Computer Vision II - Recognition: *Image Categorization*

Michael Yang

Roadmap (3 lectures)

- Object Detection
- Scene Understanding
- Image Categorization

Roadmap (last lecture)

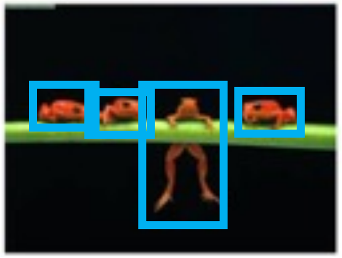
- Part Based Detector (cont. last last lecture)
  - Deformable Part Model
  - Poselets
- Scene Understanding Problem
- Context
- Spatial Layout
- 3D Scene Understanding

Class-based recognition: Level of Detail

- Image Categorization
  - One or more categories per image

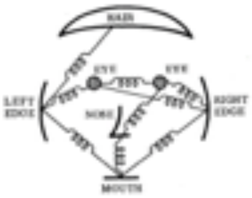
Frog, branch

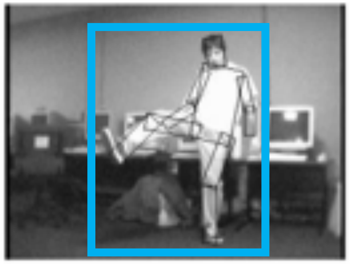
- Object Class Detection
  - Also find bounding box



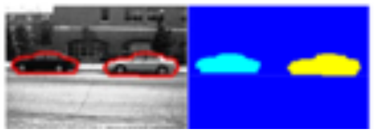
2D bounding box for each frog

- Part-based Object Detection
  - Find parts of the object
    (and in this way the full object)

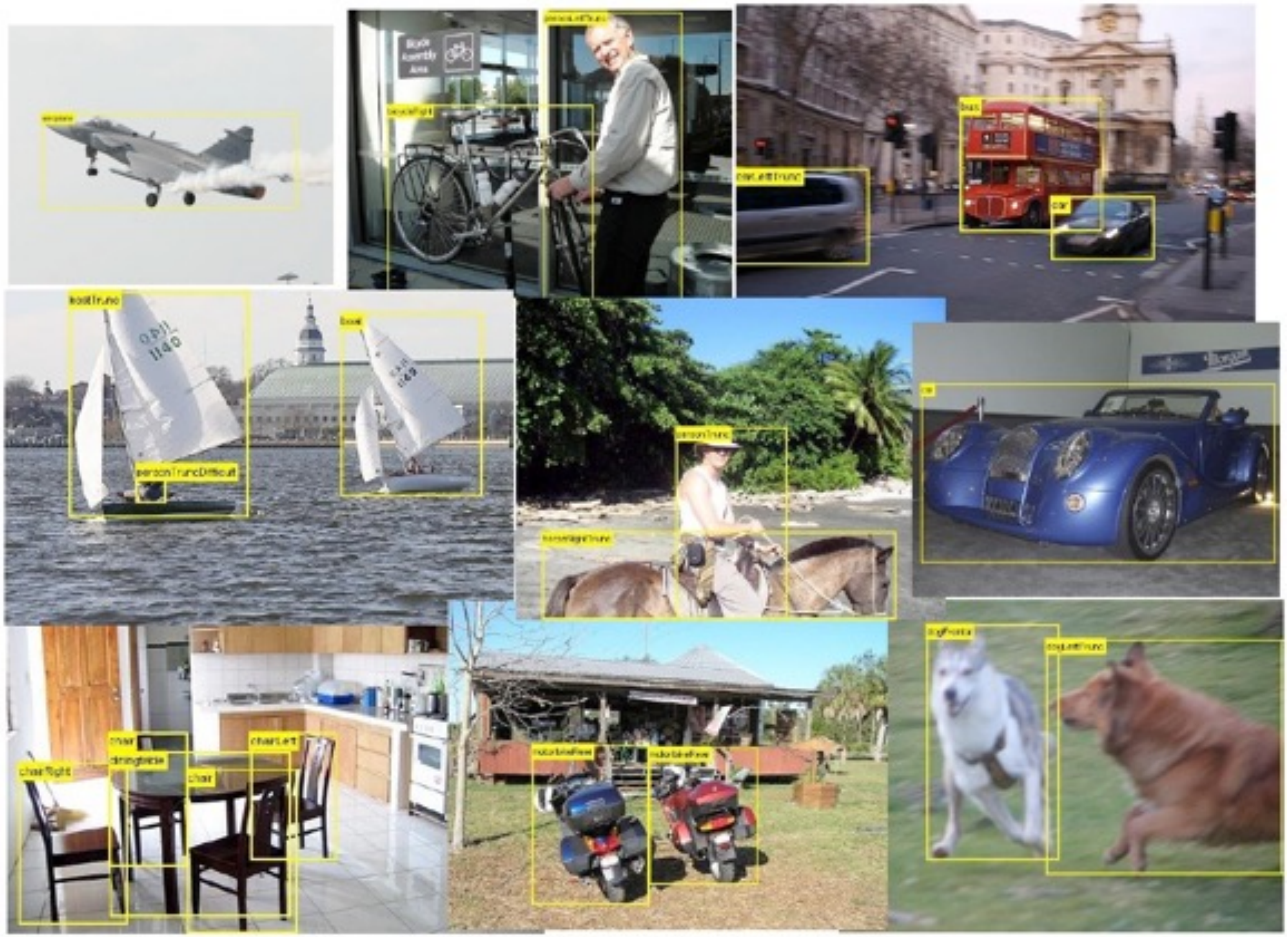




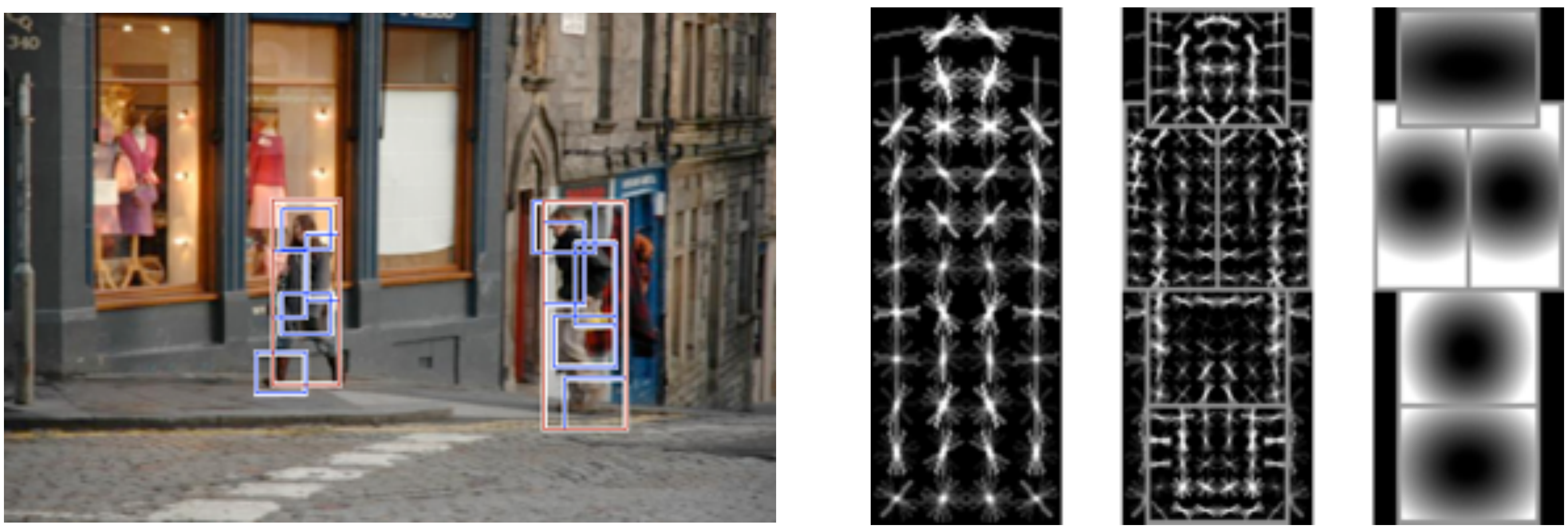
- Semantic Segmentation (see last lecture)
  (segmentation implies pixel-wise accuracy)
  - Object-class segmentation



Task: Generic object detection



DPM : Object Detection with Discriminatively Trained Part Based Models



P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010

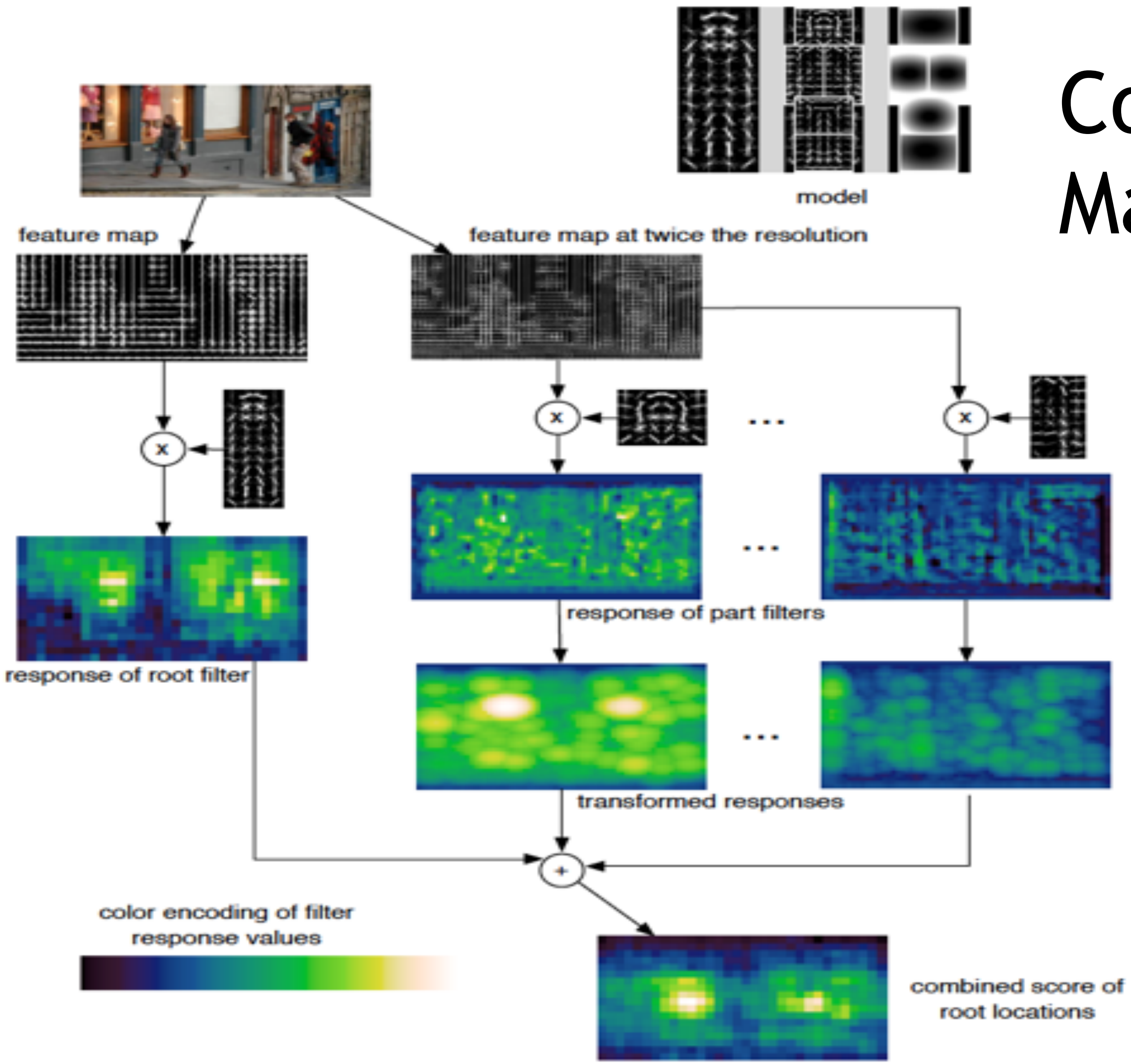
DPM: overview

- Each category detector has mixture of deformable part models (components)

- Each component has global template + deformable parts

- Fully trained from bounding boxes alone (Latent SVM)

