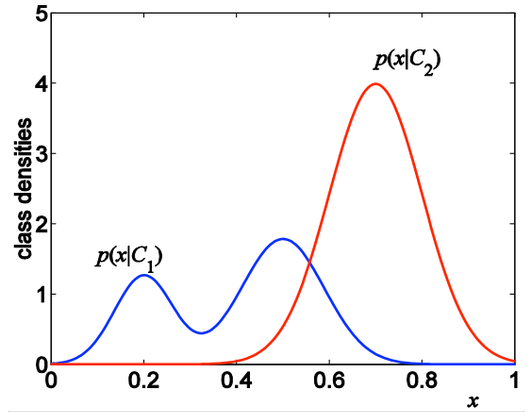
TECHNISCHE UNIVERSITÄT DARMSTADT



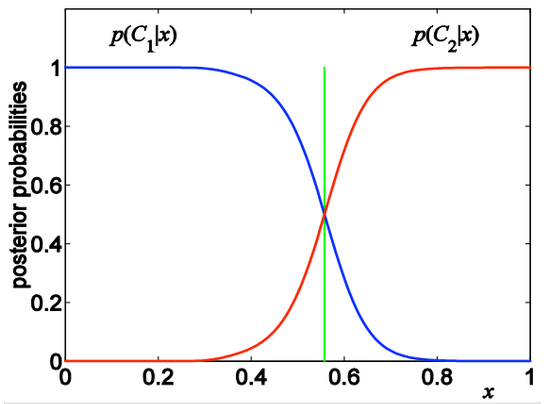
Learning and Recognition

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Generative approach:**
  - graphical models



- Discriminative approach:
  - Support Vector Machine (SVM)



category models (and/or) classifiers

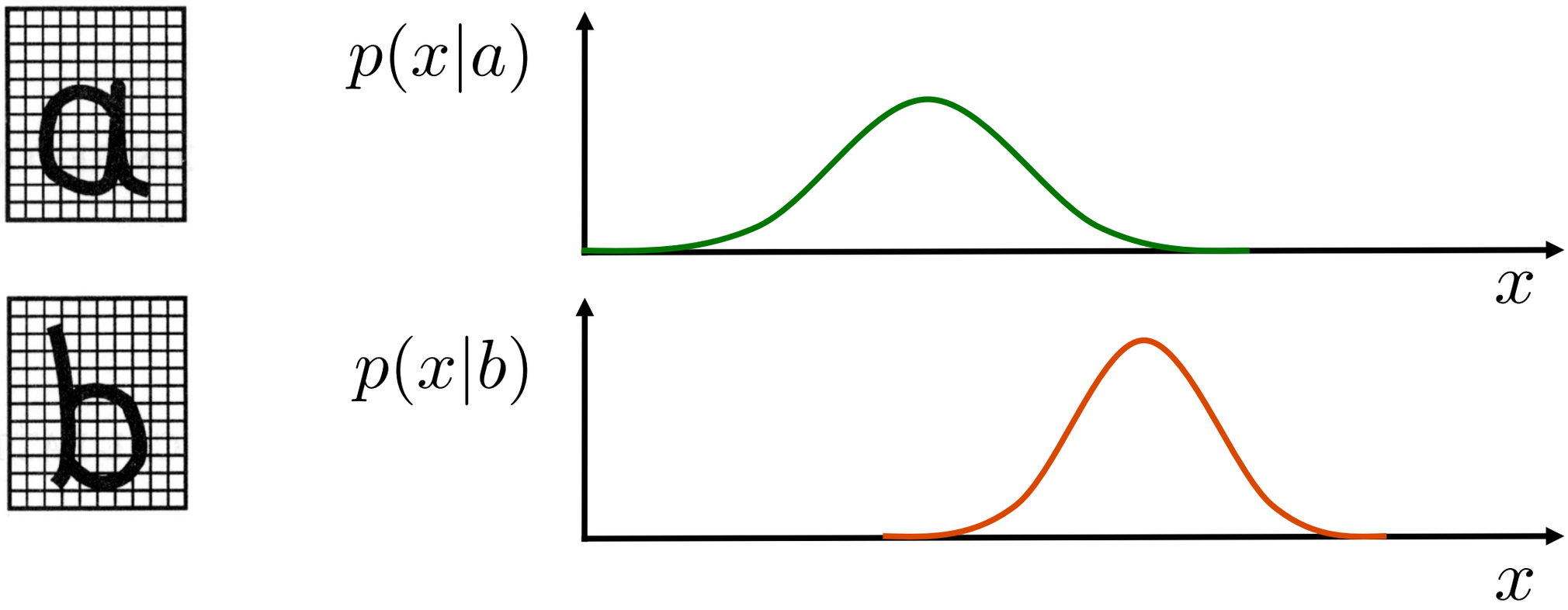
Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- 1st concept: Class conditional probabilities

$$p(x|C_k)$$

  - Probability of making an observation x knowing that it comes from some class C_k.
  - Here x is a feature (vector).
  - x measures / describes properties of the data.



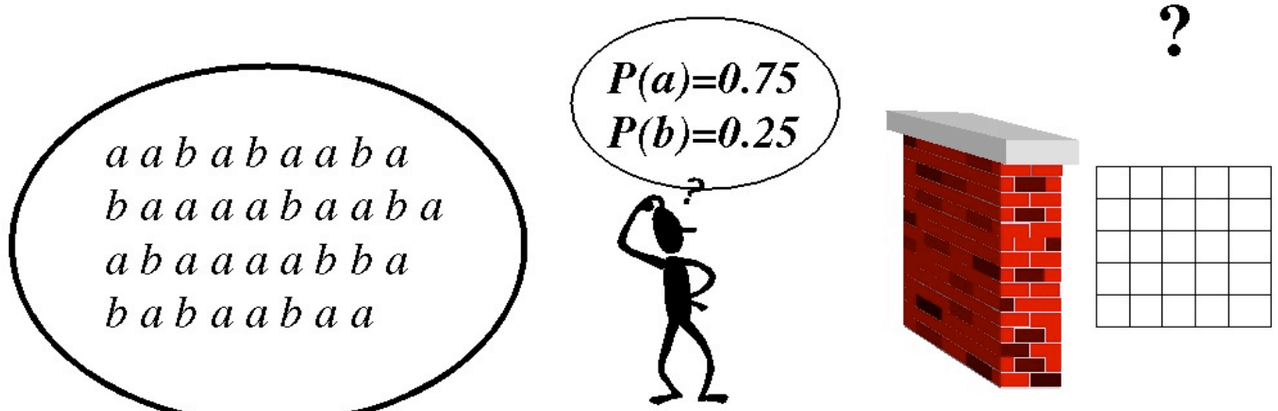
Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- 2nd concept: **Class priors** $p(C_k)$

(a priori probability of a data point belonging to a particular class)

- Example:



$$C_1 = a \qquad p(C_1) = 0.75$$

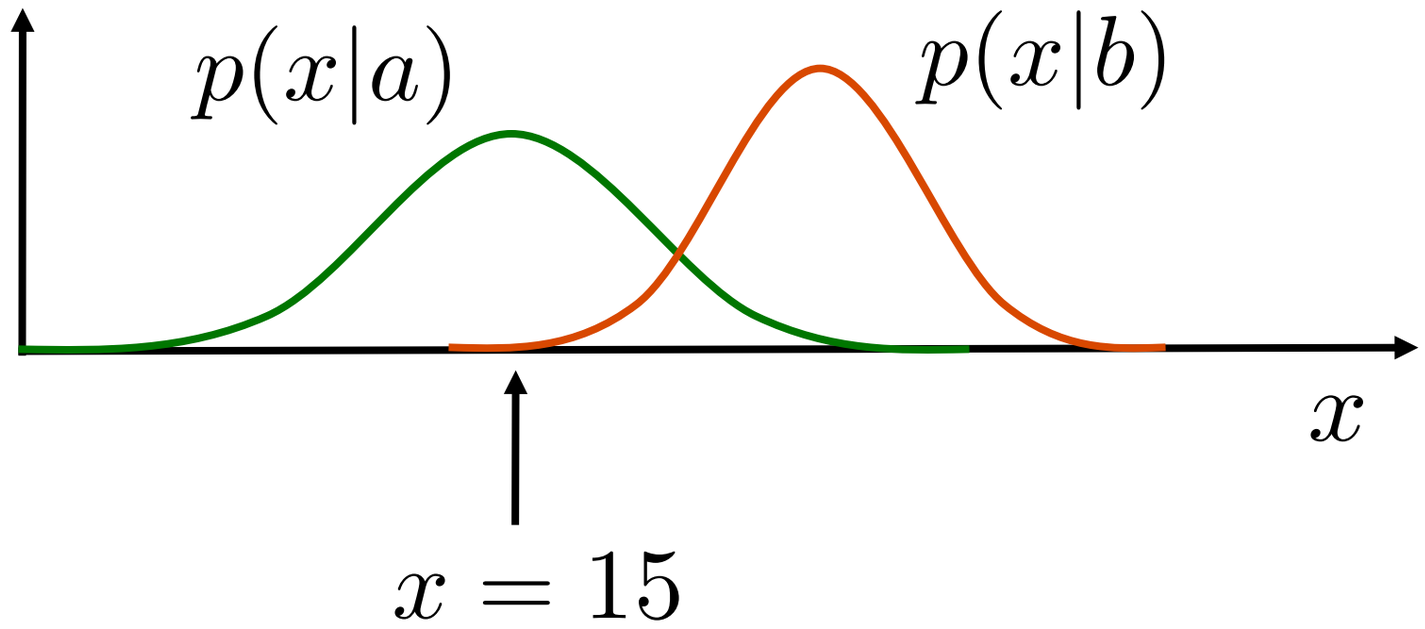
$$C_2 = b \qquad p(C_2) = 0.25$$

- Generally: $\sum_k p(C_k) = 1$

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example:



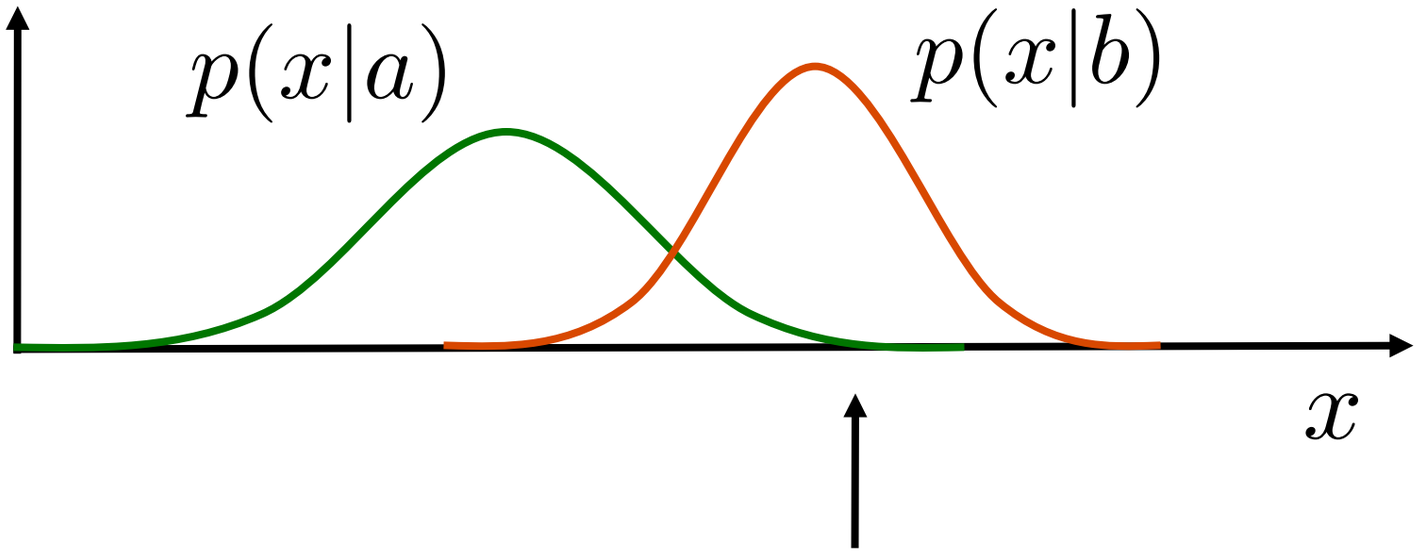
Question:

- How do we decide which class the data point belongs to?
- Here, we should decide for class a .

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example:

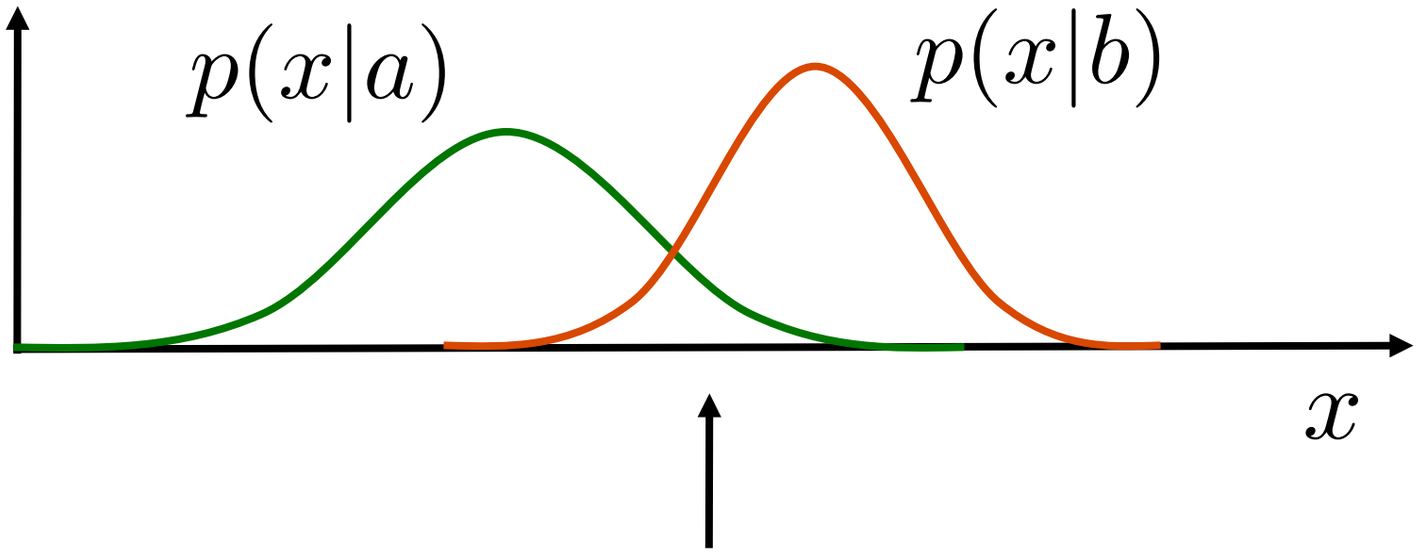


- Question:
  - How do we decide which class the data point belongs to?
  - Since $p(x|a)$ is a lot smaller than $p(x|b)$ we should now decide for class b.