DPM: Detection



Combine
Many Parts

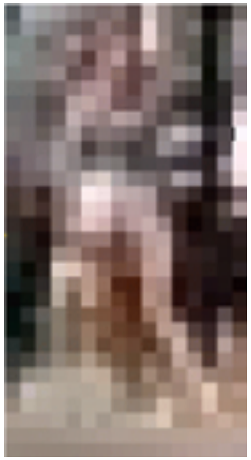
Poselet

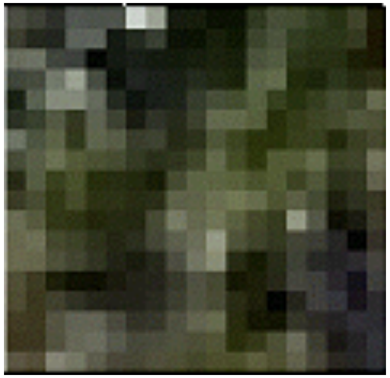
One poselet one classifier

not a model for whole human body

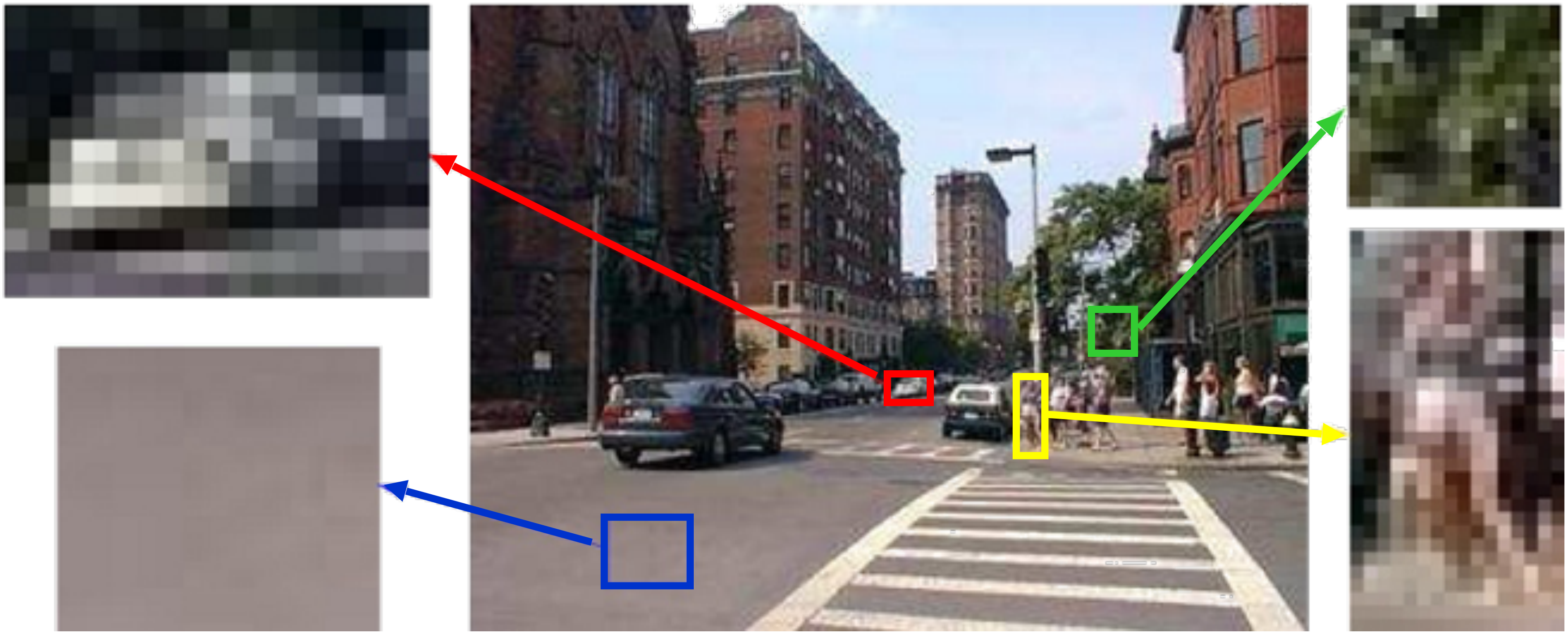
Spatial layout is especially important

1. Context for recognition





1. Context for recognition



Spatial layout is especially important

1. Context for recognition
2. Scene understanding

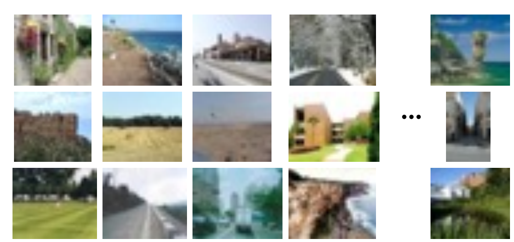
Geometry estimation as recognition

Region

Features
Color
Texture
Perspective
Position

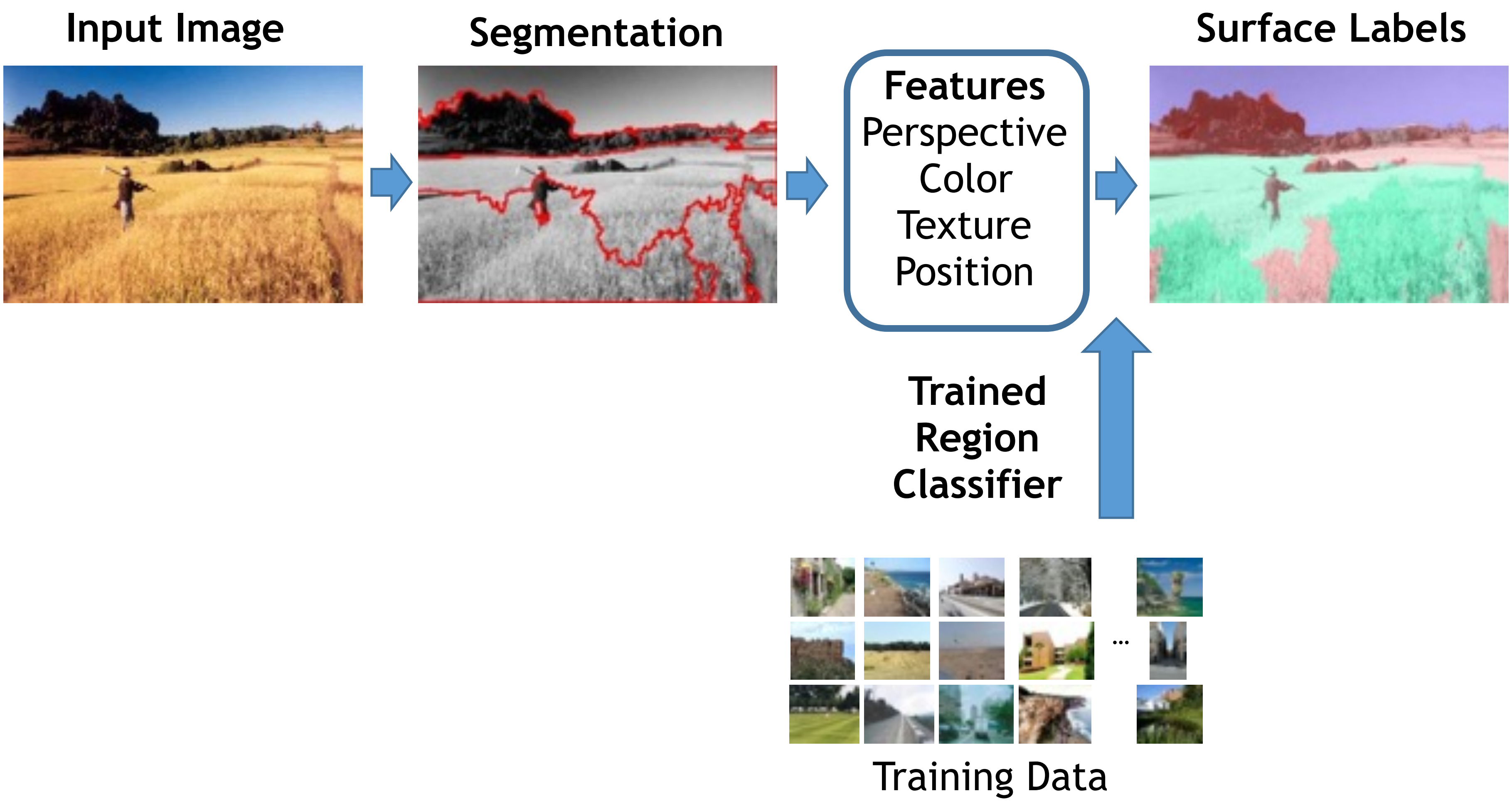
Surface Geometry Classifier

Vertical, Planar



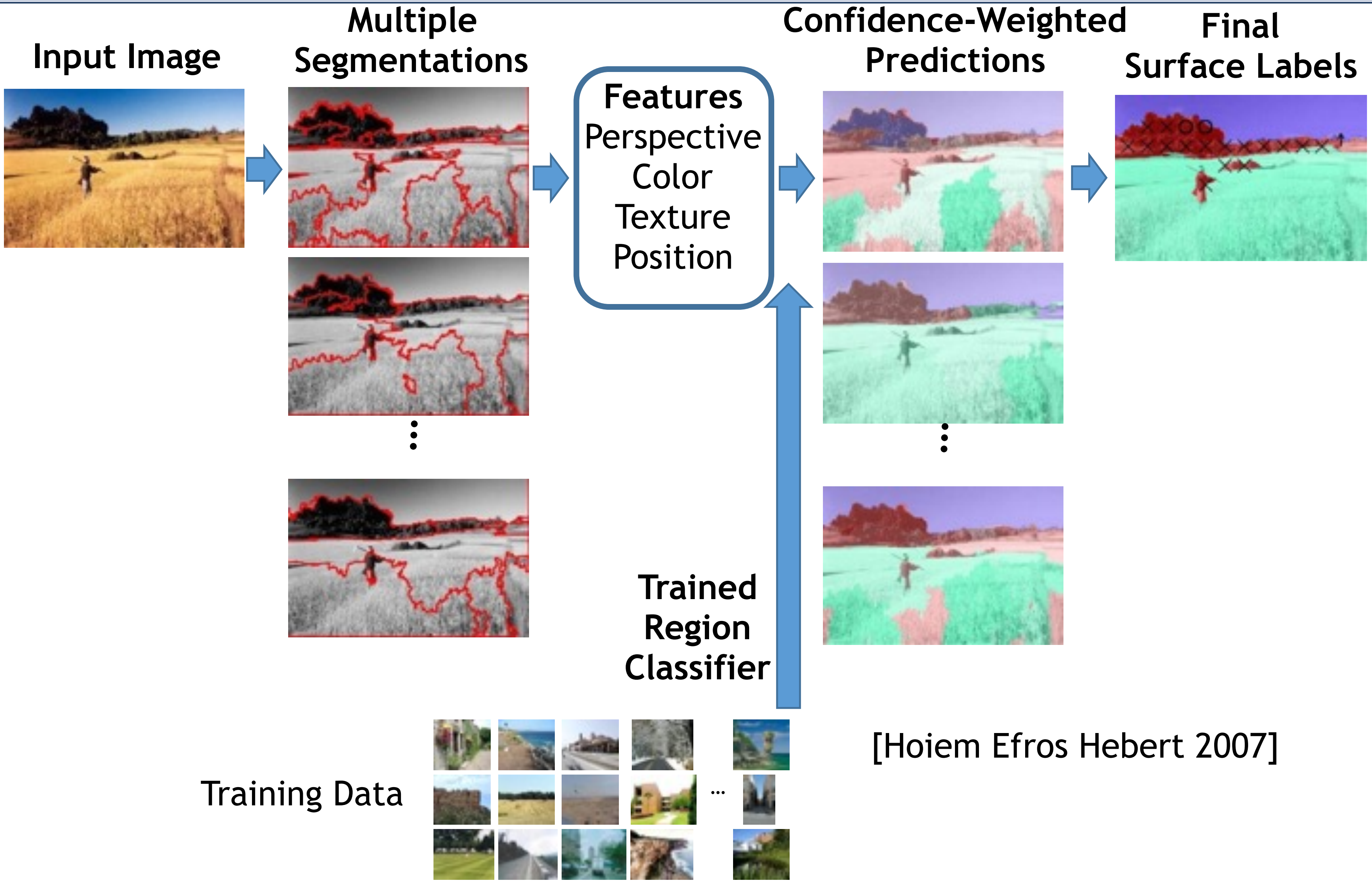
Training Data

Surface Layout Algorithm



[Hoiem Efros Hebert 2007]

Surface Layout Algorithm



Roadmap (this lecture)

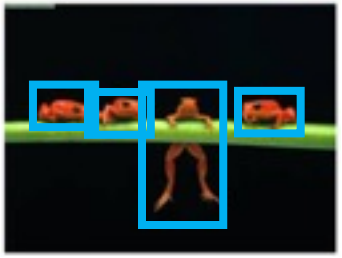
- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching

Class-based recognition: Level of Detail

- Image Categorization
  - One or more categories per image

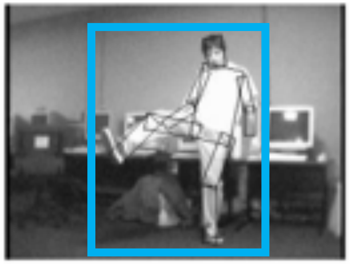
Frog (branch)

- Object Class Detection
  - Also find bounding box

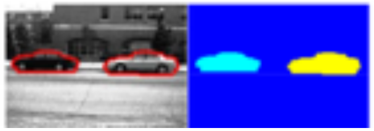


2D bounding box for each frog

- Part-based Object Detection
  - Find parts of the object
    (and in this way the full object)



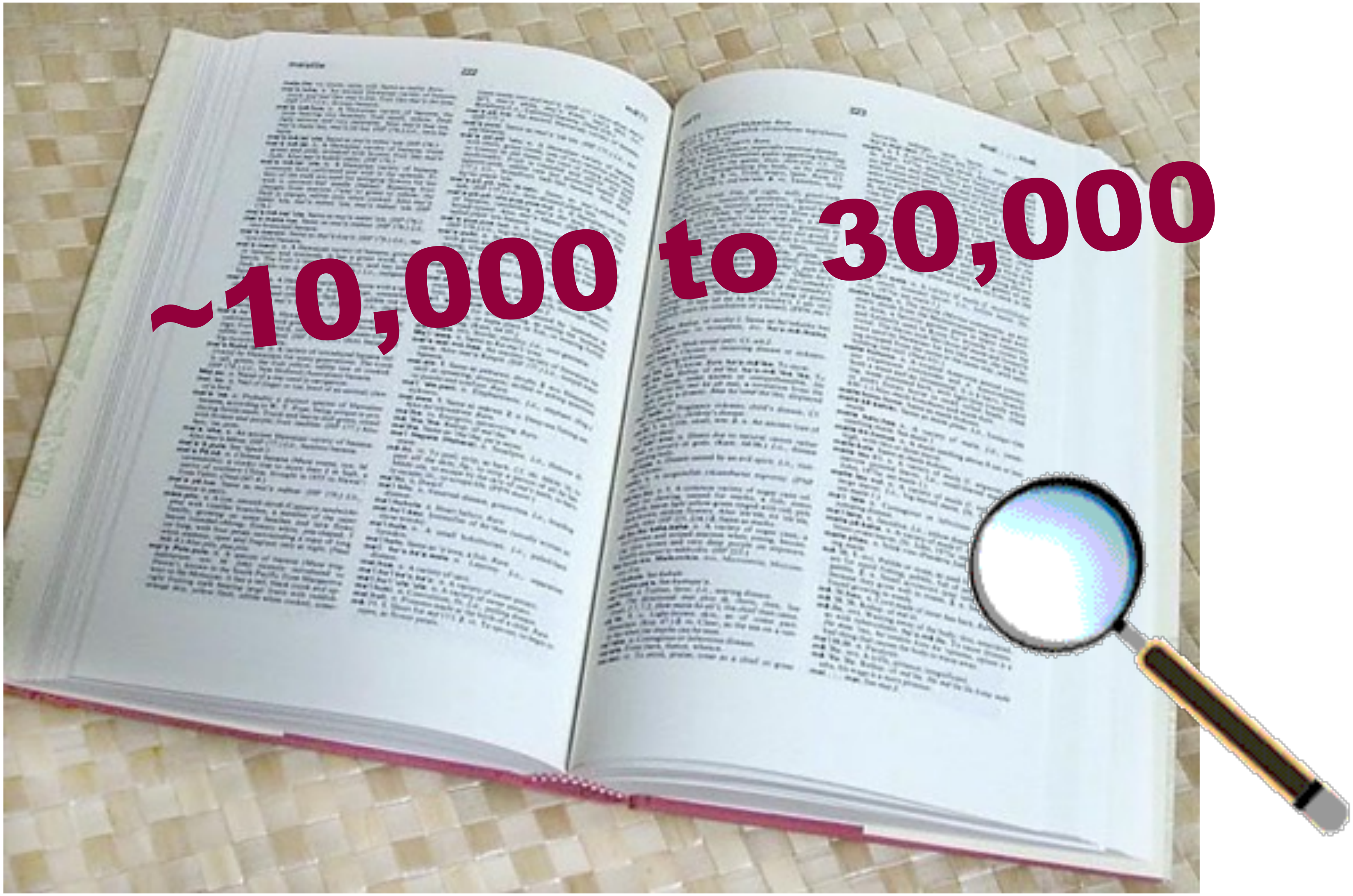
- Semantic Segmentation
  (segmentation implies pixel-wise accuracy)
  - Object-class segmentation



Why?

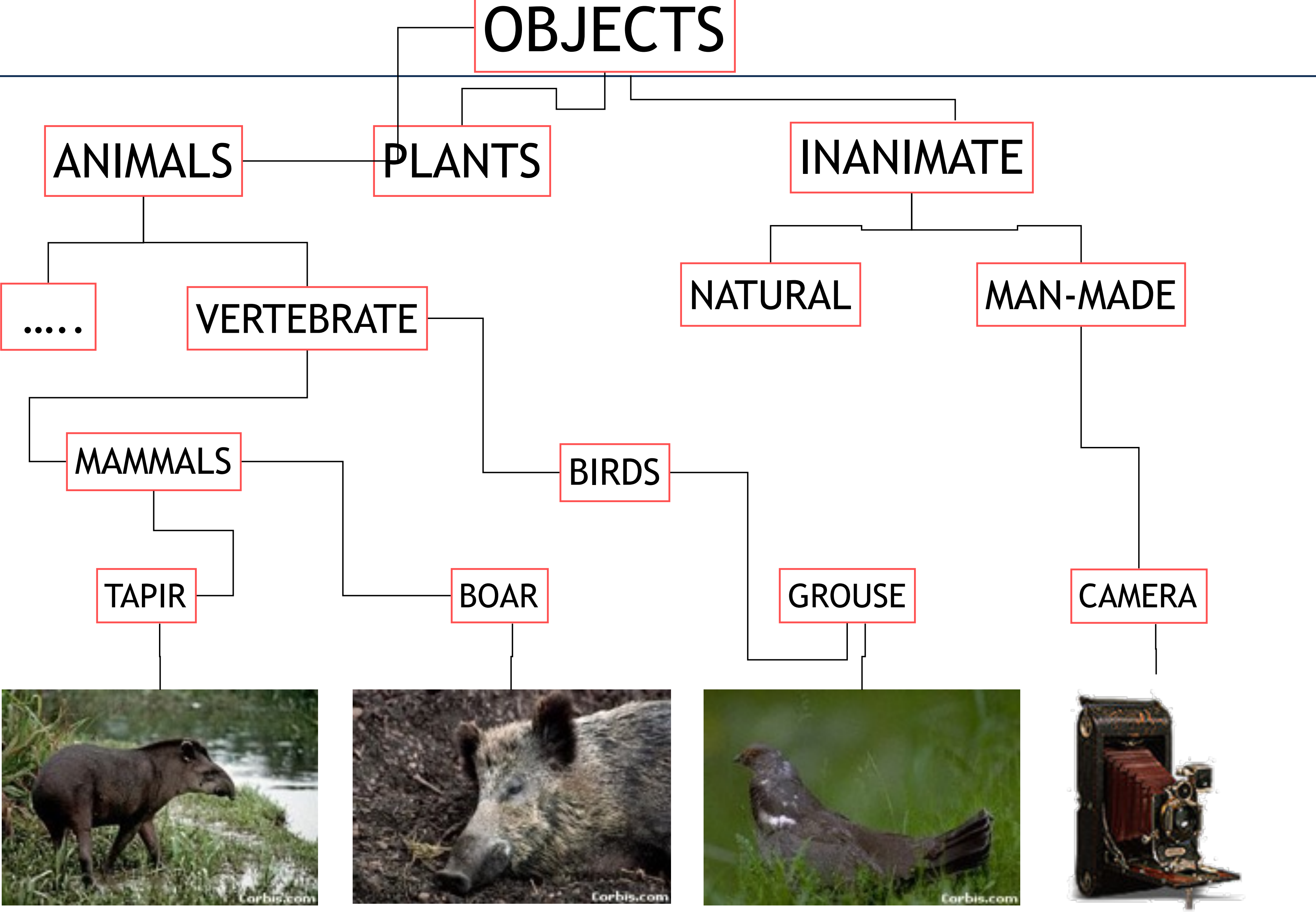
Application

How many visual object categories are there?

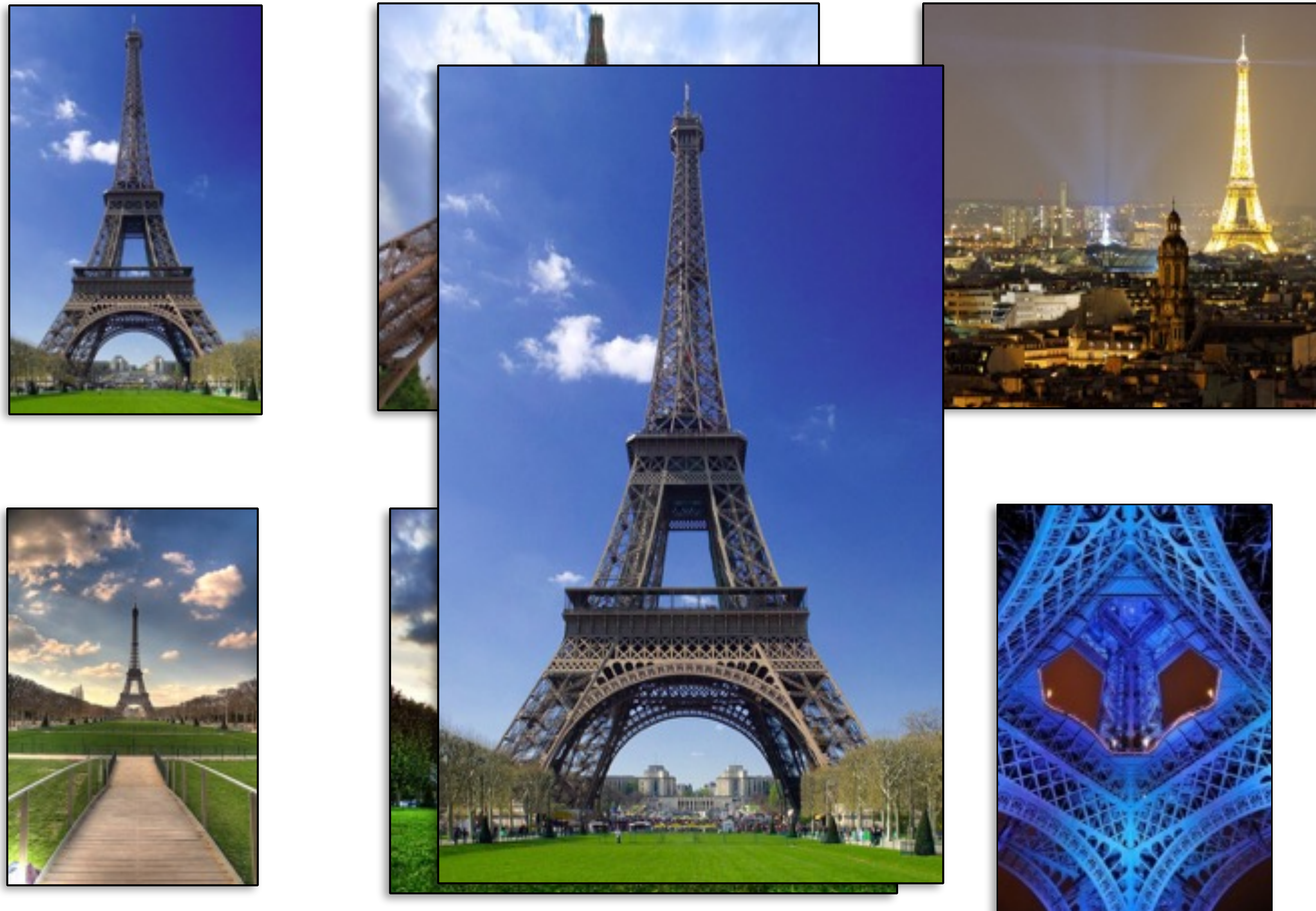


Biederman 1987


~10,000 to 30,000


OBJECTS
ANIMALS
PLANTS
INANIMATE
.....
VERTEBRATE
NATURAL
MAN-MADE
MAMMALS
BIRDS
TAPIR
BOAR
GROUSE
CAMERA
Corbis.com
Corbis.com
Corbis.com

Variation within an object class



Viewpoint/Scales/Illumination Variability

Images from Flickr

Recognition: A machine learning approach

Slides adapted from Fei-Fei Li, Rob Fergus, Antonio Torralba, Kristen Grauman, and Derek Hoiem

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = "apple"

f() = "tomato"

f() = "cow"