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example:



- Question:
  - How do we decide which class the data point belongs to?
  - Remember that $p(a) = 0.75$ and $p(b) = 0.25$
  - This means we should decide class a.

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Formalize this using Bayes' theorem:
  - We want to find the a-posteriori probability (posterior) of the class C_k given the observation (feature) x

class-conditional probability (likelihood)

class posterior

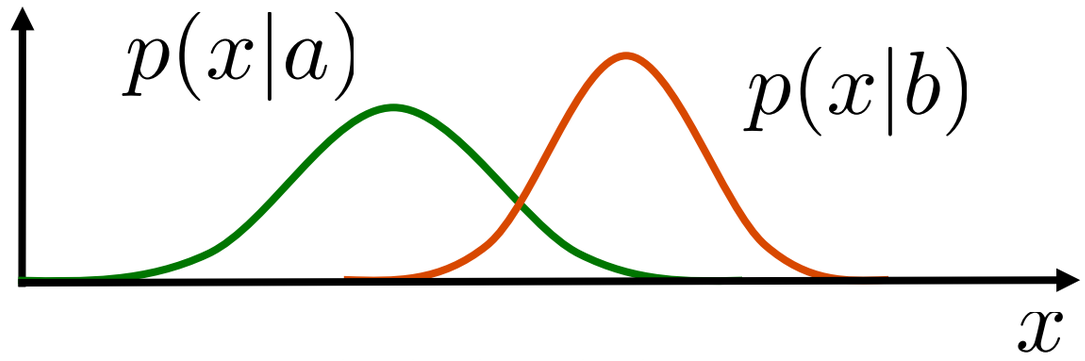
class prior

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

normalization term

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)} = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)}$$

Bayesian Decision Theory

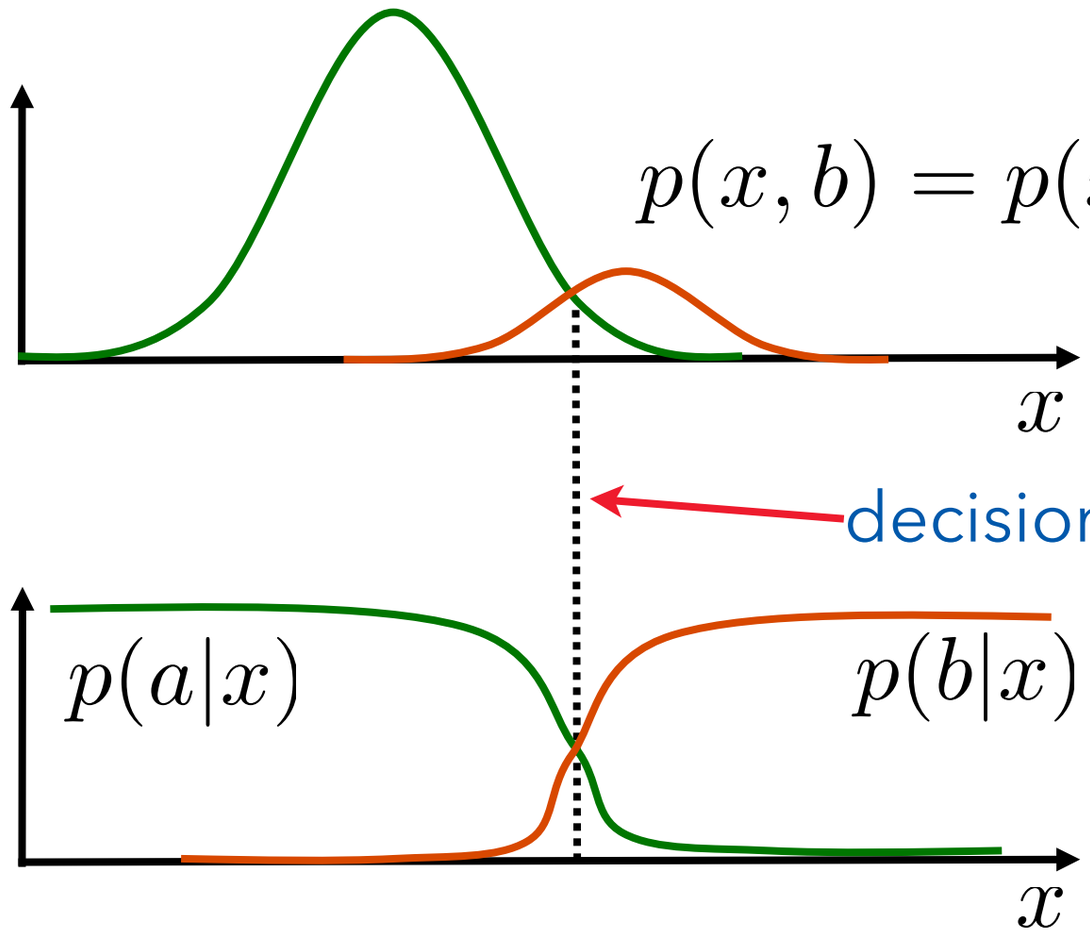


Likelihood

$$p(x,a) = p(x|a)p(a)$$

$$p(x,b) = p(x|b)p(b)$$

Likelihood × *Prior*



$$Posterior = \frac{Likelihood \times Prior}{Normalization\ factor}$$

Bayesian Decision Theory

TECHNISCHE UNIVERSITÄT DARMSTADT

- Decision rule:
  - Decide C_1 if $p(C_1|x) > p(C_2|x)$

We do not need the normalization!

  - This is equivalent to

$$p(x|C_1)p(C_1) > p(x|C_2)p(C_2)$$

  - Which is equivalent to

$$\frac{p(x|C_1)}{p(x|C_2)} > \frac{p(C_2)}{p(C_1)}$$

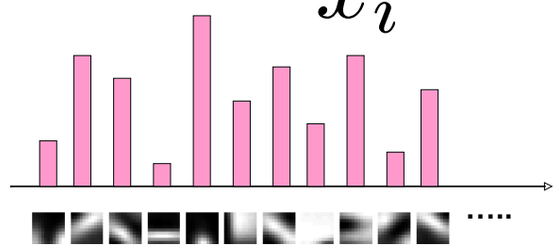
- Bayes optimal classifier:
  - A classifier obeying this rule is called a Bayes optimal classifier.

Naïve Bayes Classifier

TECHNISCHE UNIVERSITÄT DARMSTADT

- Classify image using histograms of occurrences on visual words:

$$\mathbf{x} = \quad x_i \qquad x_i \in [0, 1]$$



- simpler alternative: only take presence/absence of a word into account: $x_i \in \{0, 1\}$
- Naïve Bayes classifier assumes that visual words are conditionally independent given object class

$$P(\mathbf{x}|c) = \prod_{i=1}^{m} P(x_i|c)$$

[T. Hofmann]

Naive Bayes Classifier

TECHNISCHE UNIVERSITÄT DARMSTADT

- Multinomial model for each object class:

$$P(\mathbf{x}|c) = \prod_{i=1}^{m} P(x_i|c)$$

- Class priors: $P(c)$, with $\sum_c P(c) = 1$

- Posterior probabilities:

$$P(c|\mathbf{x}) = \frac{P(c)\prod_{t=1}^{n} P(x_t|c)}{\sum_{c'} P(c')\prod_{t=1}^{n} P(x_t|c')}$$

Naive Bayes Classifier: Decision

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bayes optimal decision:

$$c^* = \text{argmax}_c \; P(c|\mathbf{x})$$

$$= \text{argmax}_c \left[\log P(c) + \sum_{t=1}^{n} \log P(x_t|c) \right]$$

Naive Bayes Classifier: Estimation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Count empirical frequencies

$$\hat{P}(c) = \frac{1}{n}|\{i : y_i = c\}|$$

$$\hat{P}(w|c) = \frac{\sum_{i:y_i=c} |\{t : x_{it} = w\}|}{\sum_i n_i}$$

Training Set:

$$\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$$

$$\mathbf{x}_i \in \{w_1, \ldots, w_m\}^{n_i}$$

- Smoothing (e.g. Laplace)

$$\hat{P}(w|c) = \frac{1 + \sum_{i:y_i=c} |\{t : x_{it} = w\}|}{m + \sum_i n_i}$$

Image Classification with Naive Bayes

TECHNISCHE UNIVERSITÄT DARMSTADT

- Image dataset: 7 object categories, arbitrary views, partial occlusions

[Csurka]

Example of feature extraction

TECHNISCHE UNIVERSITÄT DARMSTADT

All features detected in the image

Features corresponding to two different visual words

[Csurka]

Recognition results:

TECHNISCHE UNIVERSITÄT DARMSTADT

Table 1. Confusion matrix and the mean rank for the best vocabulary (*k=1000*).

True classes →	*faces*	*buildings*	*trees*	*cars*	*phones*	*bikes*	*books*
faces	**76**	4	2	3	4	4	13
buildings	2	**44**	5	0	5	1	3
trees	3	2	**80**	0	0	5	0
cars	4	1	0	**75**	3	1	4
phones	9	15	1	16	**70**	14	11
bikes	2	15	12	0	8	**73**	0
books	4	19	0	6	7	2	**69**
Mean ranks	1.49	1.88	1.33	1.33	1.63	1.57	1.57

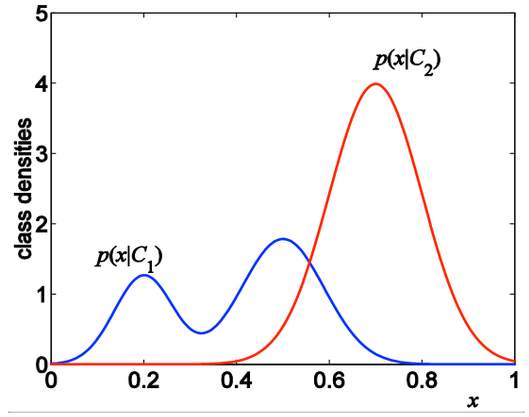
Examples of correctly classified images:



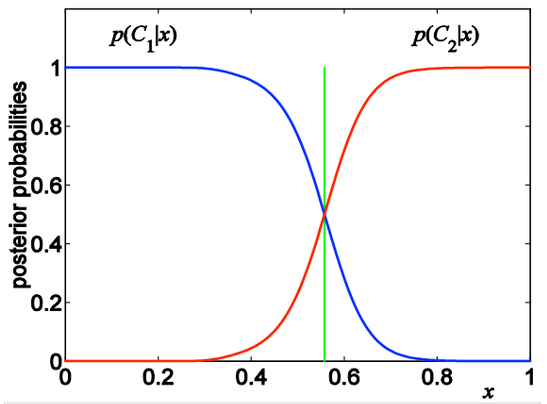

Learning and Recognition

TECHNISCHE UNIVERSITÄT DARMSTADT

- Generative approach:
  - graphical models



- **Discriminative approach:**
  - Support Vector Machine (SVM)



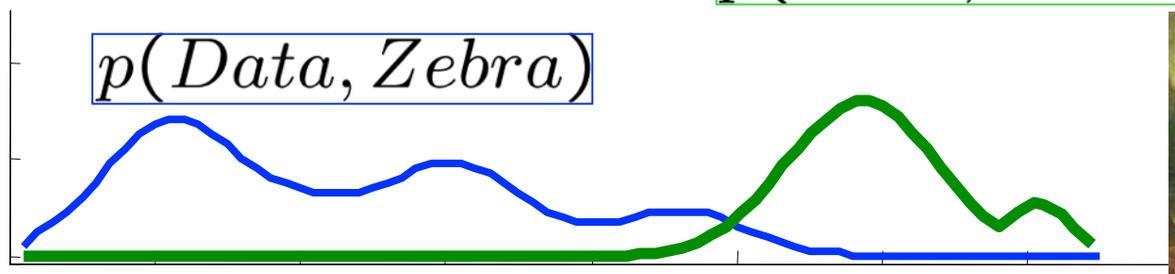
category models (and/or) classifiers

Discriminative vs. generative

TECHNISCHE UNIVERSITÄT DARMSTADT

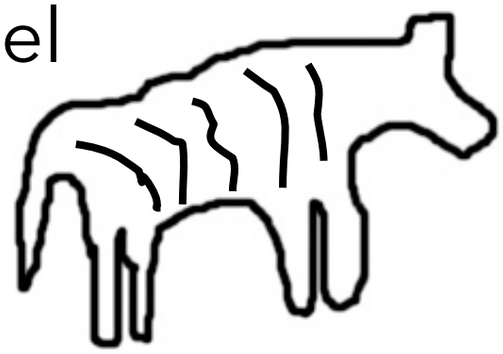
Generative model

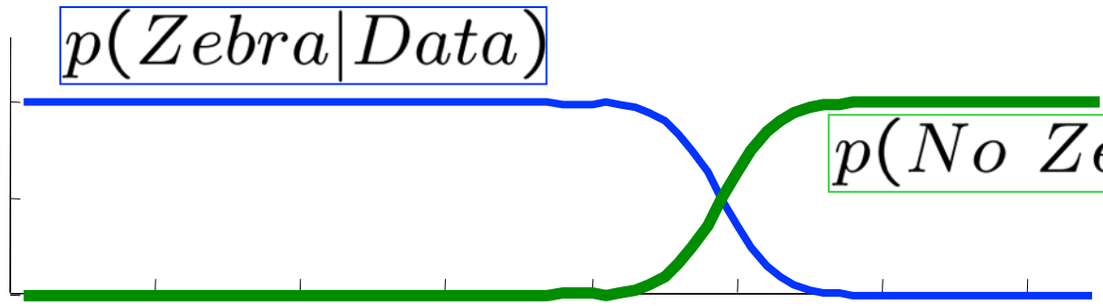
(The artist)