The machine learning framework

$$y = f(\mathbf{x})$$

output

prediction function

Image feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

Image Categorization-Steps

Training

Training Images

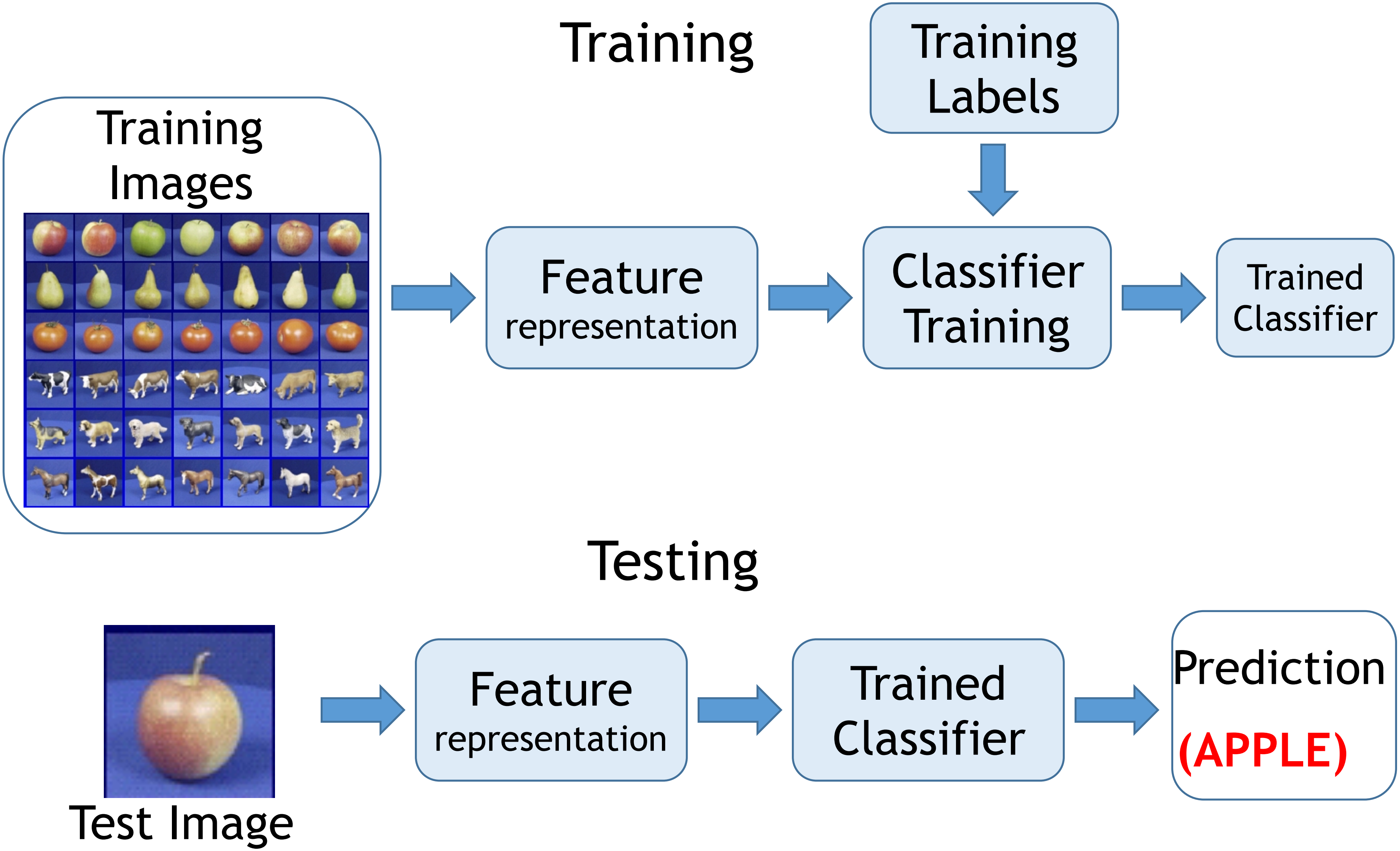
Feature representation

Training Labels

Classifier Training

Trained Classifier

Image Categorization-Steps



Generalization

Training set (labels known)

Test set (labels unknown)

- How well does a learned model *generalize* from the data it was trained on to a new test set?

Generalization depends on:

- Invariance properties of the feature representation
  - There is a tradeoff between invariance and discriminability

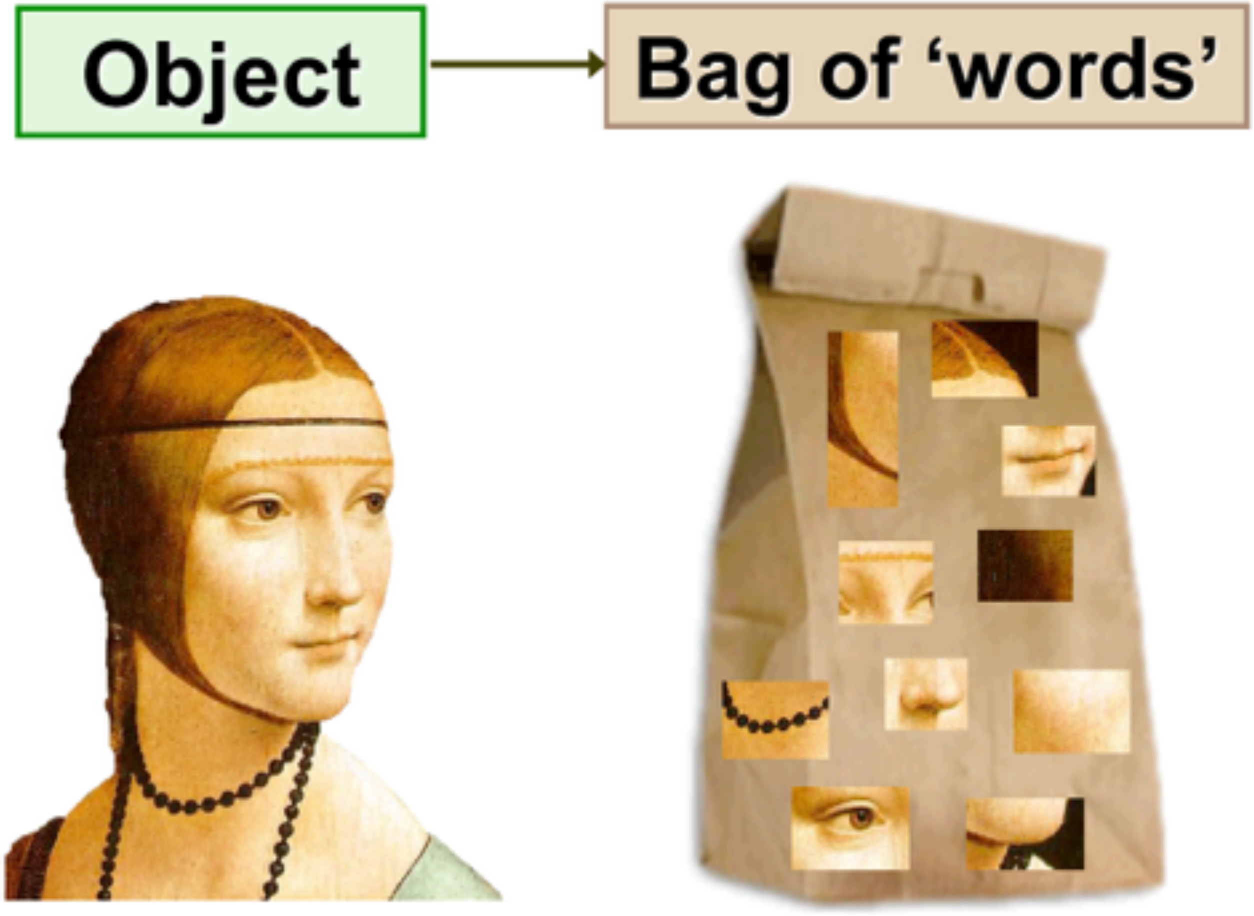
- Training data
  - Some intra-class variations must be adequately represented
    in the training data (hard to model analytically)

- Statistical model
  - Some models are more powerful than others and
    able to generalize better.

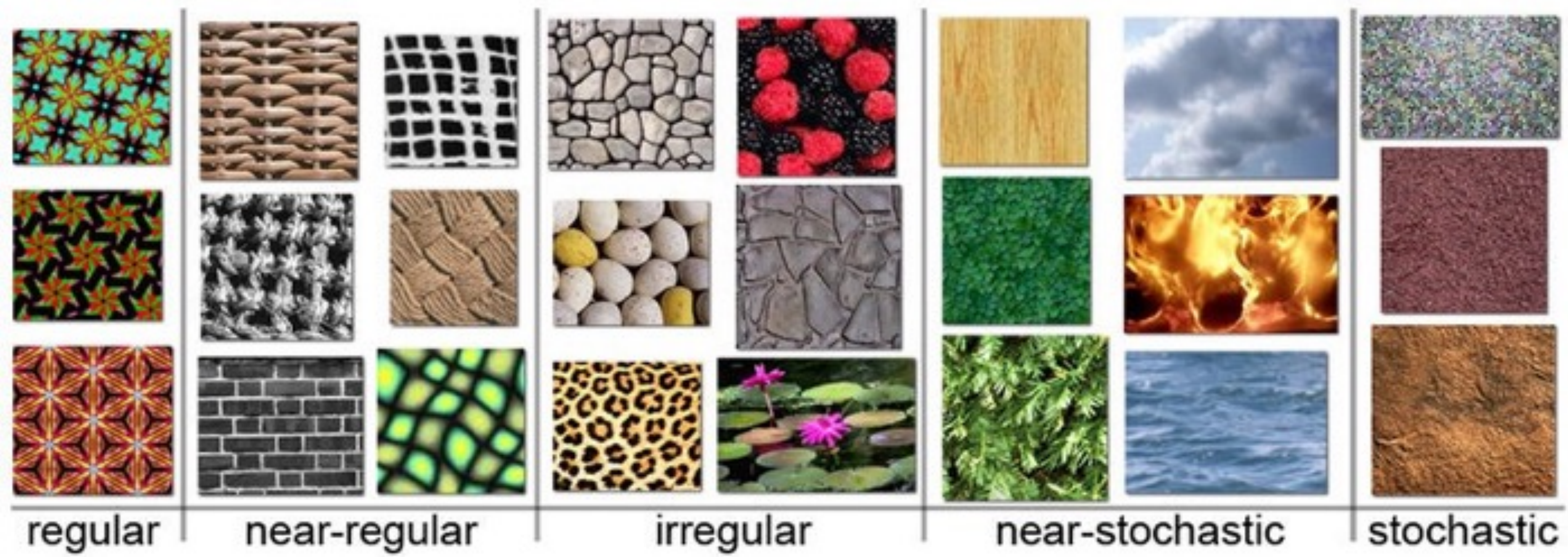
Roadmap (this lecture)

- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching

Image Categorization – Bag of Words Approach



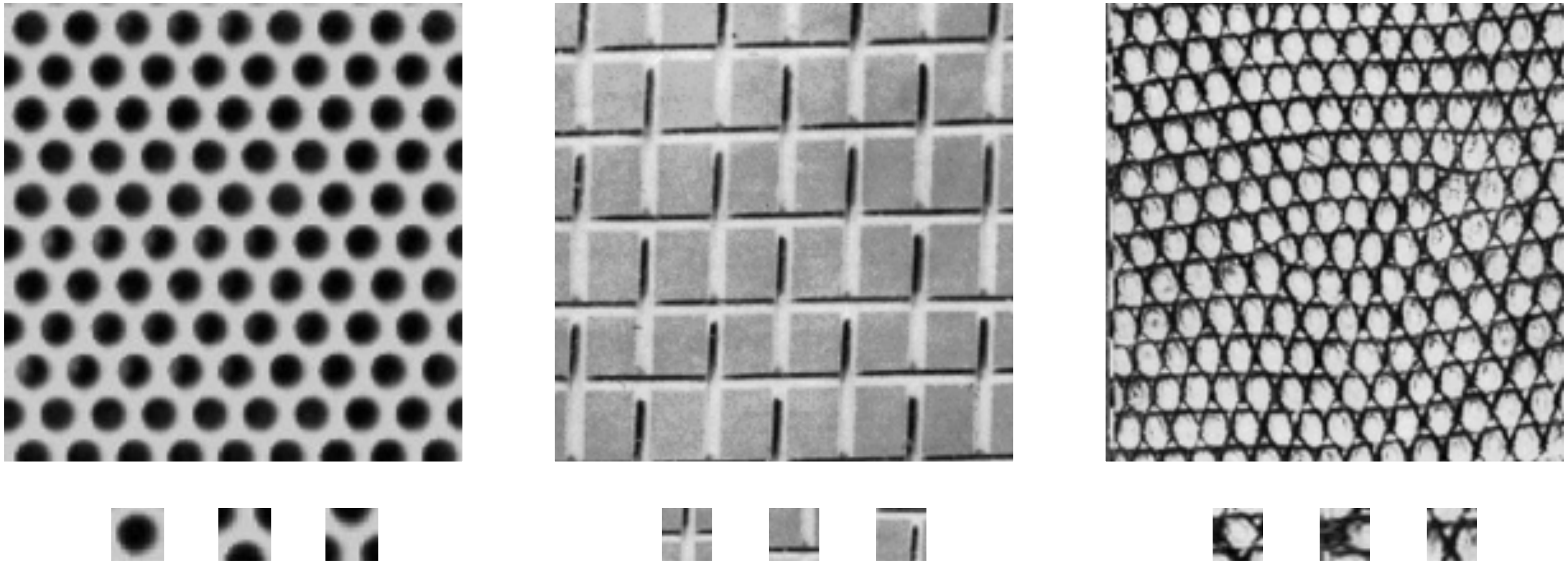
Origin 1: Texture recognition



Example textures (from Wikipedia)

Origin 1: Texture recognition

- Texture is characterized by the repetition of basic elements or *textons*
- For stochastic textures, it is the identity of the textons, not their spatial arrangement, that matters



Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

Origin 2: Bag-of-words models

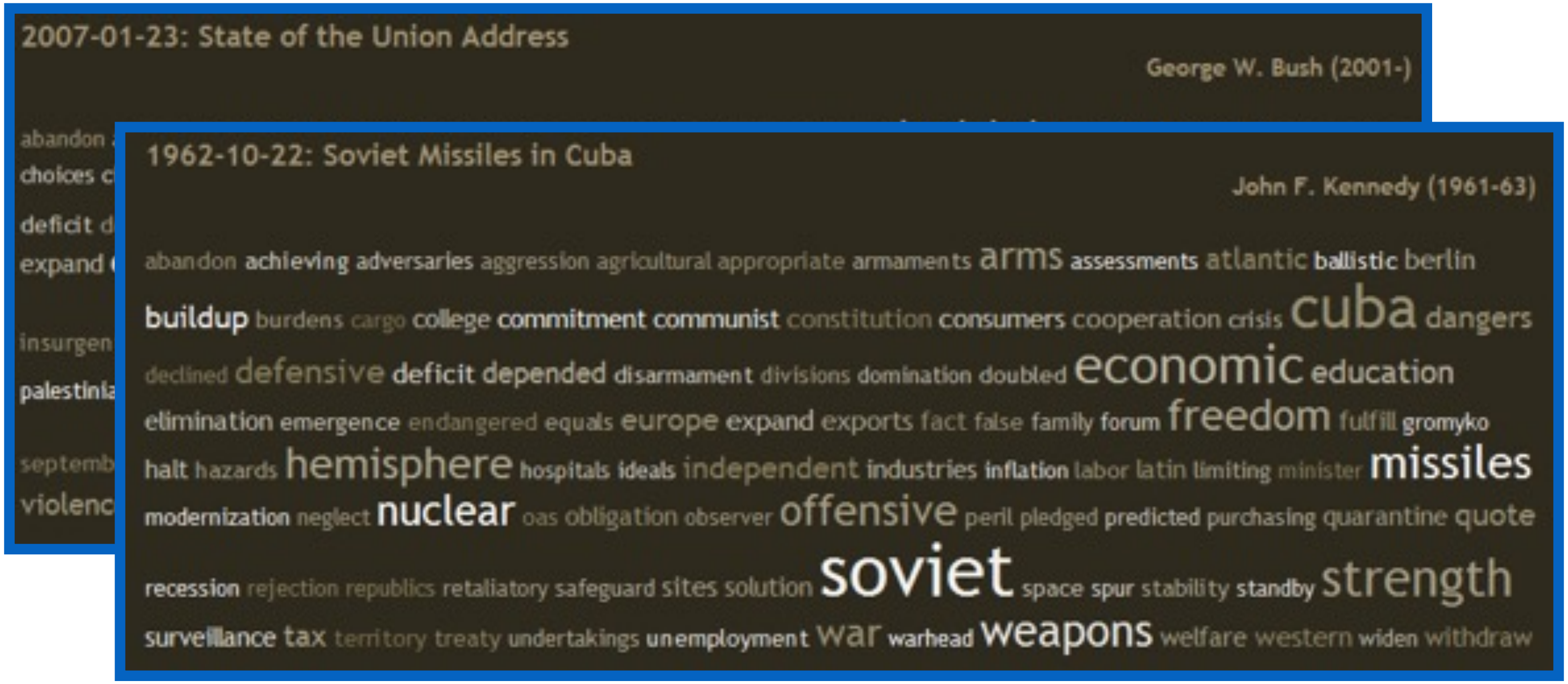
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

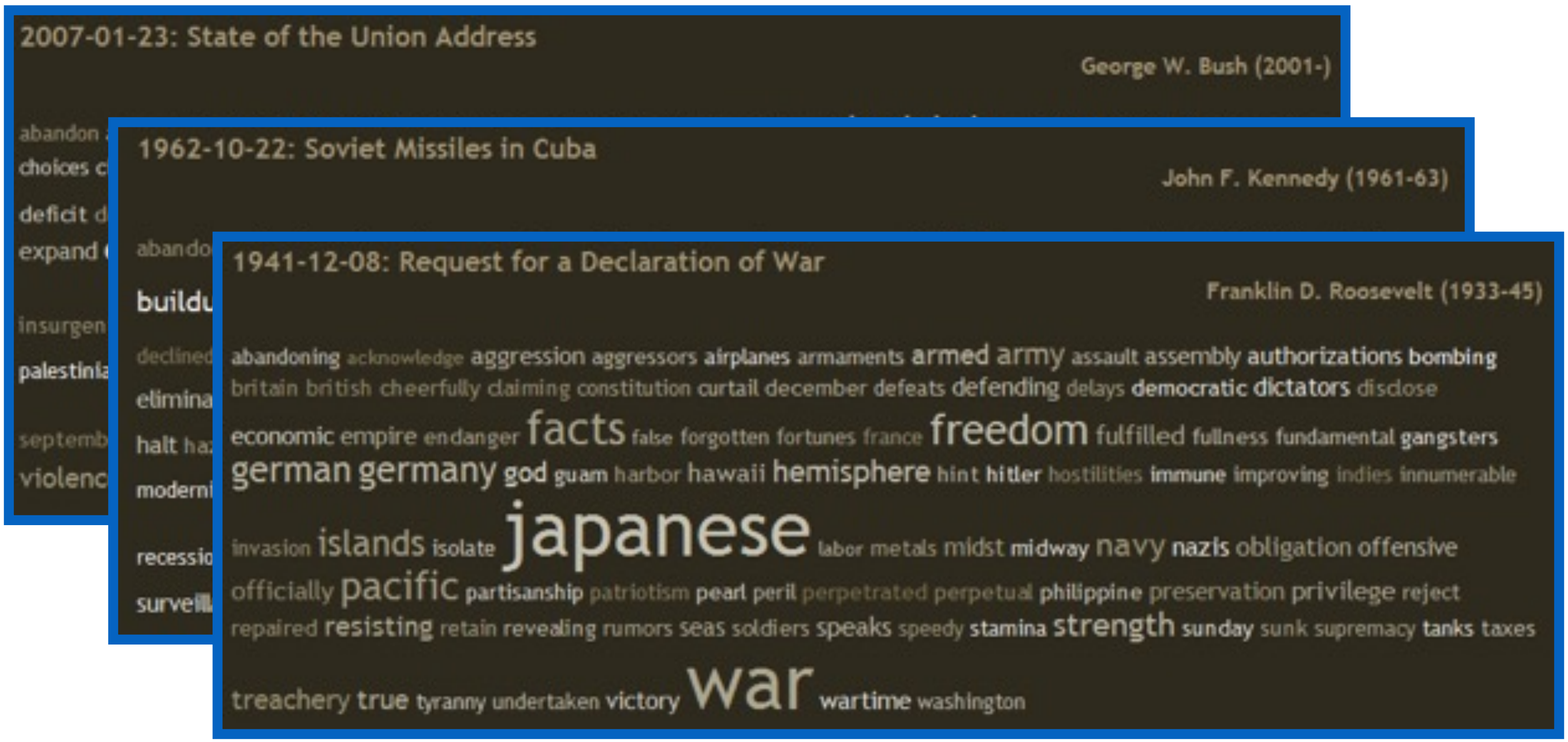
Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



Origin 2: Bag-of-words models

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)



Bags of words for object recognition

face, flowers, building

- Works pretty well for image-level classification and for recognizing object *instances*

Bags of words for object recognition

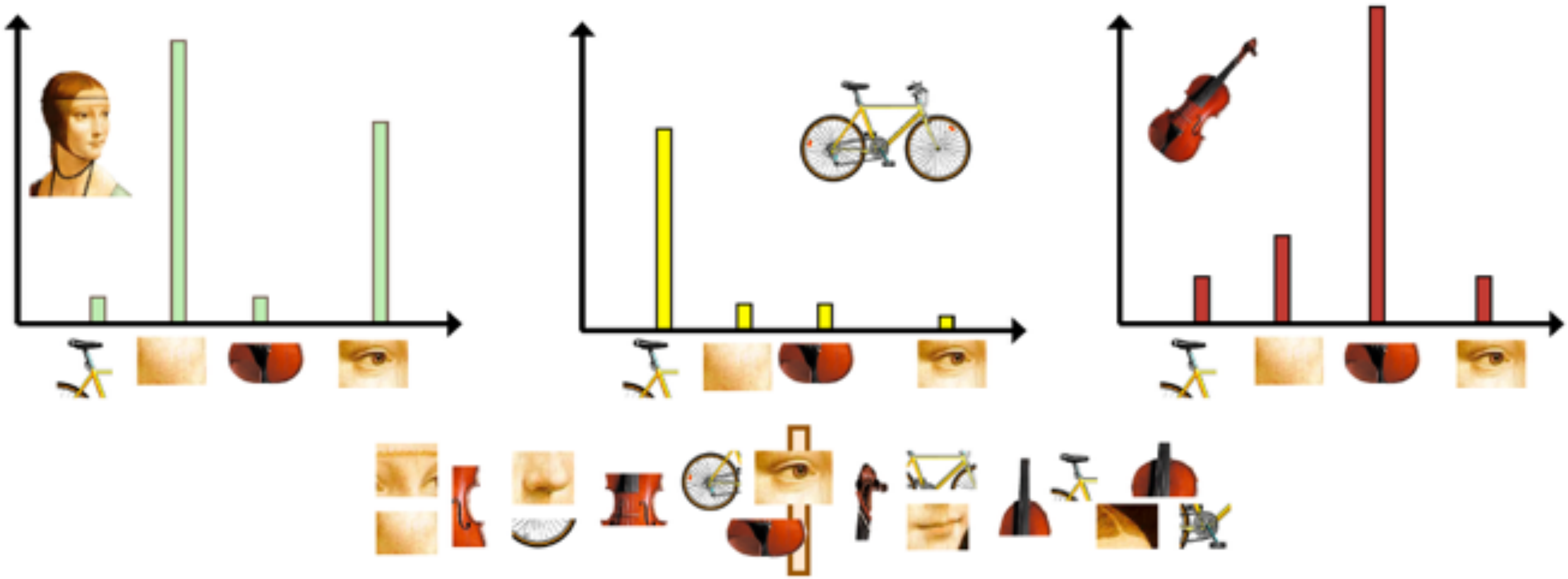




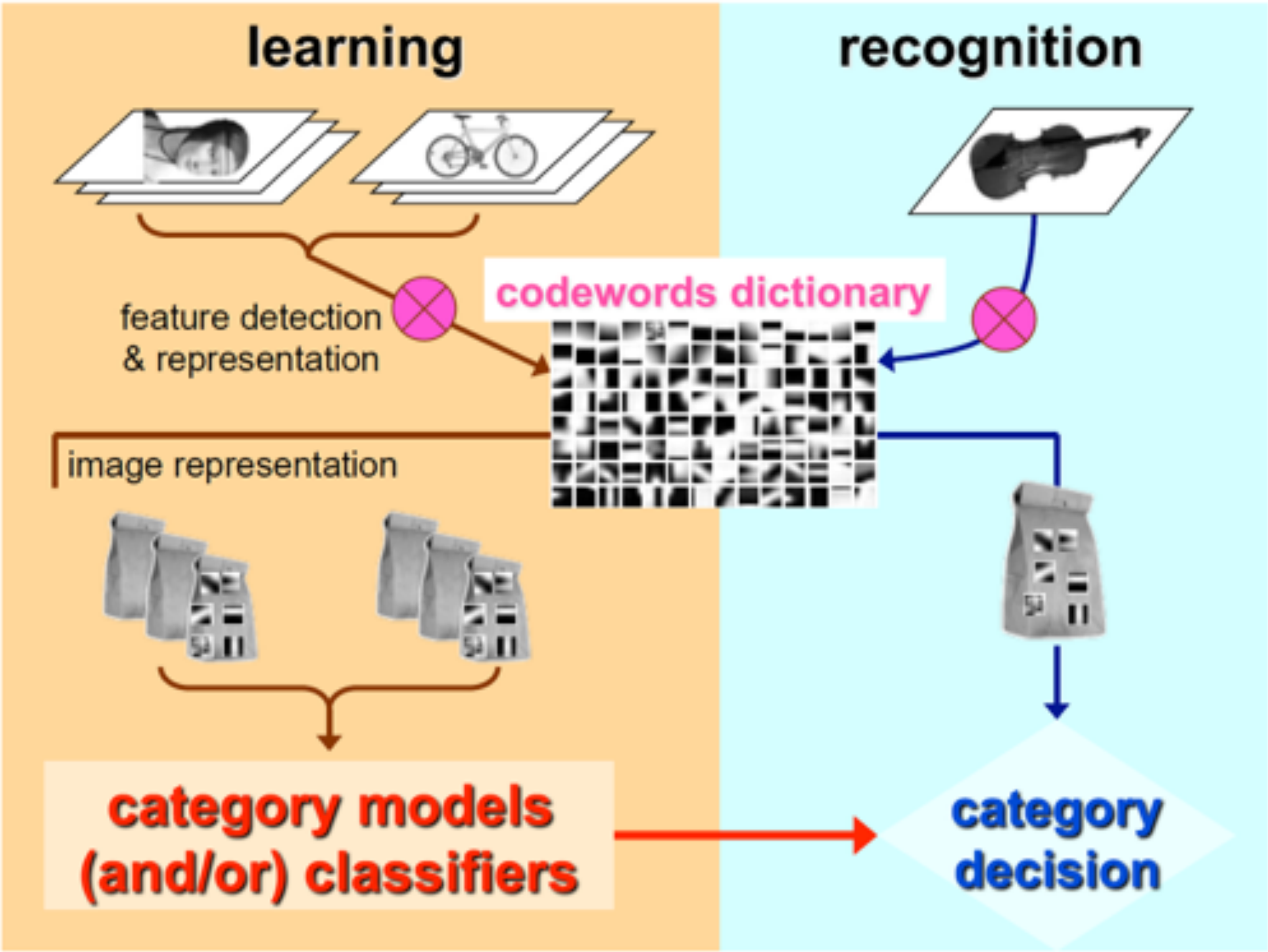

class	bag of features Zhang et al. (2005)	bag of features Willamowski et al. (2004)	Parts-and-shape model Fergus et al. (2003)
airplanes	**98.8**	97.1	90.2
cars (rear)	98.3	**98.6**	90.3
cars (side)	**95.0**	87.3	88.5
faces	**100**	99.3	96.4
motorbikes	**98.5**	98.0	92.5
spotted cats	**97.0**	—	90.0