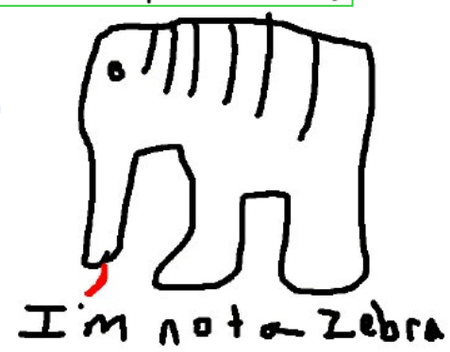


Discriminative model

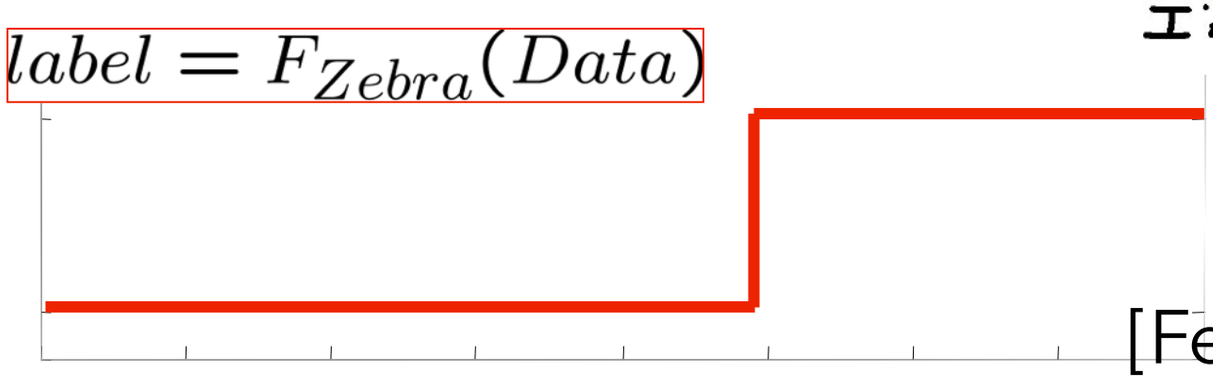
(The lousy painter)







Discriminant function

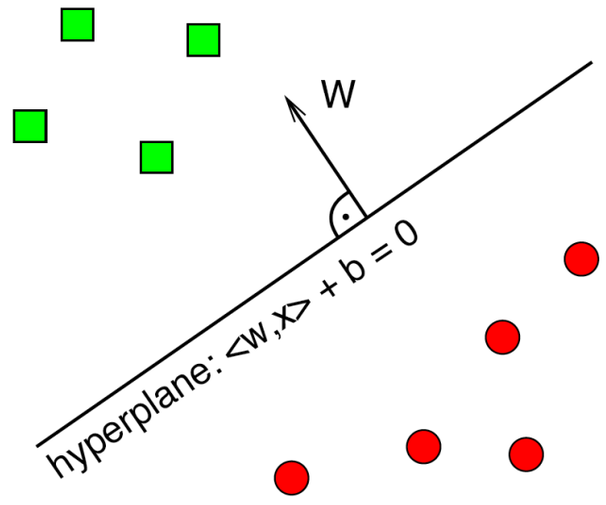


[Fergus]

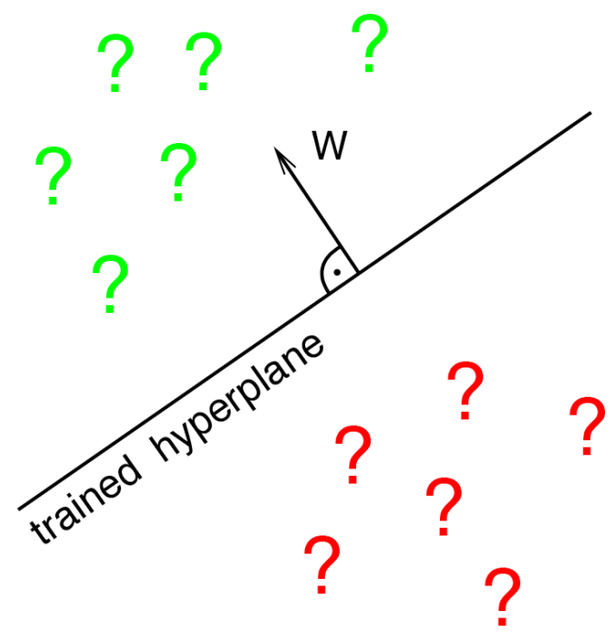
Discriminant Functions

TECHNISCHE UNIVERSITÄT DARMSTADT

- Linear discriminant function:
  - Linear hyperplane:
    $$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$
  - trained on samples of both classes
    C_1 (■) and C_2 (●)

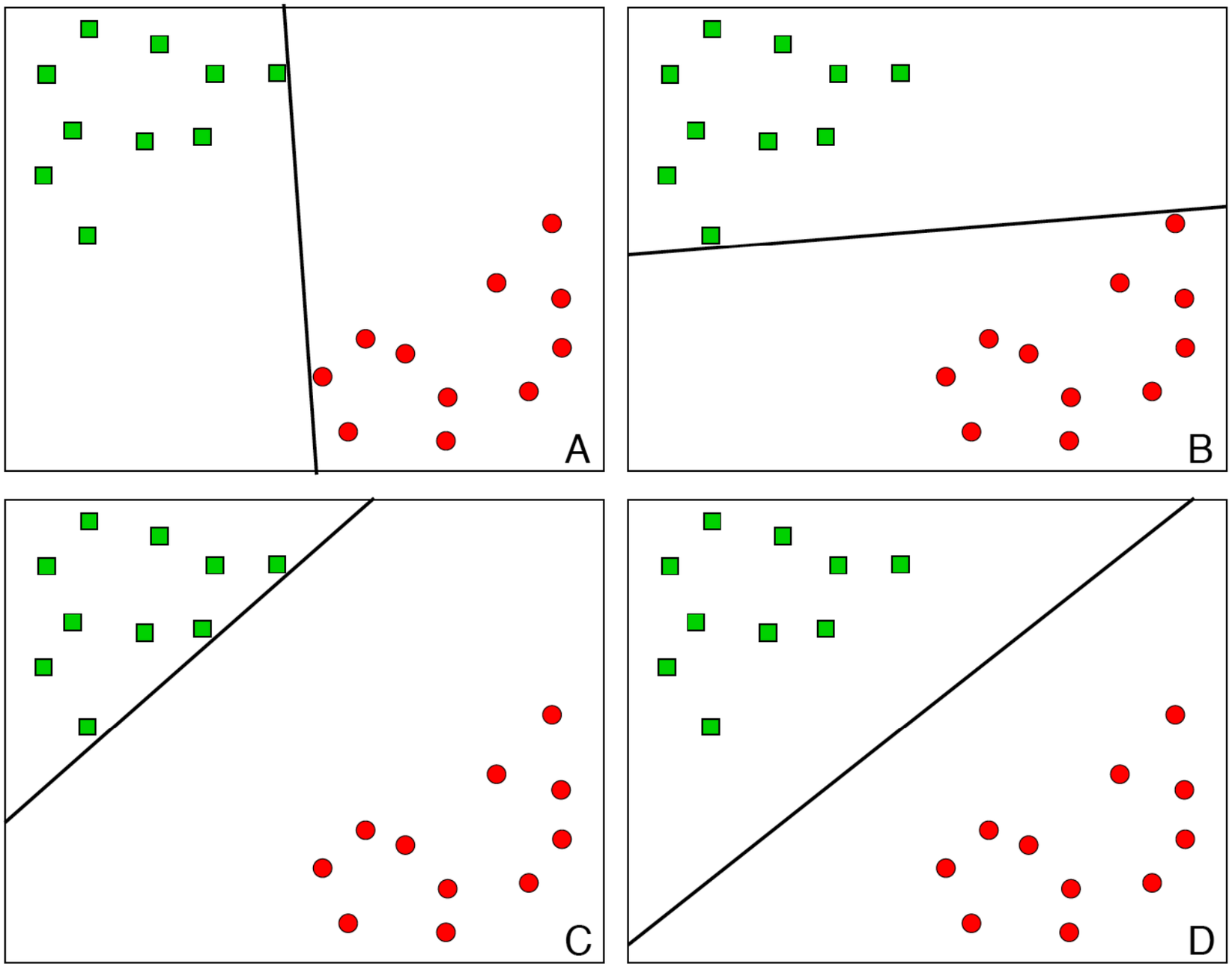


- Classification:
  - decide class C_1 (?) when $y(\mathbf{x}) > 0$
  - decide class C_2 (?) when $y(\mathbf{x}) < 0$



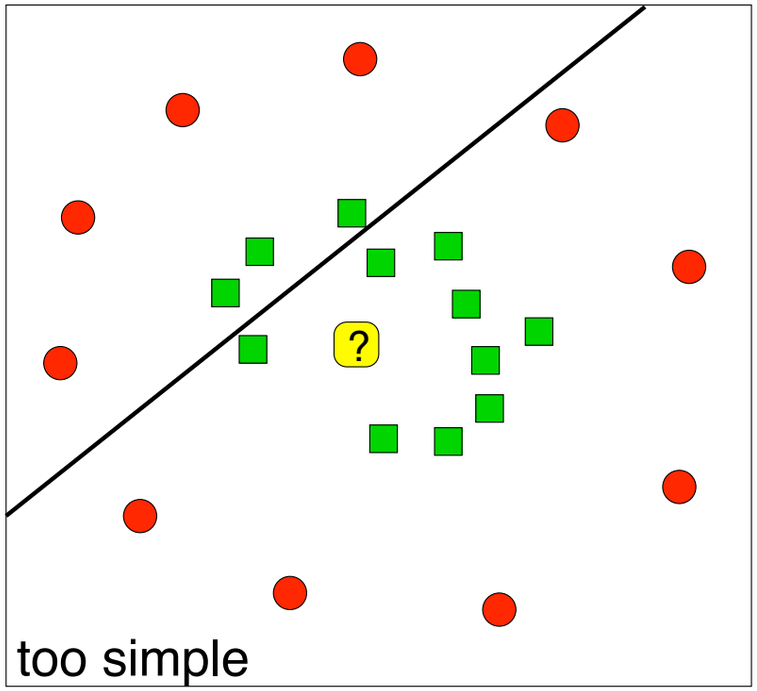
Which Hyperplane is Best? and Why?

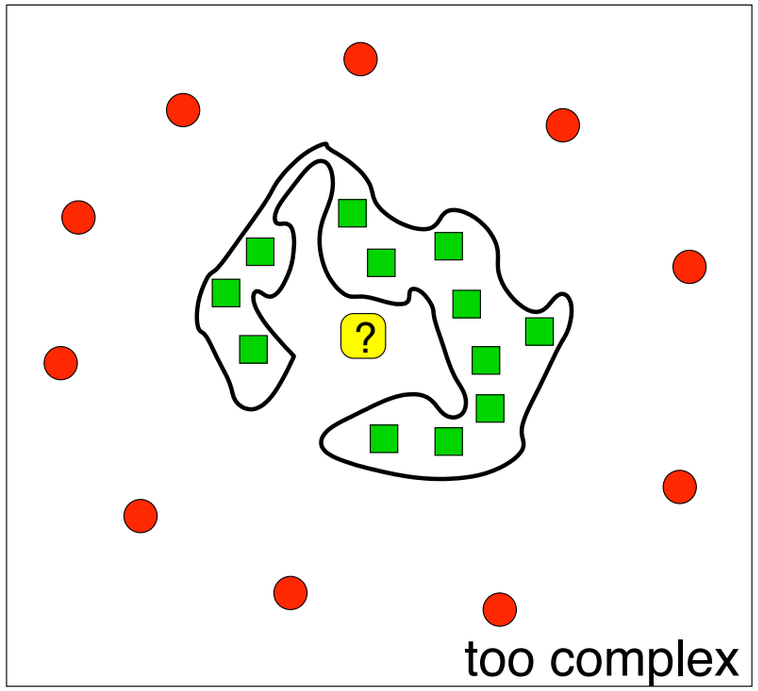
TECHNISCHE UNIVERSITÄT DARMSTADT



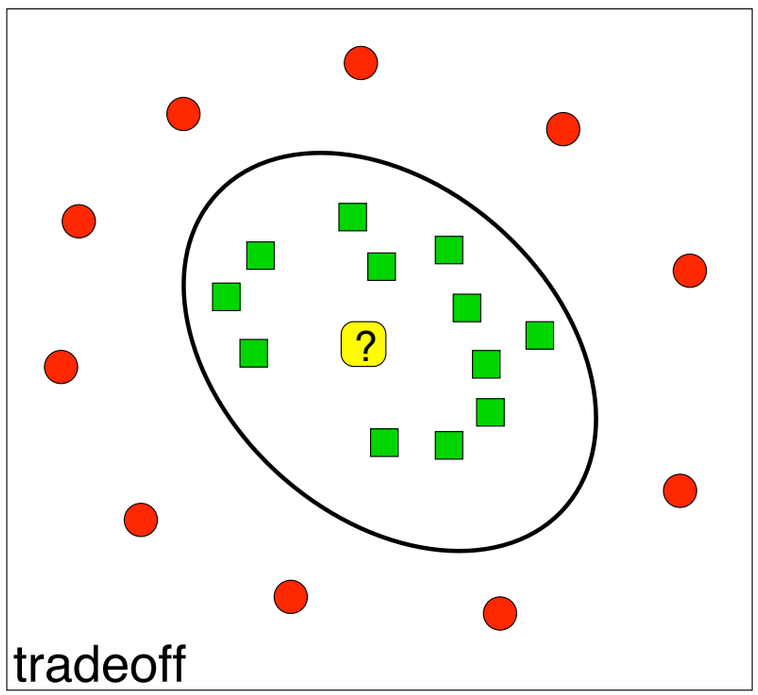
"Learning Power"

TECHNISCHE UNIVERSITÄT DARMSTADT





negative example
positive example
? new patient



[Florian Markowetz]

Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

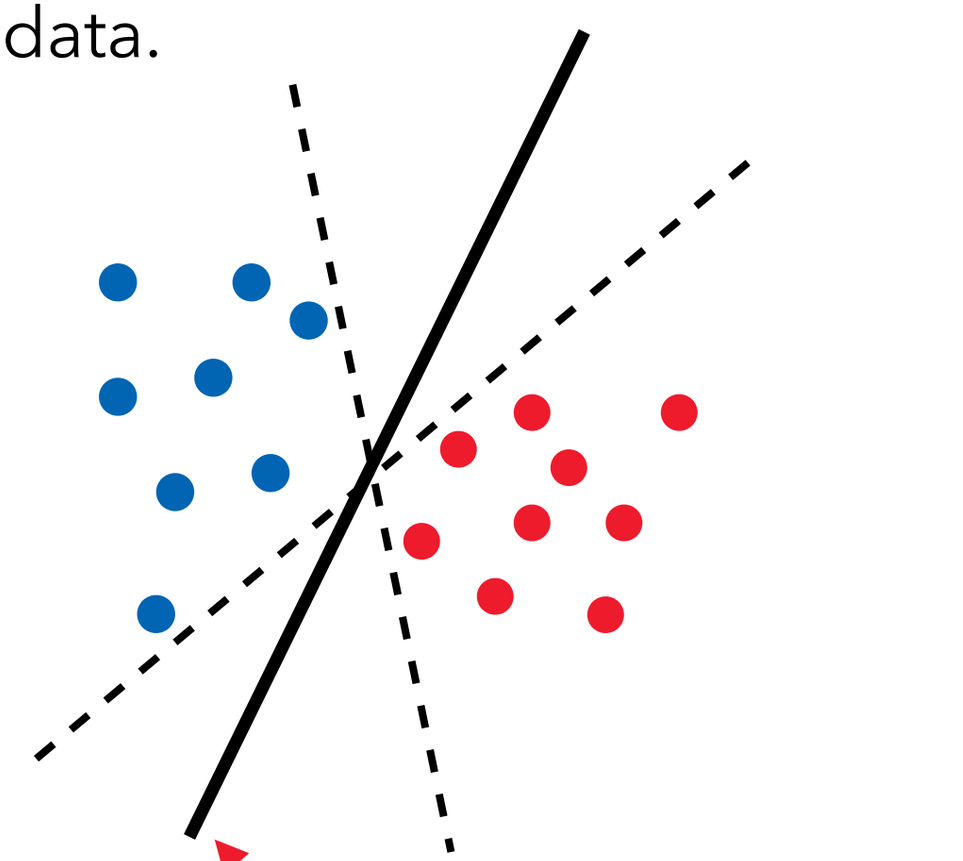
- Basic idea
  - Limit the “learning power”
  - Minimize an upper bound on the true risk (generalization ability)

- Key result (Vapnik):
  - Assume that the data points lie in a sphere of fixed radius
  - The linear classifier with maximal margin minimizes the learning power (VC dimension)
  - This in turn lowers a bound on the true risk
  - Leads to good generalization to unseen data

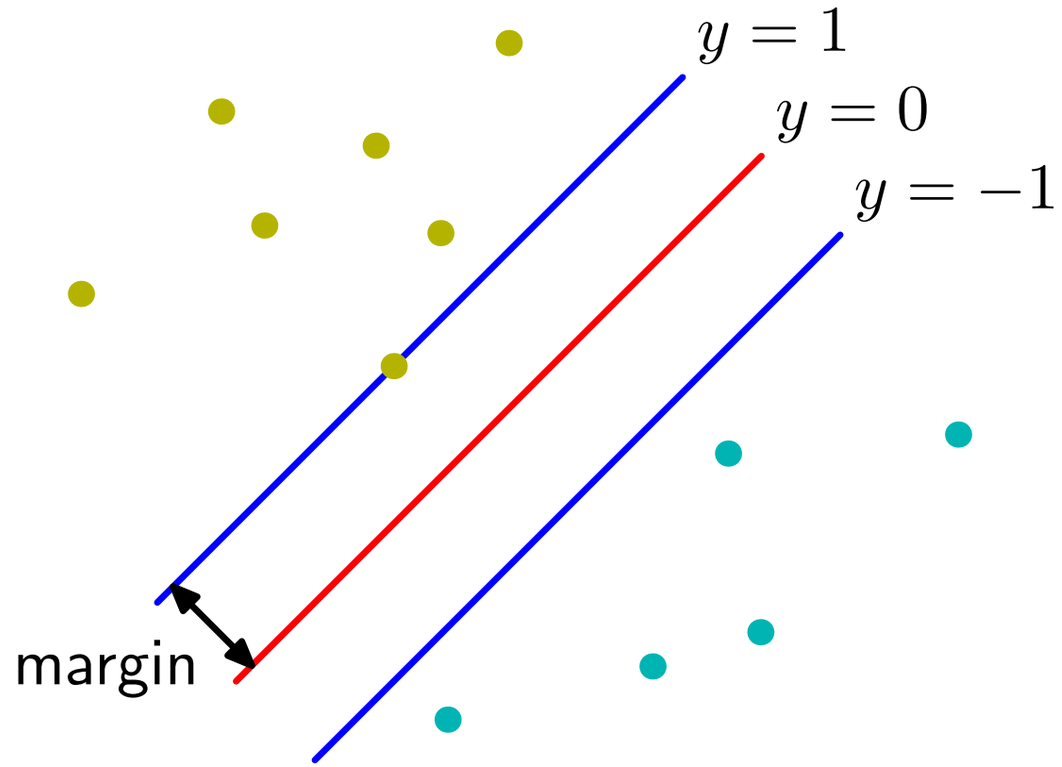
Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- Intuitively:
  - We should find the hyperplane with the maximum "distance" to the data.



generalizes best to unseen data



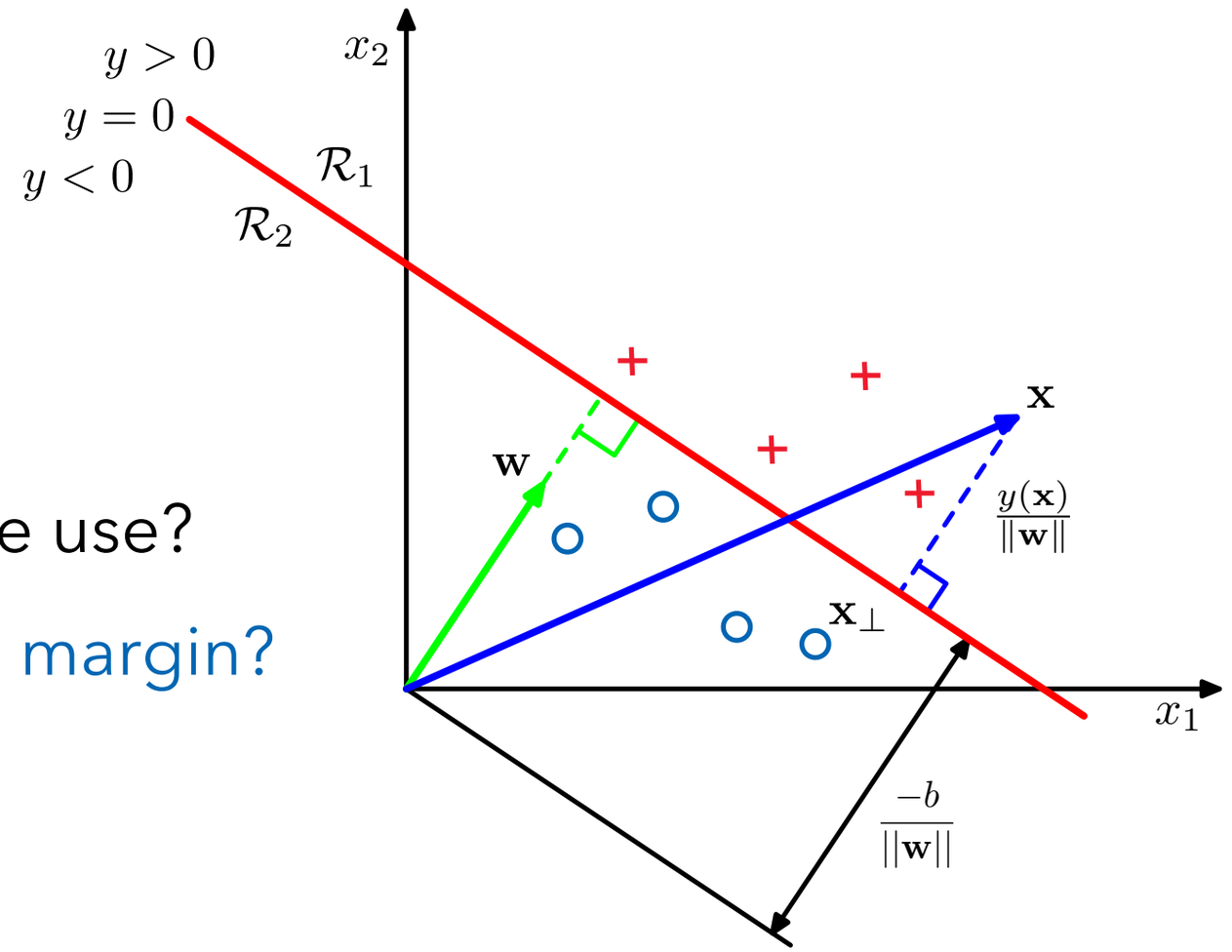
Maximize the margin
(distance to the closest data point)

Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- For now: linearly separable data $x_i \in \mathbb{R}^d$
  - N training data points: $\{x_i, y_i\}_{i=1}^N$ $y_i \in \{-1, 1\}$
  - Hyperplane that separates the data:

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$



  - Which hyperplane shall we use?
  - How can we maximize the margin?

Support Vector Machines

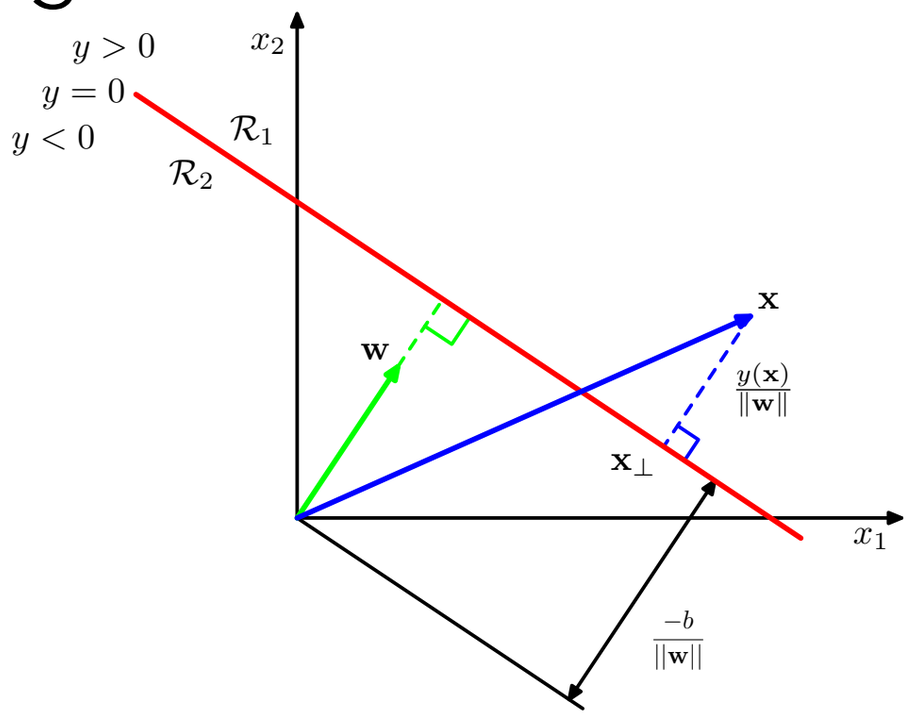
TECHNISCHE UNIVERSITÄT DARMSTADT

- Want to find a hyperplane so that the data is linearly separated:

$$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1, \qquad \forall i$$

  - Enforce $y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) = 1$ for at least one data point.