Bag of Words

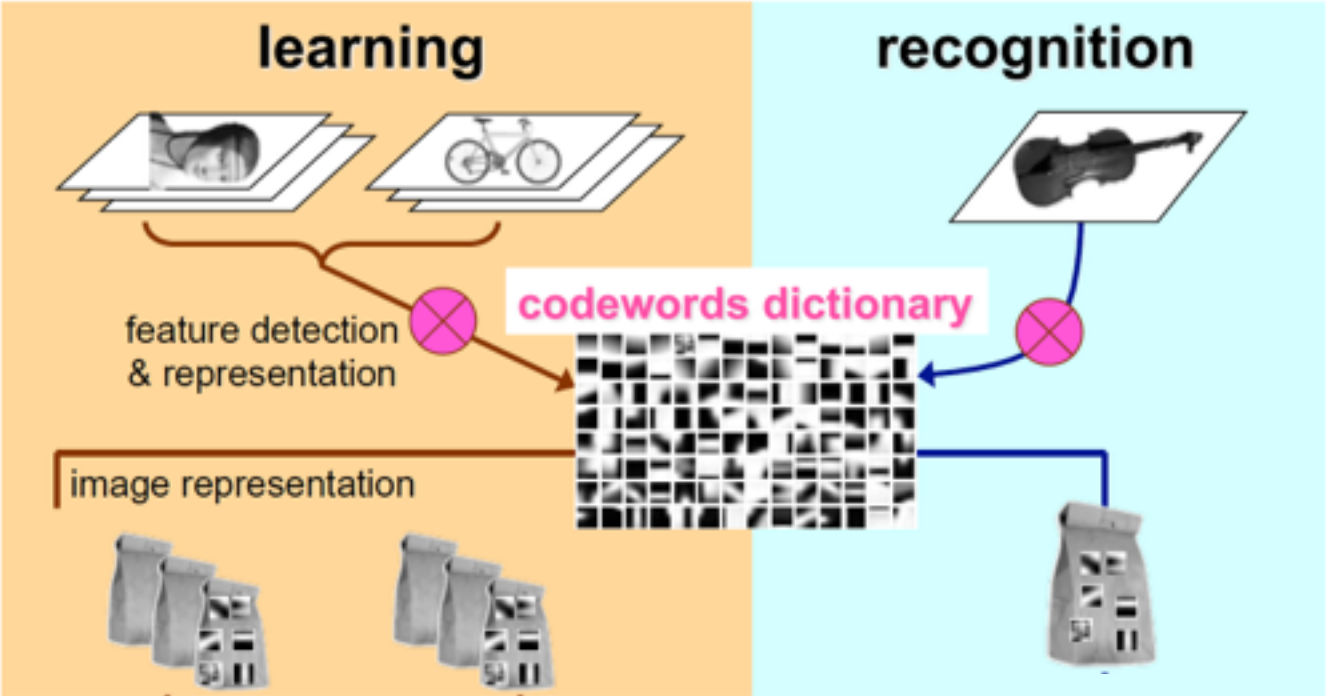
- Independent features
- Histogram representation



Bag of Words - Overview



Object Representation

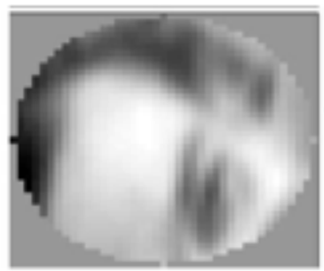


Feature Detection and Representation

Feature Detection and Representation

Compute descriptor

e.g. SIFT, shape context, etc.

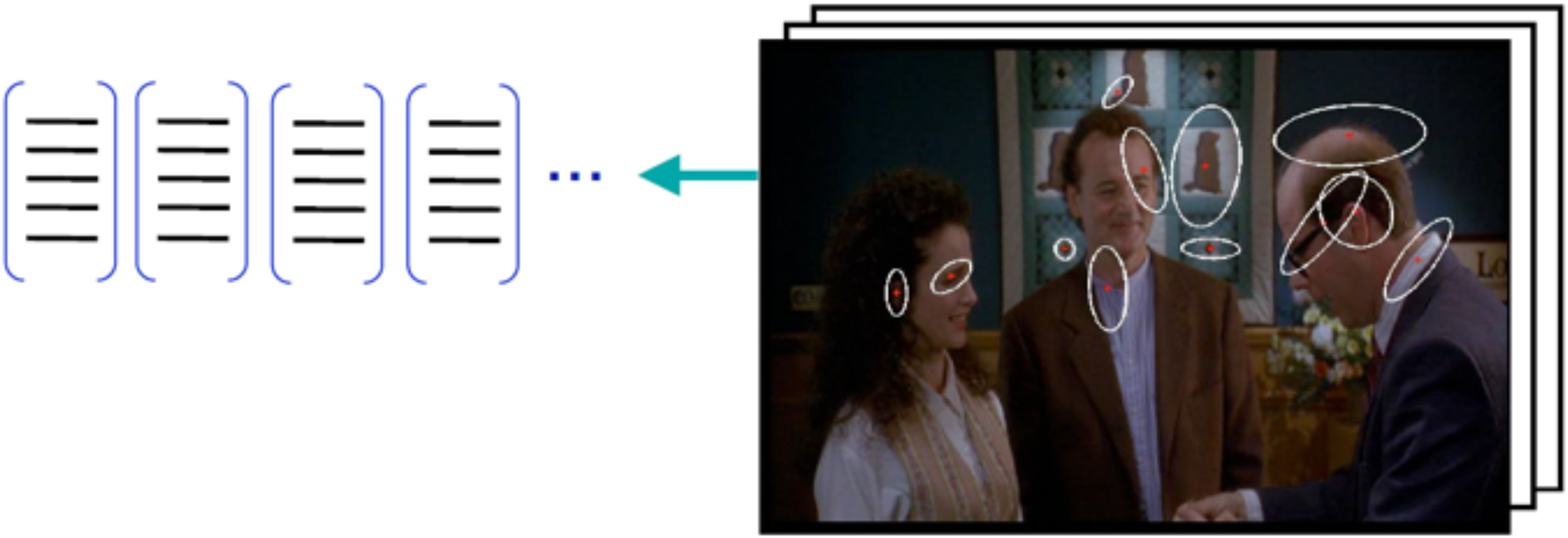


Normalize patch

Detect patches

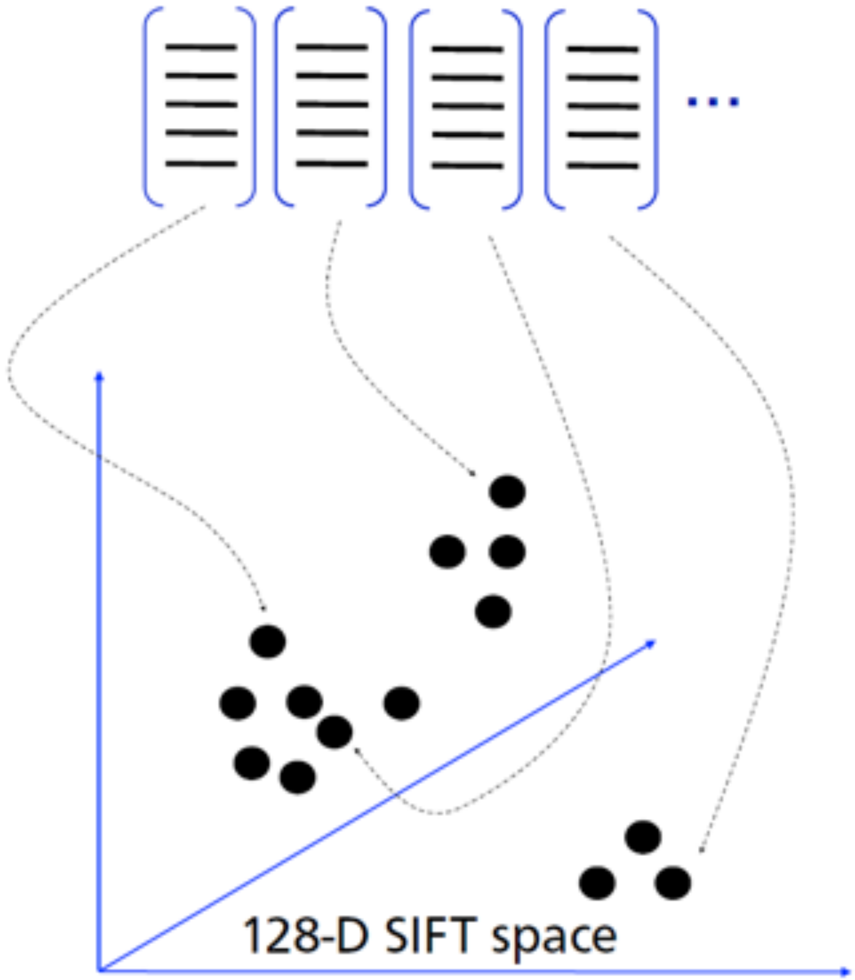
Local interest operator (e.g. Harris-Laplace) or regular grid