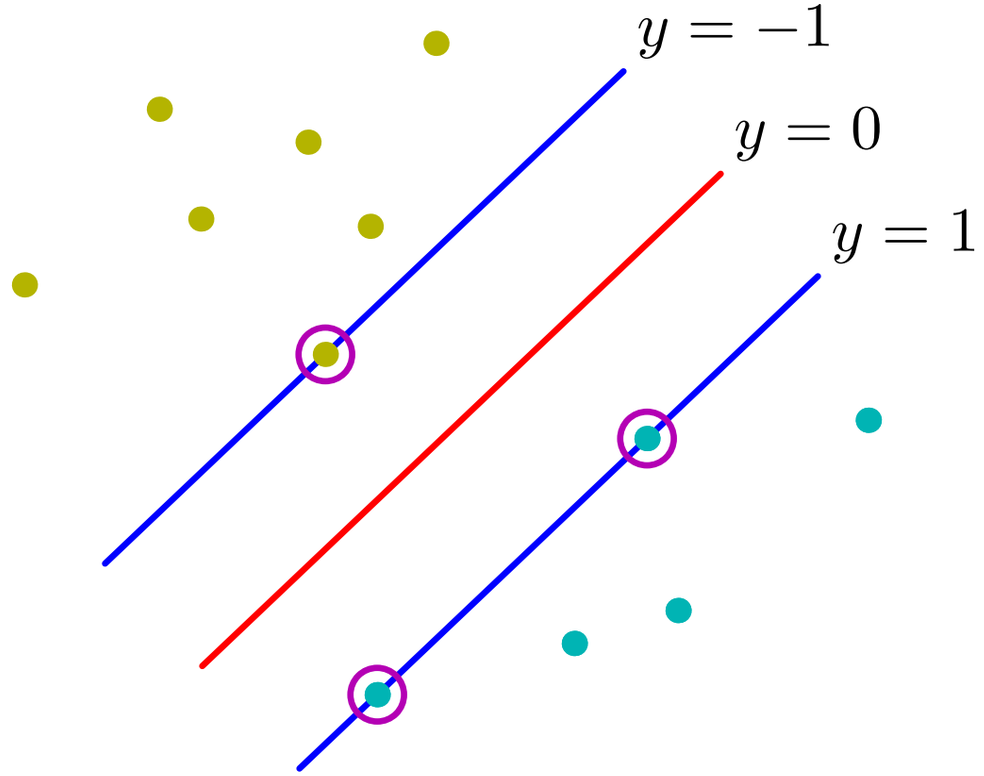
- Then we can easily express the margin:
  - The distance to the hyperplane is: $\dfrac{y(\mathbf{x}_i)}{||\mathbf{w}||} = \dfrac{\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b}{||\mathbf{w}||}$
  - This means that the margin is $\dfrac{1}{||\mathbf{w}||}$



Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- Illustration:



- Support vectors:
  - All data points that lie on the margin (i.e $y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) = 1$)

Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- Maximizing the margin $1/||\mathbf{w}||$ is equivalent to minimizing $||\mathbf{w}||^2$.
- Formulate as constrained optimization problem:

$$\arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$
$$\text{s.t.} \quad y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1 \geq 0$$

- Lagrangian formulation: $\alpha_i \geq 0$

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1)$$

  - Minimize Lagrangian

Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- Minimize $L(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1)$

- Set the gradient to zero:

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

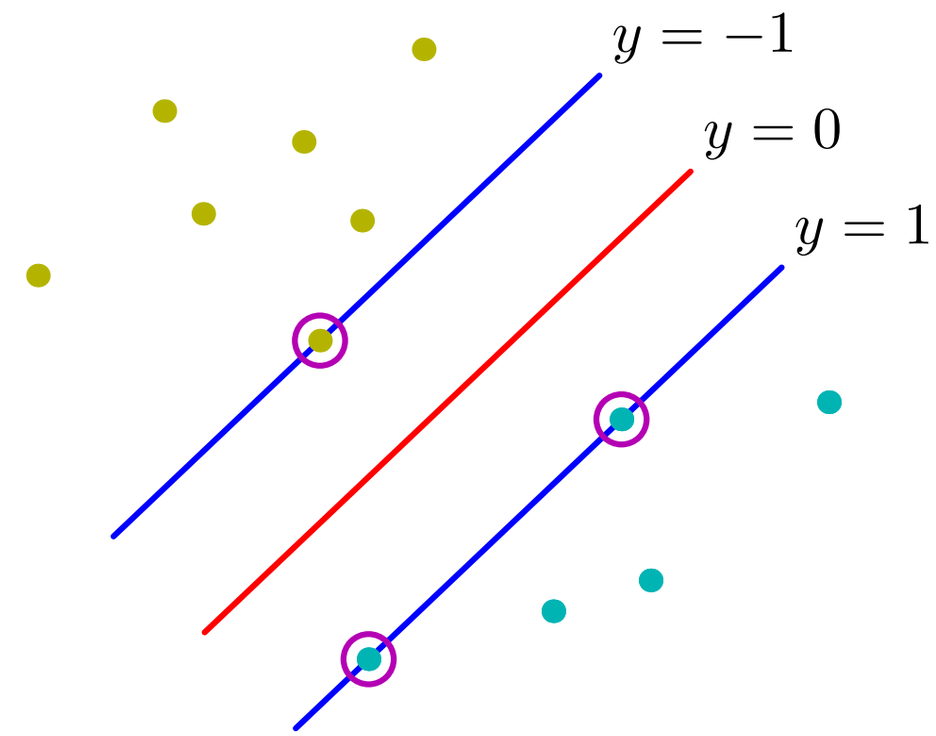
- The separating hyperplane is a linear combination of the input data.
- Solving this quadratic program is relatively easy.
  - Standard implementations exist.

Sparsity

TECHNISCHE UNIVERSITÄT DARMSTADT

- Important property:
  - Almost all the α_i are zero.
  - There are only a few support vectors.
  - But the hyperplane was written as:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

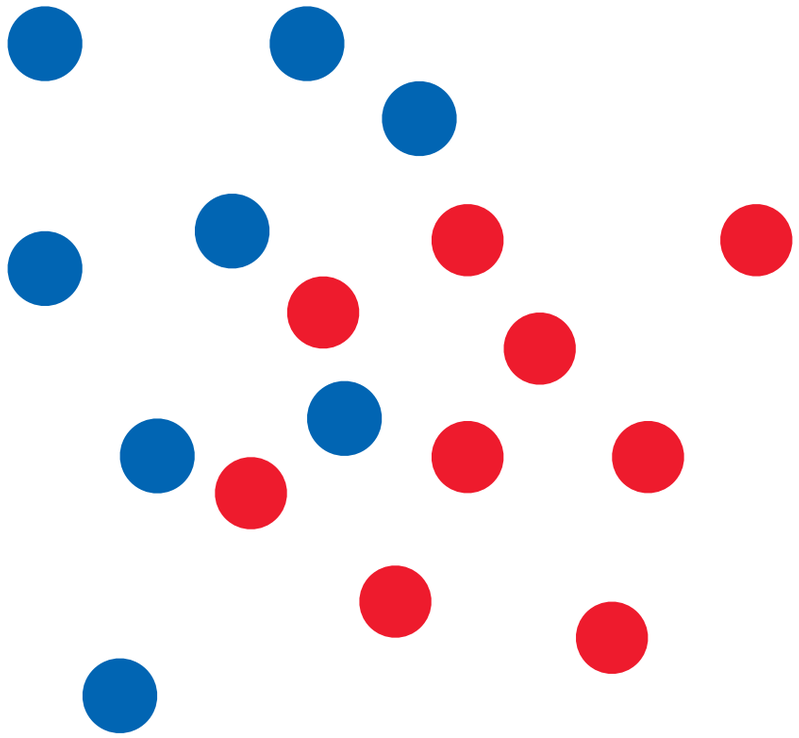


- SVMs are sparse learning machines:
  - The classifier only depends on a few data points.

Non-separable data

TECHNISCHE UNIVERSITÄT DARMSTADT

- What if our data is not linearly separable?



- Solution: Allow for data points to “violate the margin”.

Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- Instead of requiring that the data is perfectly linearly separable:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \geq +1 \qquad \text{for} \quad y_i = +1$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \leq -1 \qquad \text{for} \quad y_i = -1$$

- we allow for small violations ξ_i from perfect separation:

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \geq +1 - \xi_i \qquad \text{for} \quad y_i = +1$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b \leq -1 + \xi_i \qquad \text{for} \quad y_i = -1$$

$$\xi_i \geq 0 \qquad \forall i$$

Support Vector Machines

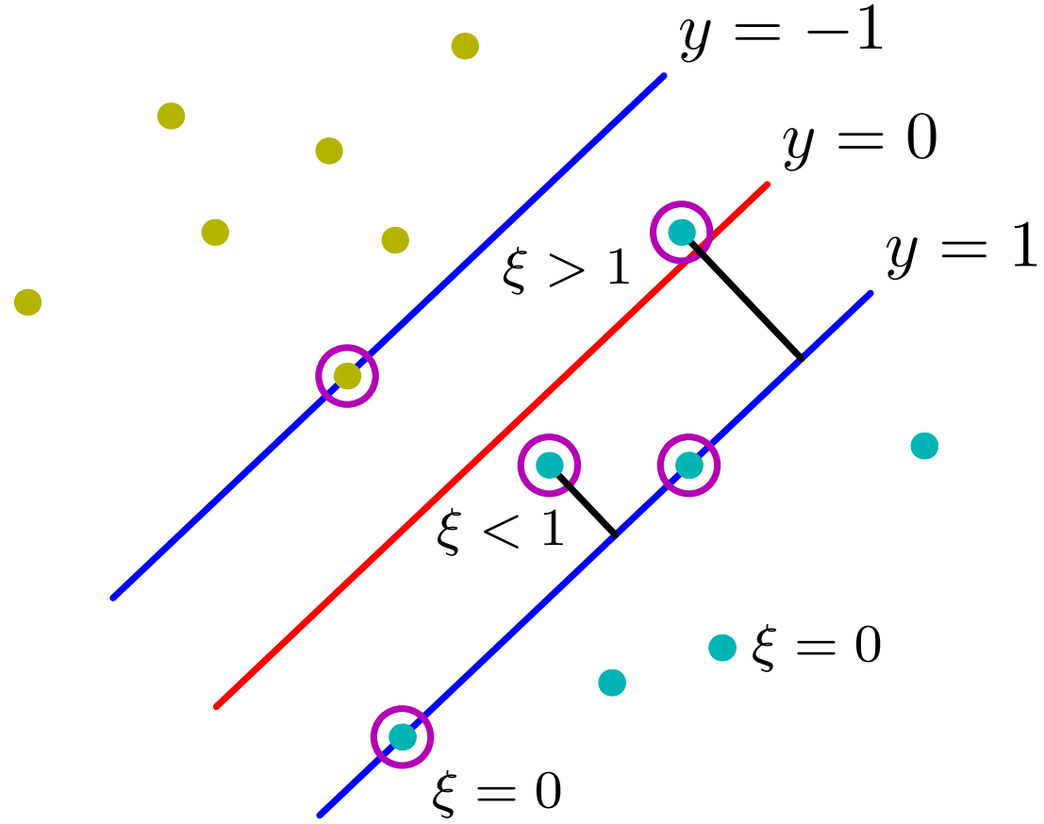
TECHNISCHE UNIVERSITÄT DARMSTADT

- More concisely, we require that:

$$y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0 \qquad \forall i$$

  - The ξ_i are called "slack variables".

- Illustration:



Support Vector Machines

TECHNISCHE UNIVERSITÄT DARMSTADT

- We have to penalize the deviations:

$$\arg\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{s.t.} \quad y_i(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b) - 1 + \xi_i \geq 0$$

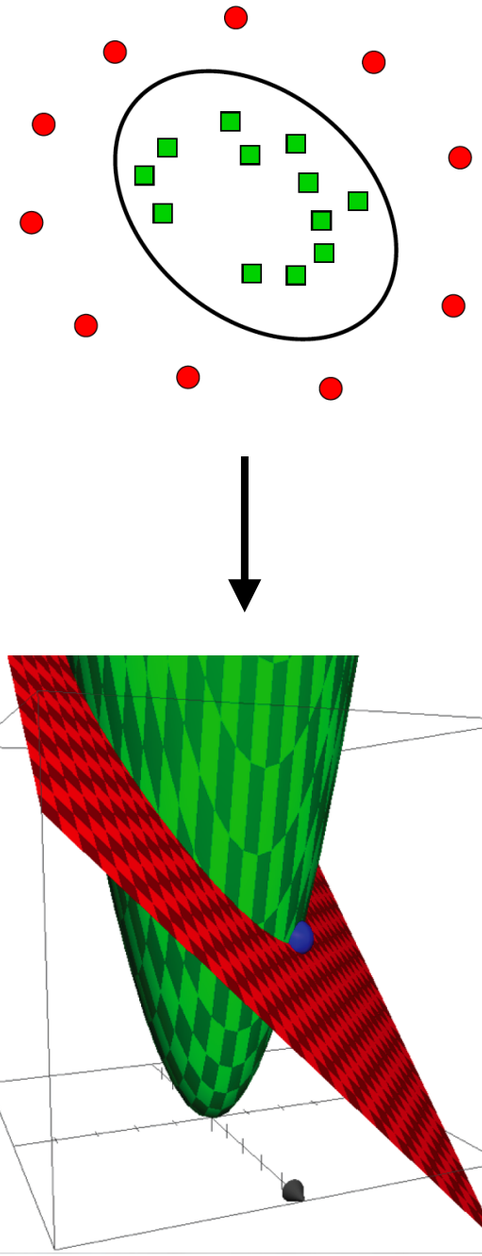
$$\xi_i \geq 0$$

- Maximize the margin while minimizing the penalty for all data points that are not outside the margin.
- The weight C allows us to specify a trade-off.
  - Typically determined through cross-validation.
- Even if the data is separable, it may be better to allow for an occasional penalty.

Nonlinear Separation of Data

TECHNISCHE UNIVERSITÄT DARMSTADT

- Example:
  - Linear separation in 2D not possible
  - Linear separation in 3D however possible
  - Nonlinear transformation from 2D to 3D...



$$\begin{aligned} \Phi: \quad \mathbb{R}^2 &\rightarrow \mathbb{R}^3 \\ (g_1, g_2) &\mapsto (z_1, z_2, z_3) := (g_1^2, \sqrt{2}g_1g_2, g_2^2) \end{aligned}$$

Nonlinear SVMs

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nonlinear transformation ϕ of the data:

$$\mathbf{x} \in \mathbb{R}^d \qquad \phi : \mathbb{R}^d \rightarrow H$$

- Hyperplane in H (linear classifier in H).

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}} \phi(\mathbf{x}) + b$$

- Nonlinear classifier in $\mathbb{R}^d$.
- Remember:
  - The hyperplane can be written in terms of the input features

$$\mathbf{w} = \sum_{i=1}^{N_S} \alpha_i y_i \phi(\mathbf{x}_i)$$

Nonlinear SVMs with Kernels

TECHNISCHE UNIVERSITÄT DARMSTADT

- Nonlinear discriminant function:

$$y(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i y_i \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}) + b$$

  - Still maximize the margin of the classifier!
    - Built-in generalization properties.
- Kernel "trick":
  - Replace the scalar product of non-linear features with a kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^{\mathrm{T}} \phi(\mathbf{x}_j)$$

  - Why?

Radial Basis Function Kernel

TECHNISCHE UNIVERSITÄT DARMSTADT

- Gaussian radial basis function (RBF) kernel:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{2\sigma^2}\right)$$

  - Measures similarities.

- Interesting property: H is infinite dimensional.
  - Intuition: $$\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^n}{n!} + \ldots$$
  - Since we only use the kernel function, this is no problem.

Histogram Kernels

TECHNISCHE UNIVERSITÄT DARMSTADT

- For bags of visual words there are even better kernels.
  - Specialized to histograms.