Feature Detection and Representation

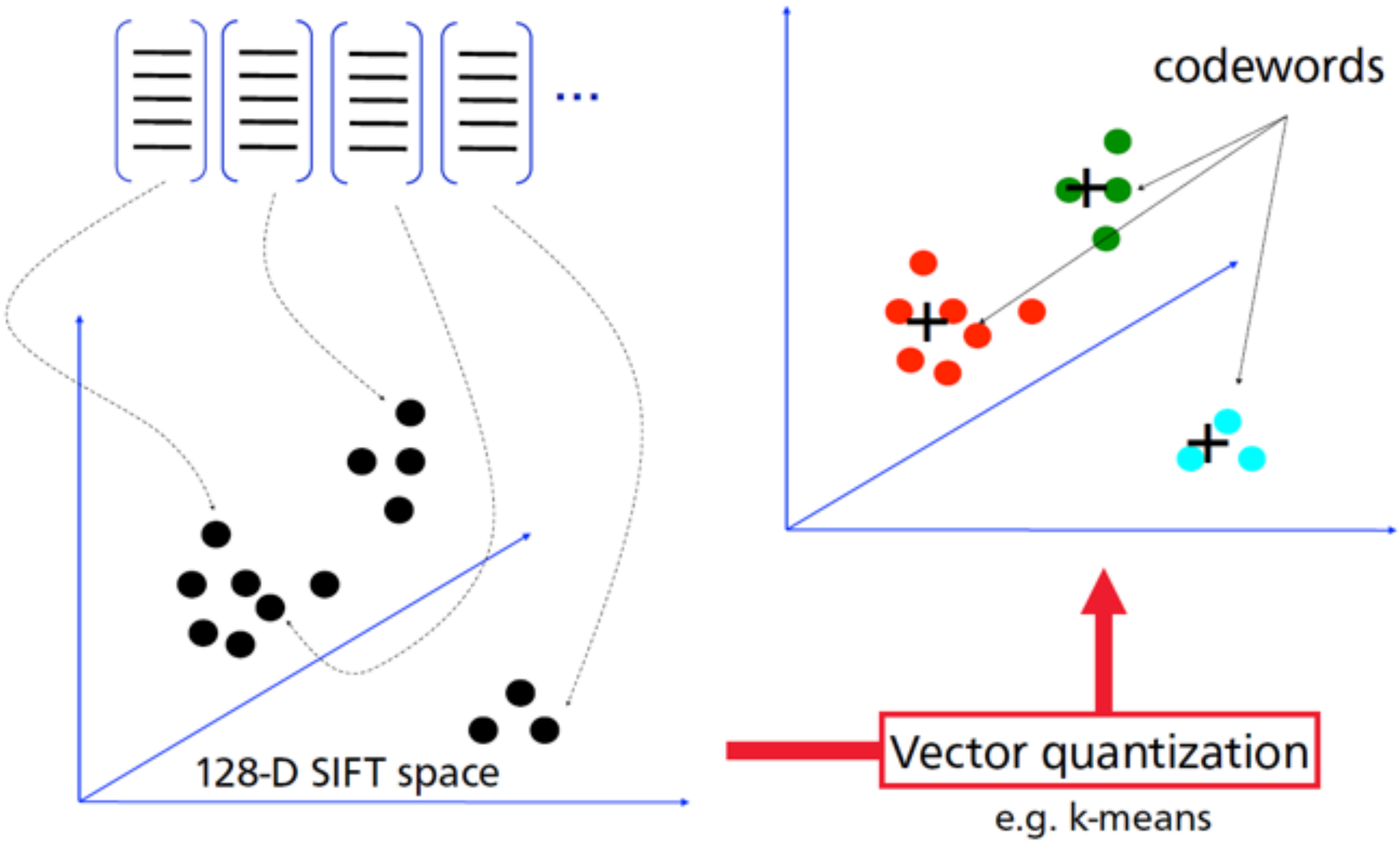


Take all training images

Feature Detection and Representation

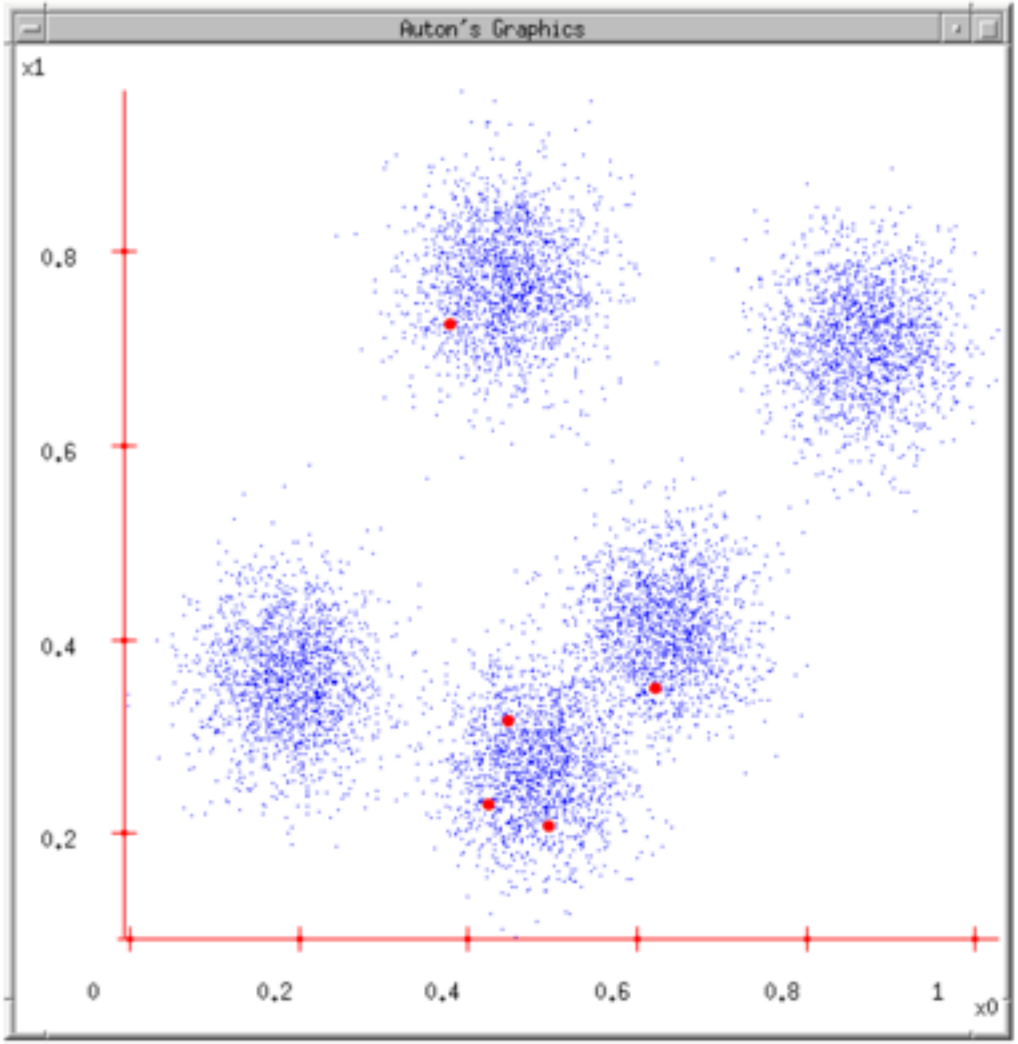


Codeword dictionary formation



K-means

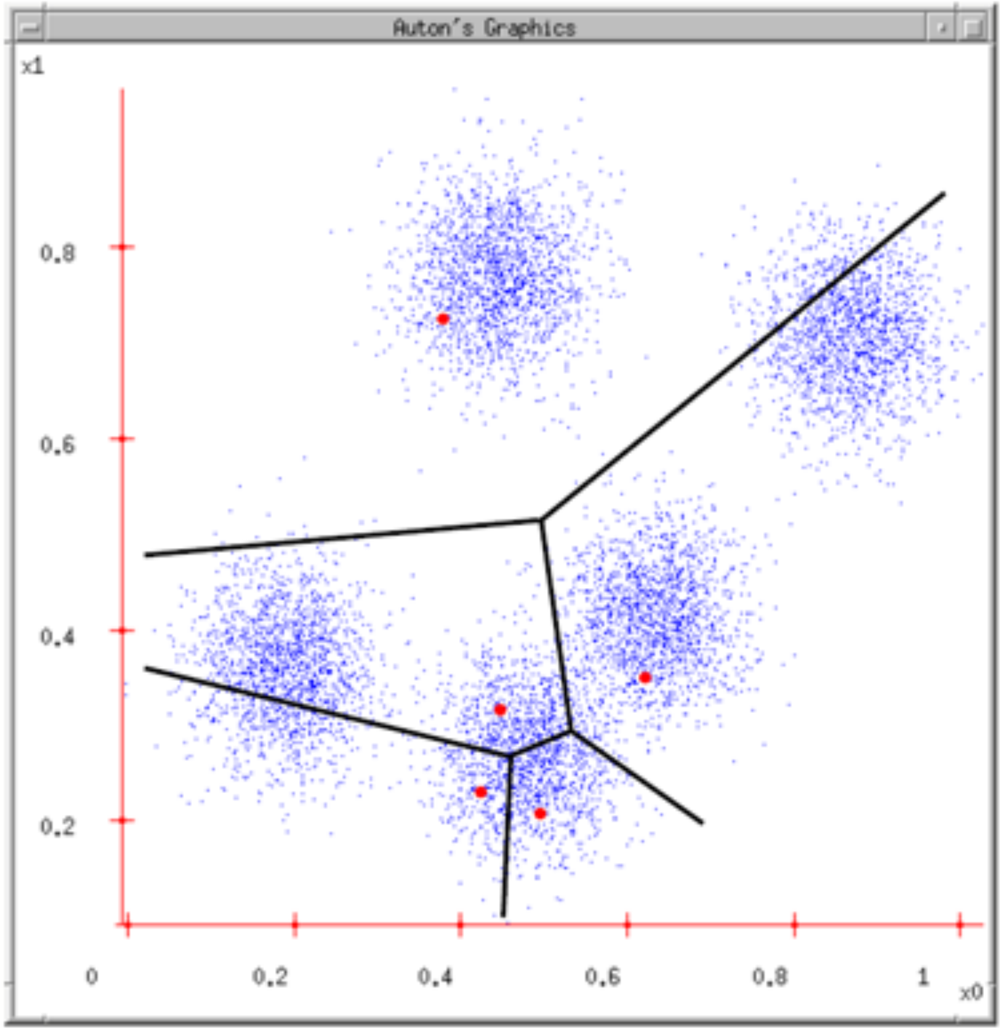
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



Reminder: K-means

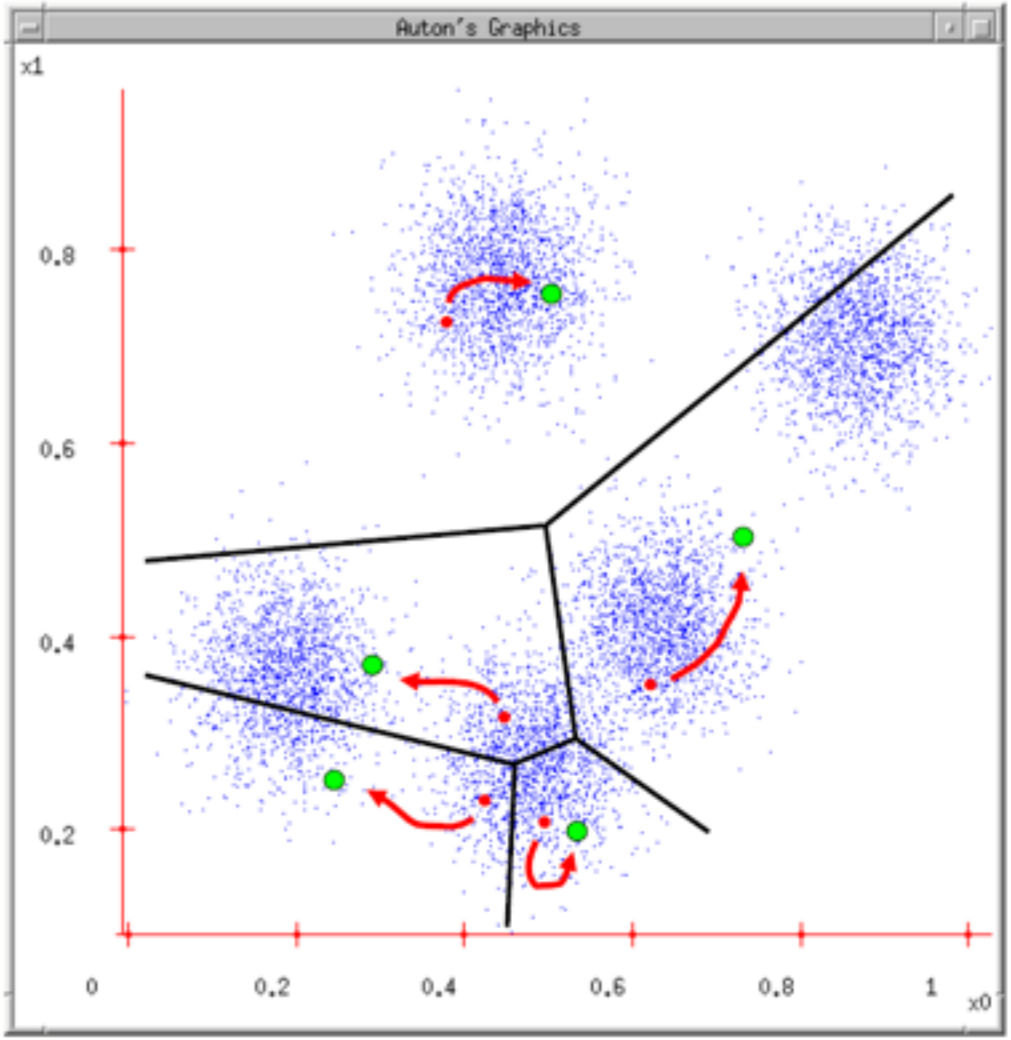
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



K-means

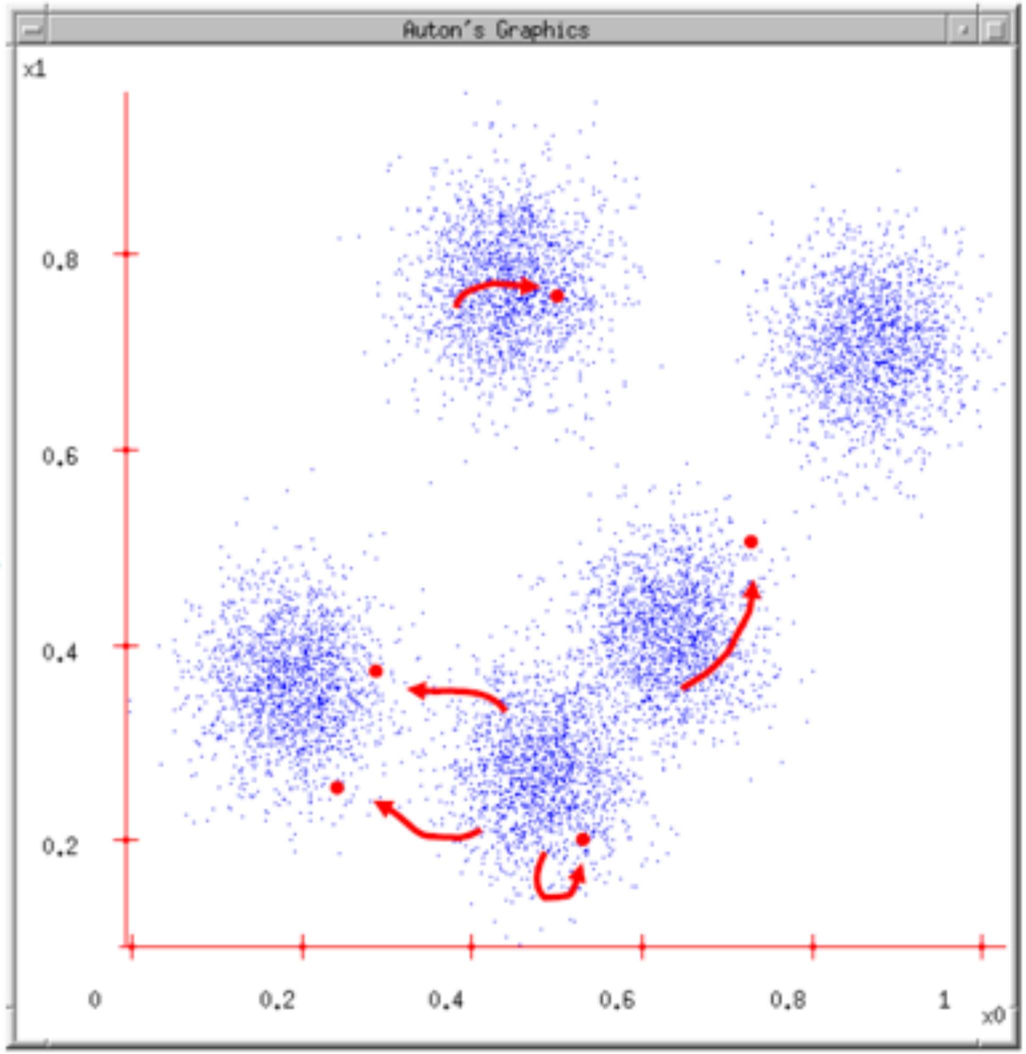
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