- Chi-Square kernel

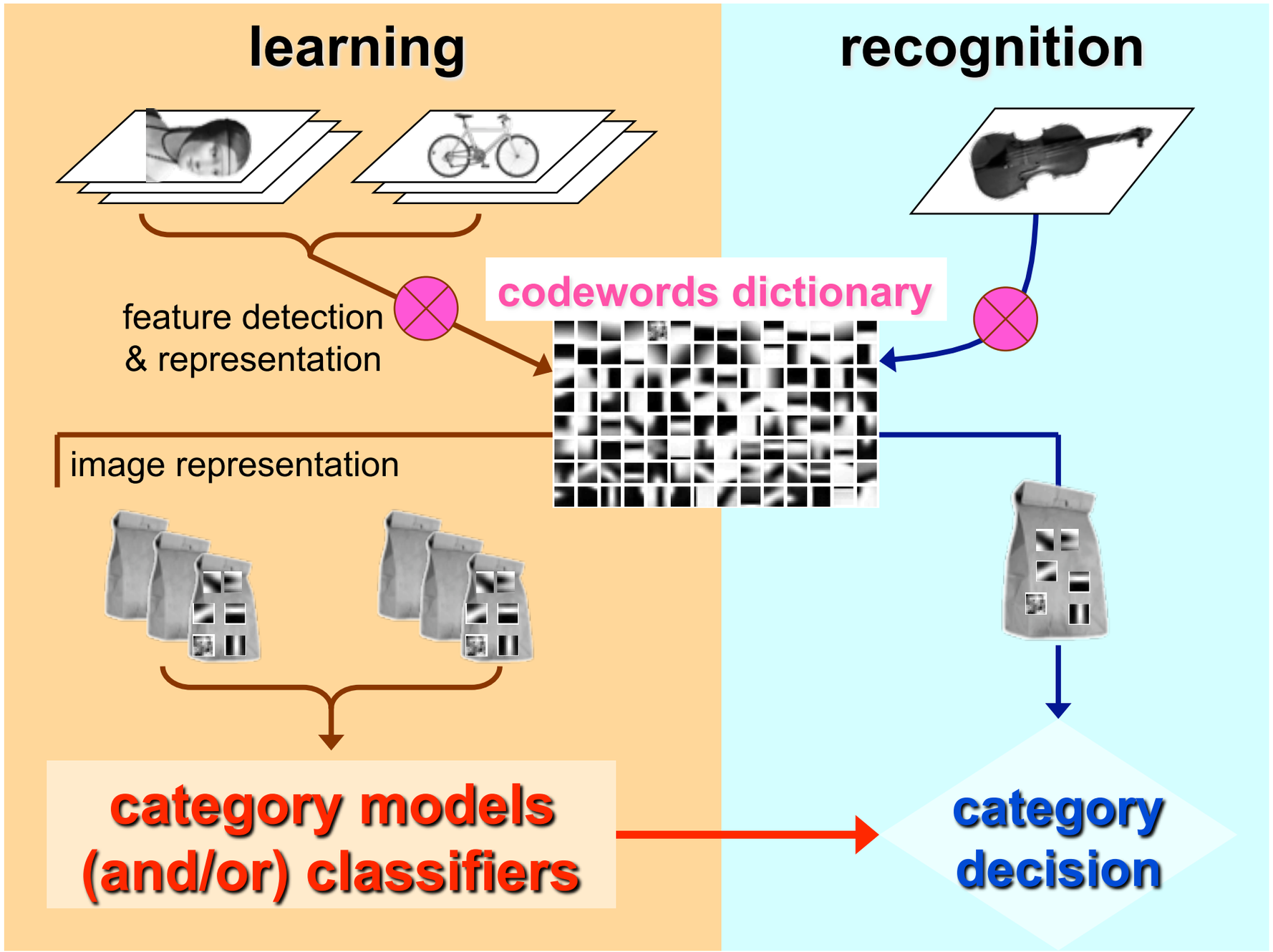
$$K(x,y) = 1 - \sum_{i=1}^{d} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)}$$

- Histogram intersection kernel

$$K(x,y) = \sum_{i=1}^{d} \min(x_i, y_i)$$

Bag-of-Words Model: Done!

TECHNISCHE UNIVERSITÄT DARMSTADT



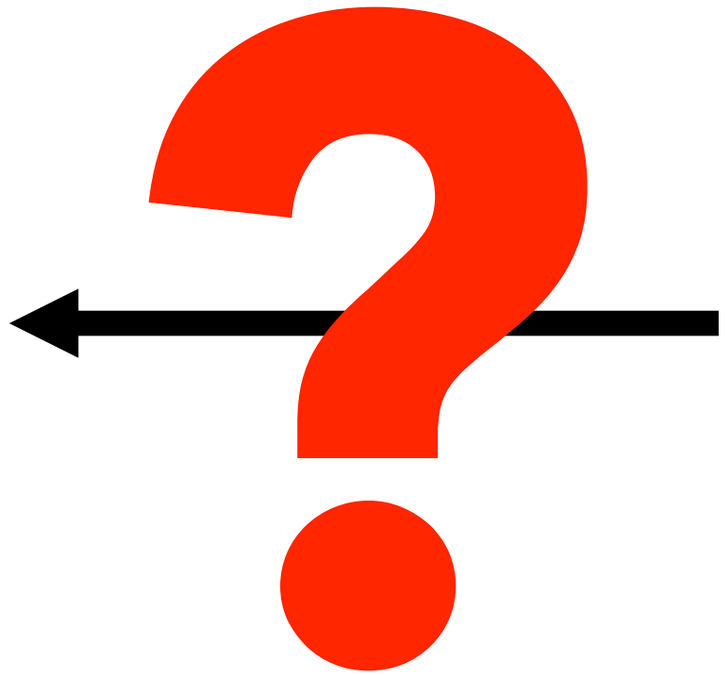
Summary & Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bag-of-words representation:
  - Sparse representation of object category
  - Many machine learning methods are directly applicable.
  - Robust to occlusions
  - Allows sharing of representation between multiple classes
- But there are also problems...

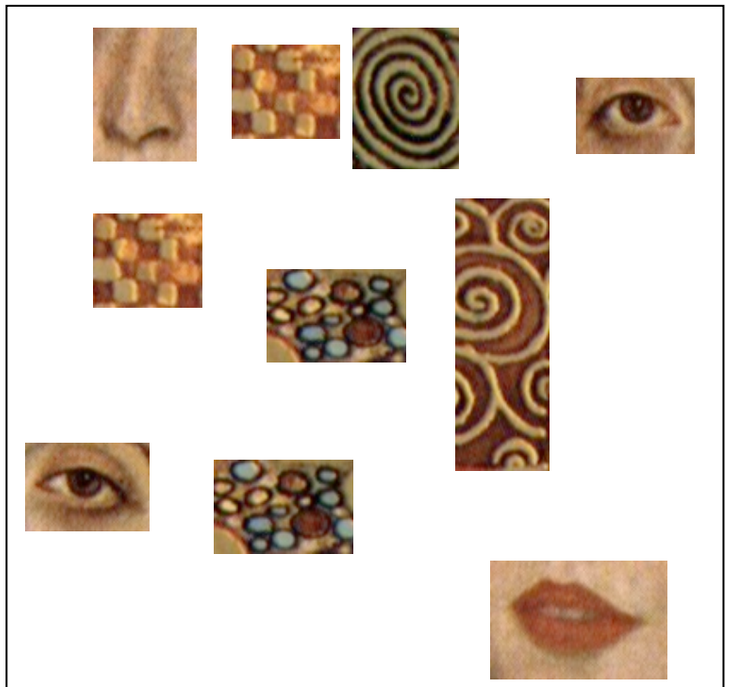
What about spatial information?

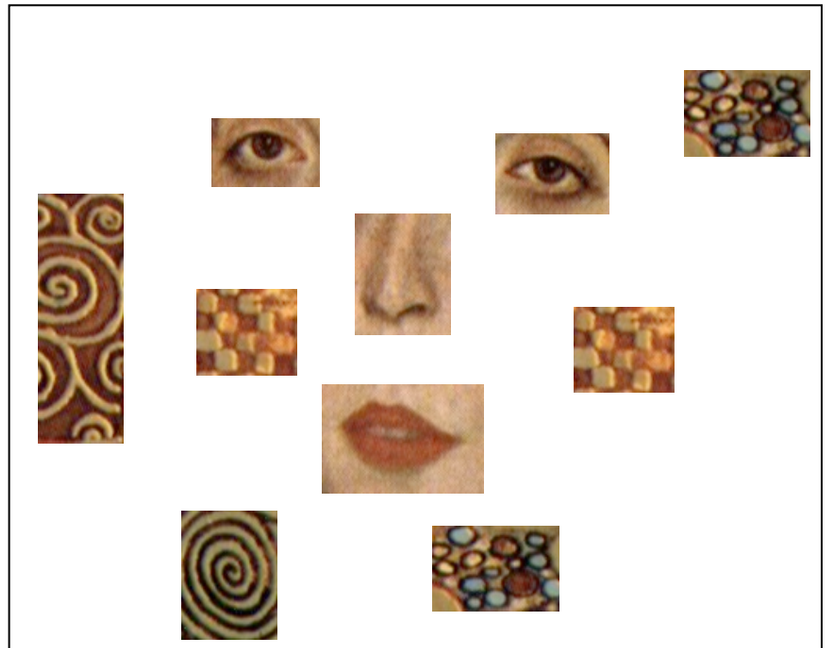
TECHNISCHE UNIVERSITÄT DARMSTADT



Problem with bag-of-words

TECHNISCHE UNIVERSITÄT DARMSTADT

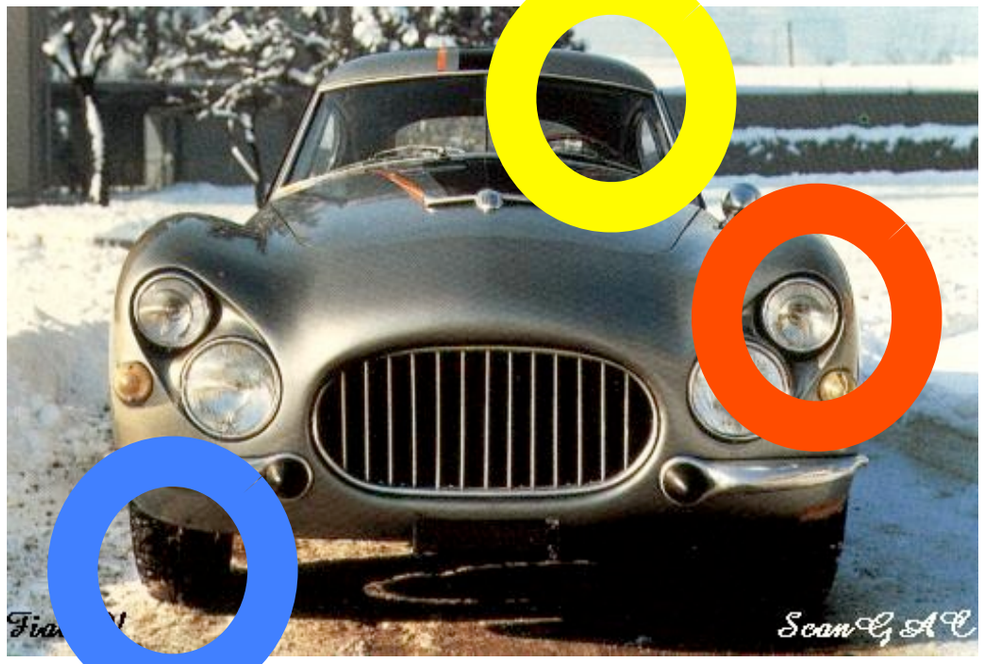




- All have equal probability for bag-of-words methods
- Location information is important!

Model: Parts and Structure

TECHNISCHE UNIVERSITÄT DARMSTADT



Part-based Representation

TECHNISCHE UNIVERSITÄT DARMSTADT

- Object as set of parts
  - Generative representation
- Model:
  - Relative locations between parts
  - Appearance of part
- Issues:
  - How to model location
  - How to represent appearance
  - How to handle occlusion/clutter

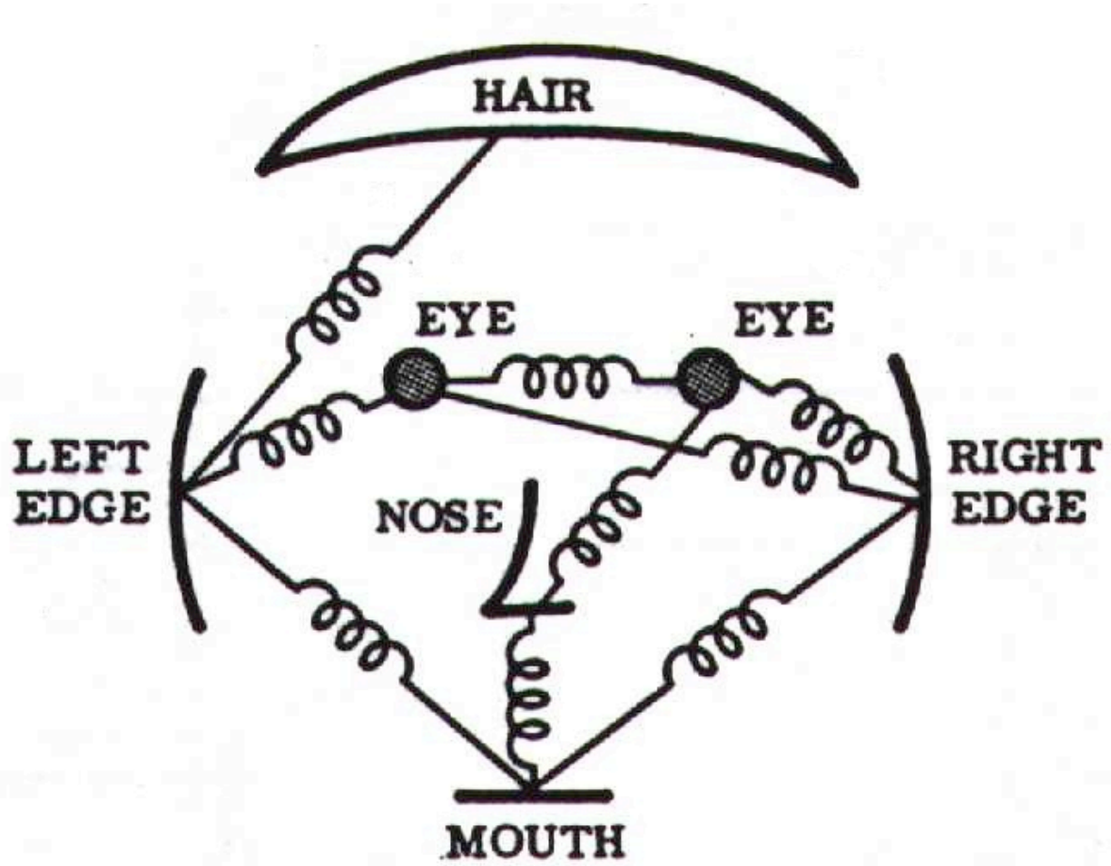
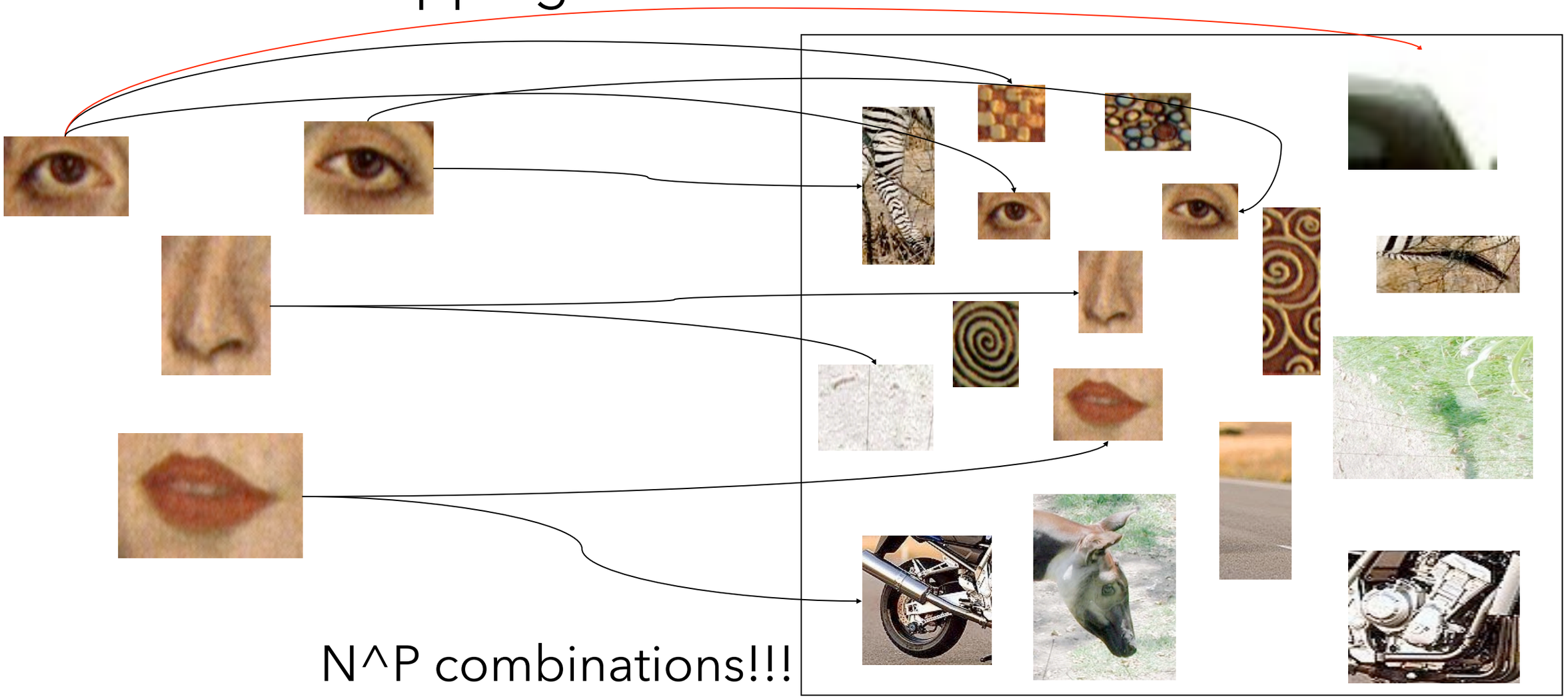


Figure from [Fischler & Elschlager 73]

The correspondence problem

TECHNISCHE UNIVERSITÄT DARMSTADT

- Model with P parts
- Image with N possible assignments for each part
- Consider mapping to be 1-1

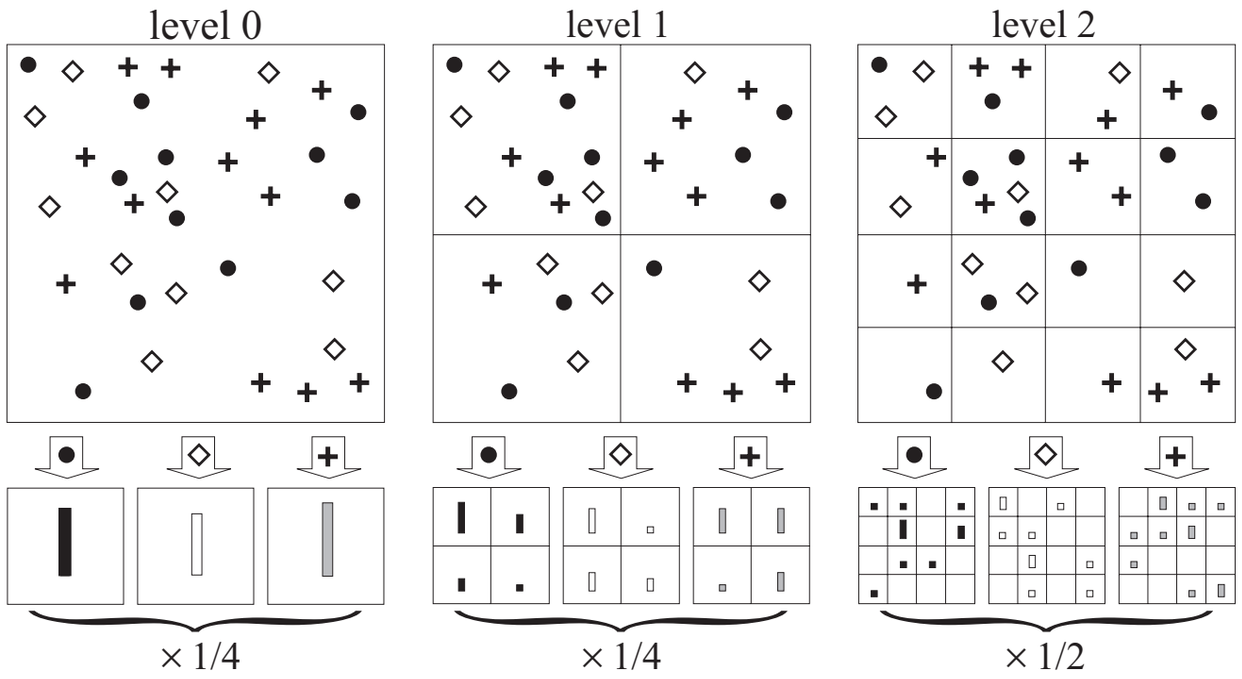


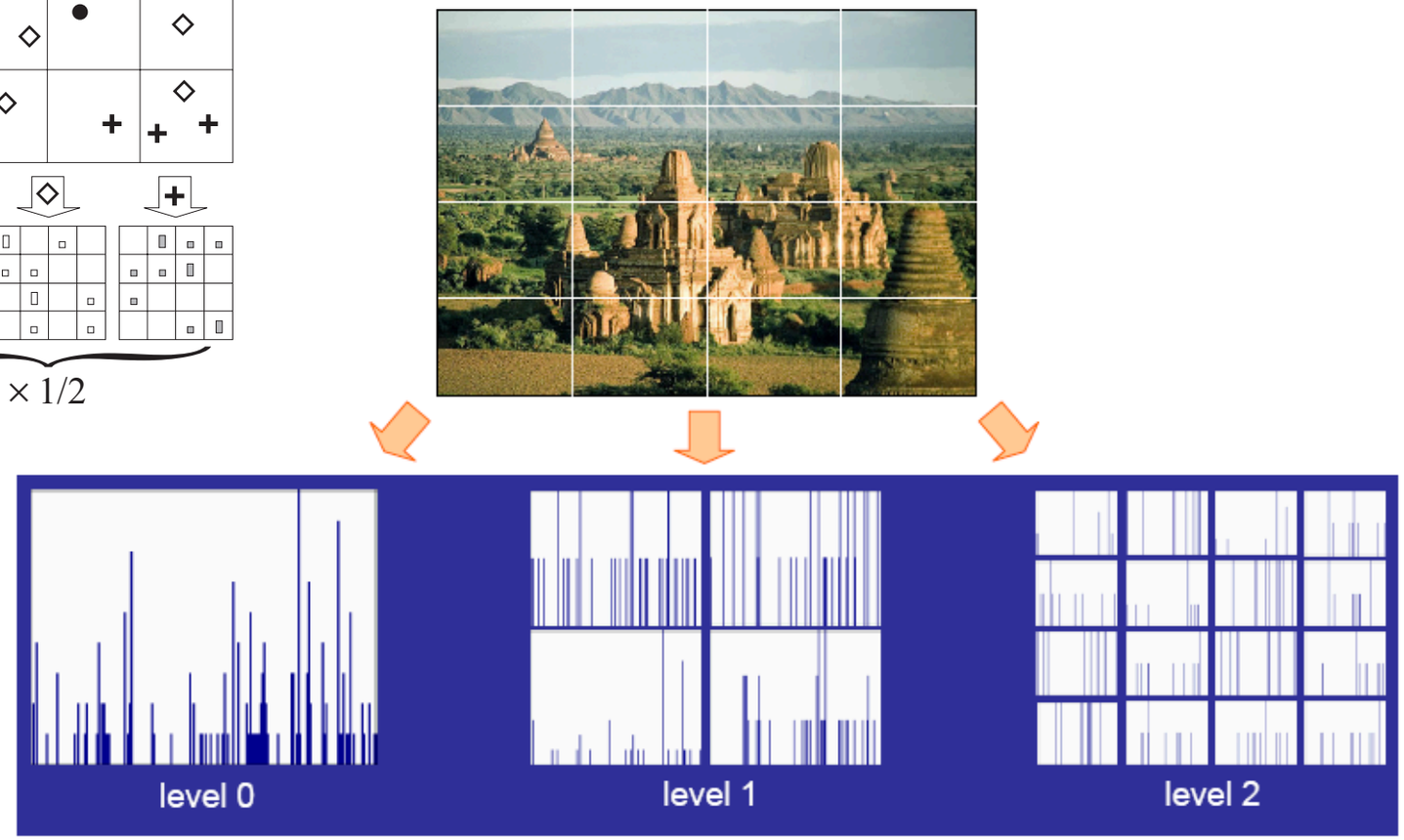
N^P combinations!!!

Pyramid match kernel

TECHNISCHE UNIVERSITÄT DARMSTADT

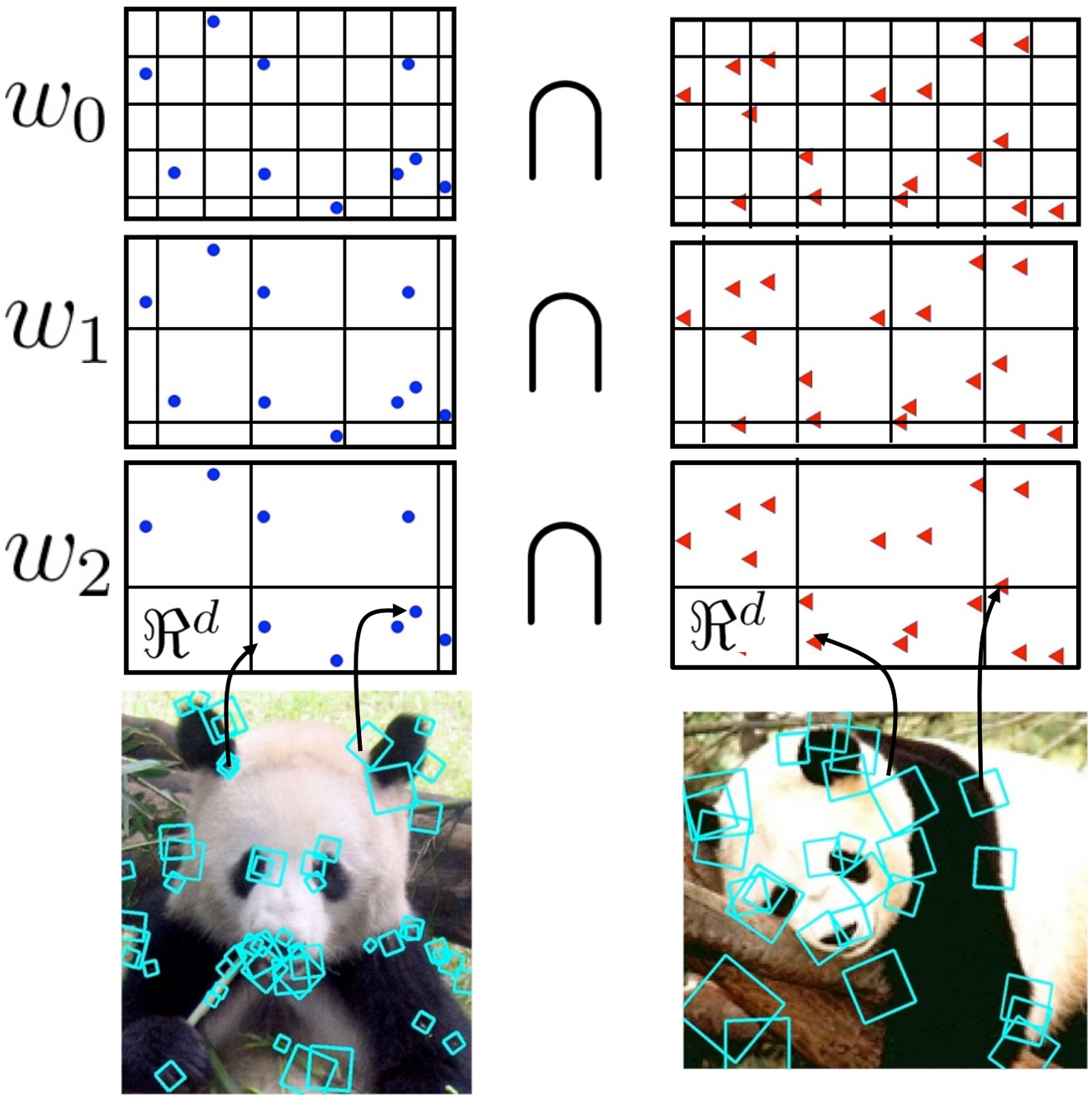
- Combine visual words with spatial information
- Can use SVMs and histogram kernels





[Lazebnik et al.]

Pyramid match kernel



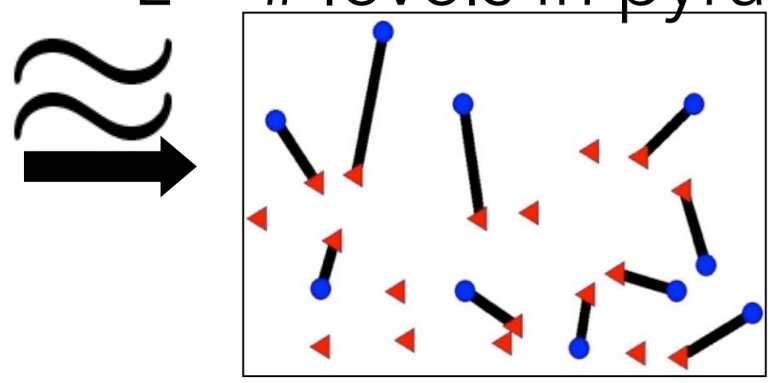
$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

Optimal match: $O(m^3)$

Pyramid match: O(mL)

m = # features

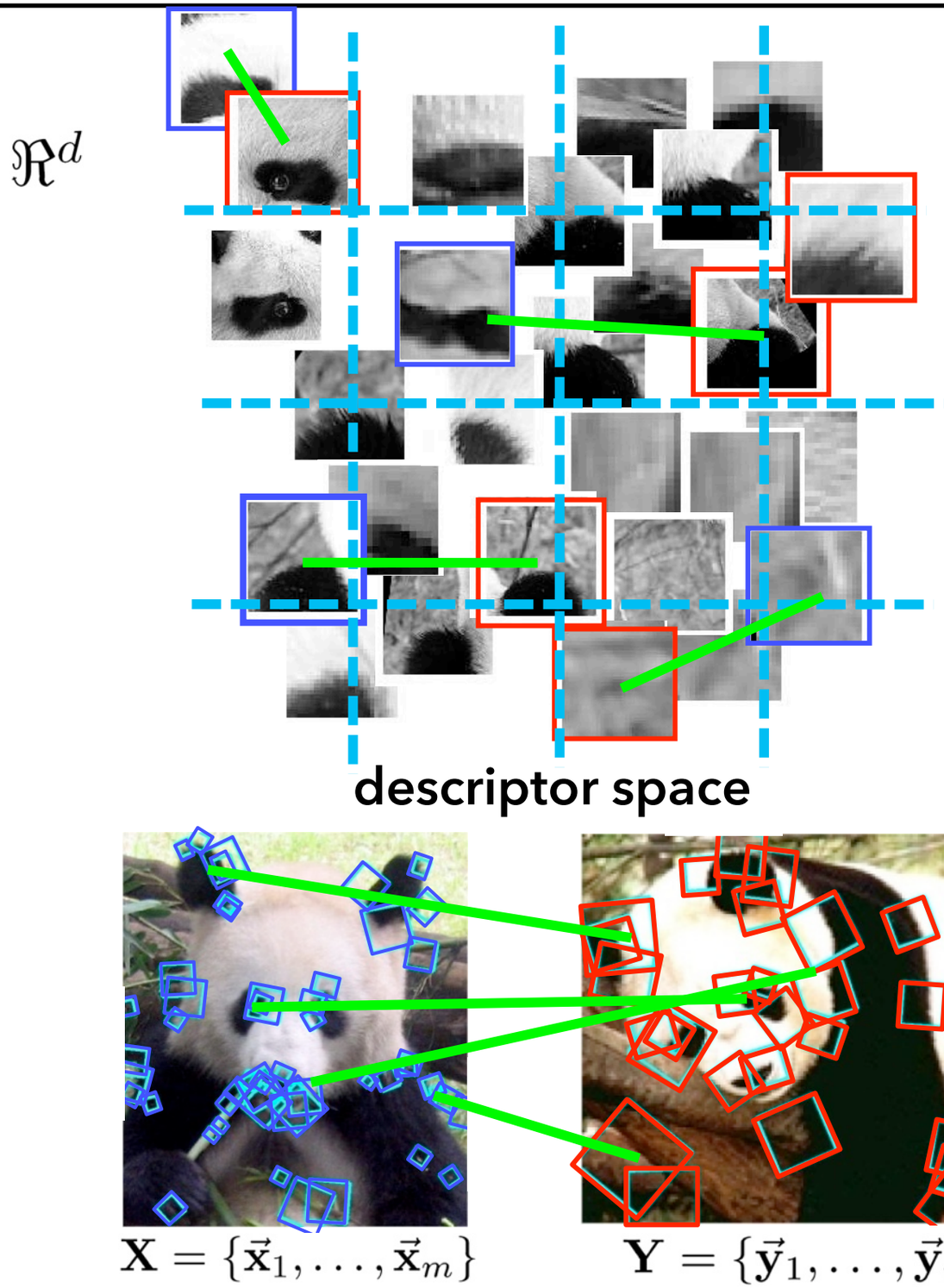
L = # levels in pyramid



optimal partial matching

[Grauman & Darrell]

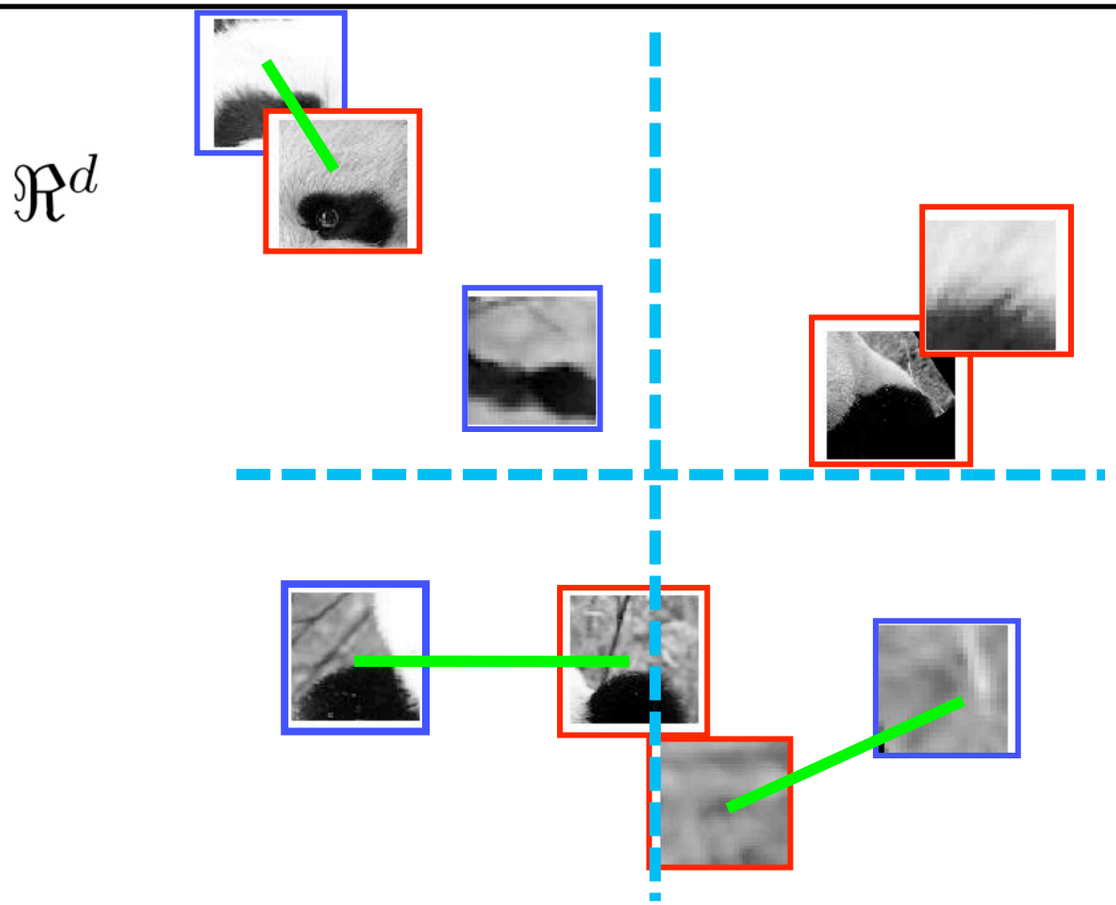
Pyramid match: main idea

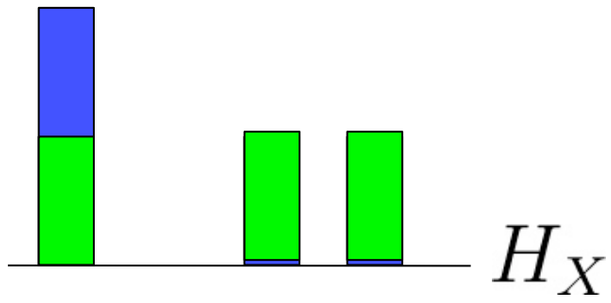


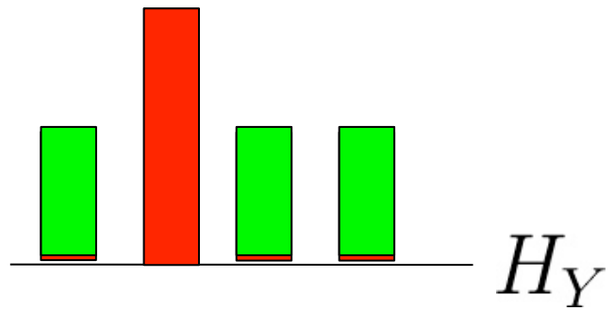
Feature space partitions serve to "match" the local descriptors within successively wider regions.

[Grauman & Darrell]

Pyramid match: main idea





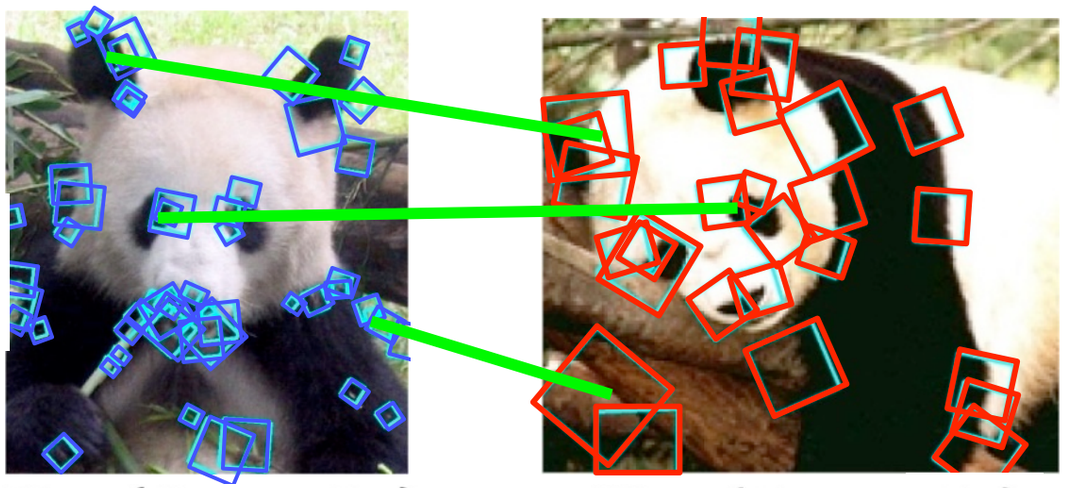


$$\mathcal{I}(H_X, H_Y) = \sum_j \min\left(H_X(j), H_Y(j)\right)$$

$$= 3$$

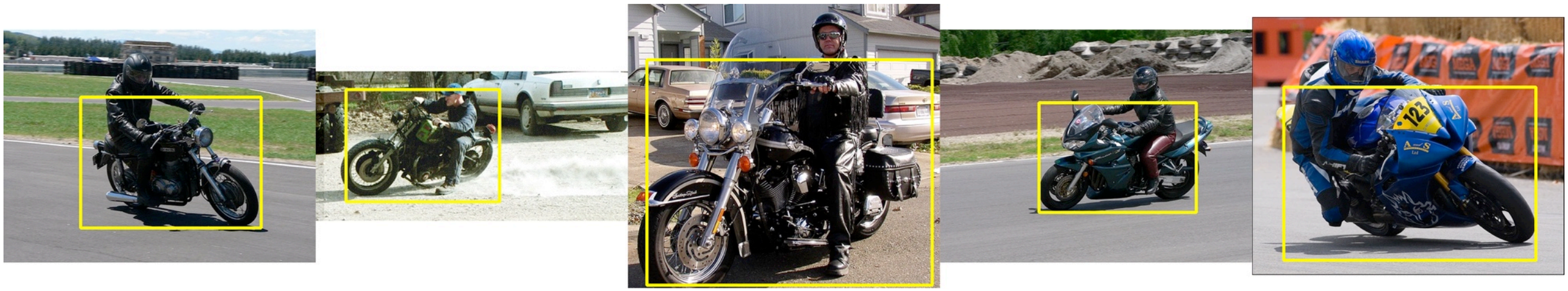
Histogram intersection counts number of possible matches at a given partitioning.

[Grauman & Darrell]



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$ $\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$

From Categorization to Detection



- We may not only want to recognize if an object is in an image, but rather find it in the image
  - Object detection / localization
- Next time:
  - Sliding window approach
  - Histogram of Oriented Gradients (HOG)
    - global descriptor for object detection