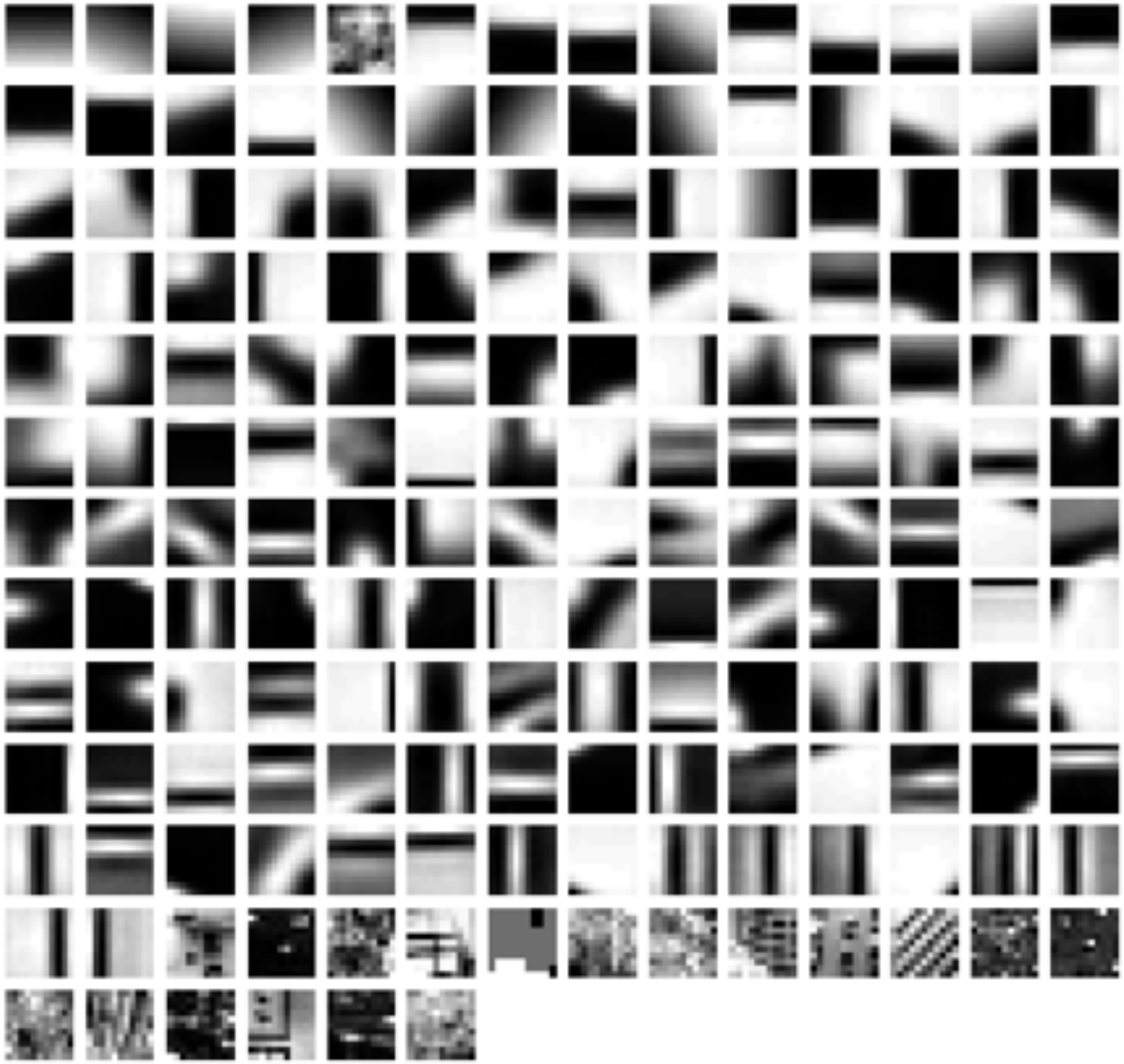
Reminder: K-means

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!
   (Repeat means go to step 3)

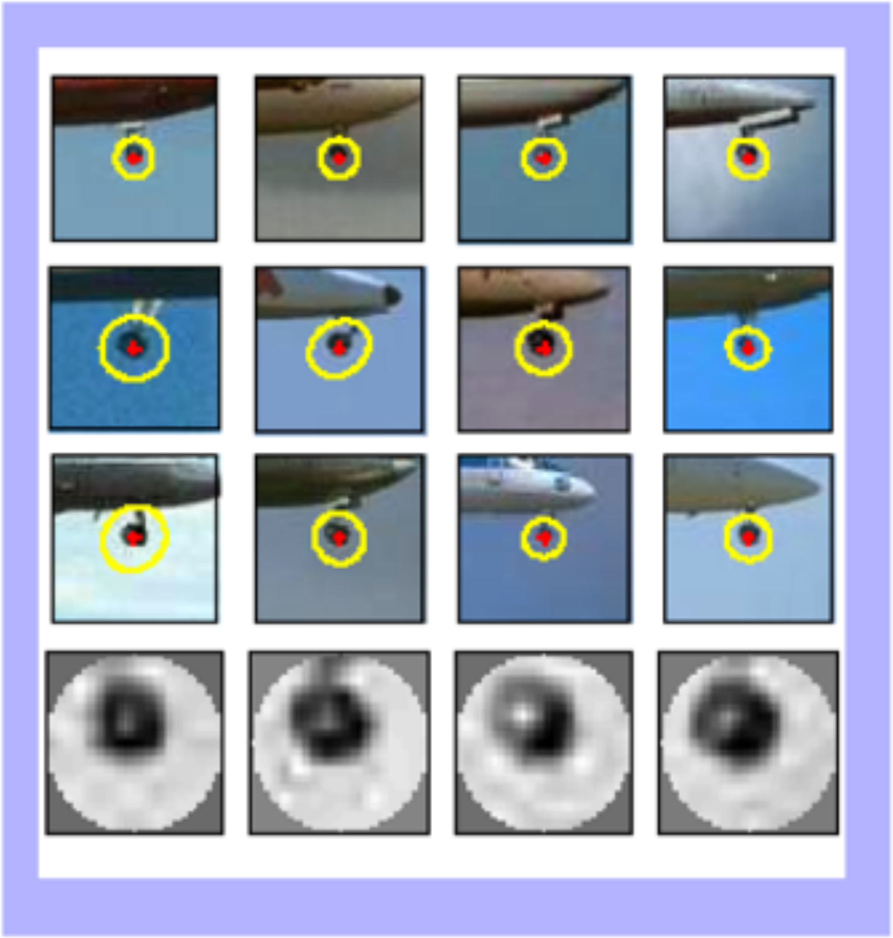


Codeword dictionary visualization

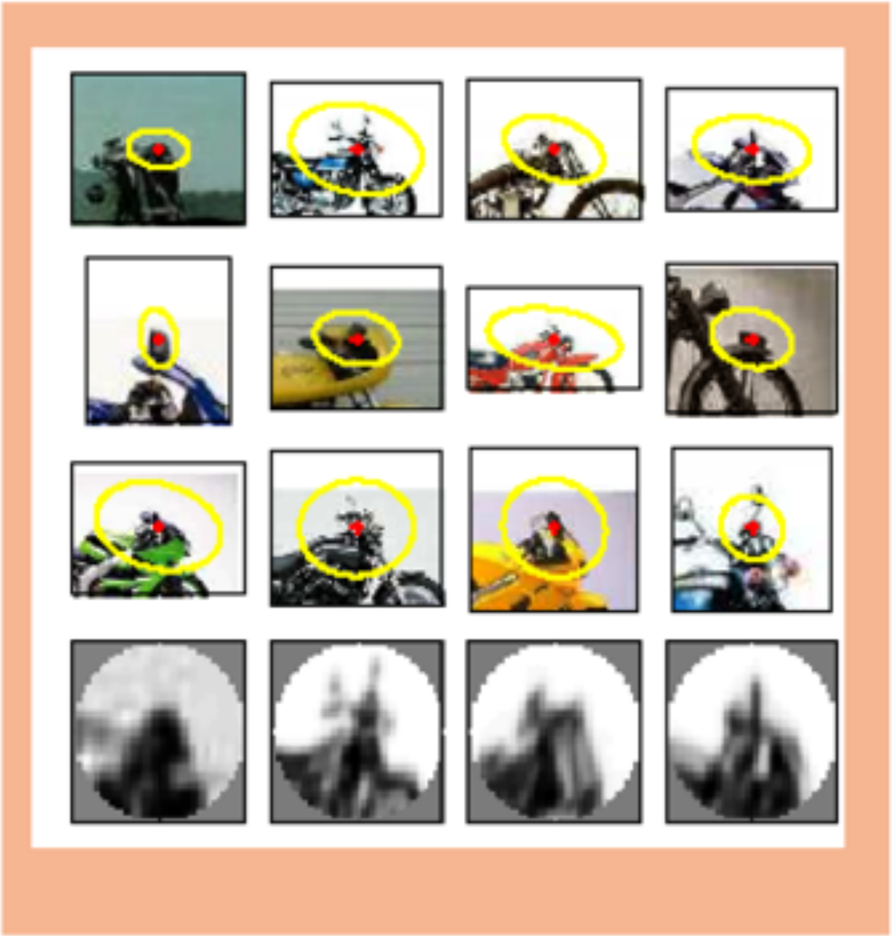


K = 174 (averaged patches for each cluster) [from Fei Fei Li]

Image Patch examples of Codewords



Examples which are assigned to same codeword

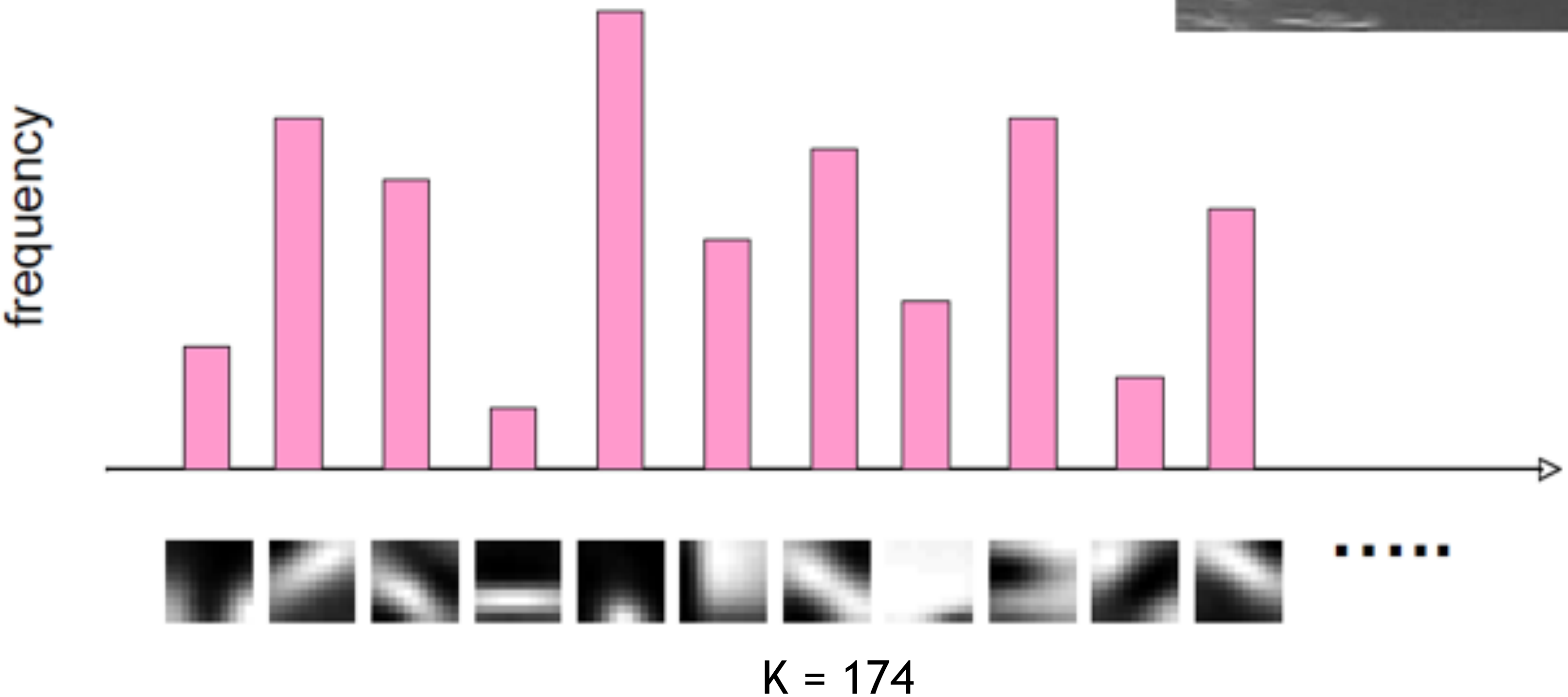


Examples which are assigned to same codeword

[from Josef Sivic]

Bag of Words – Image Representation

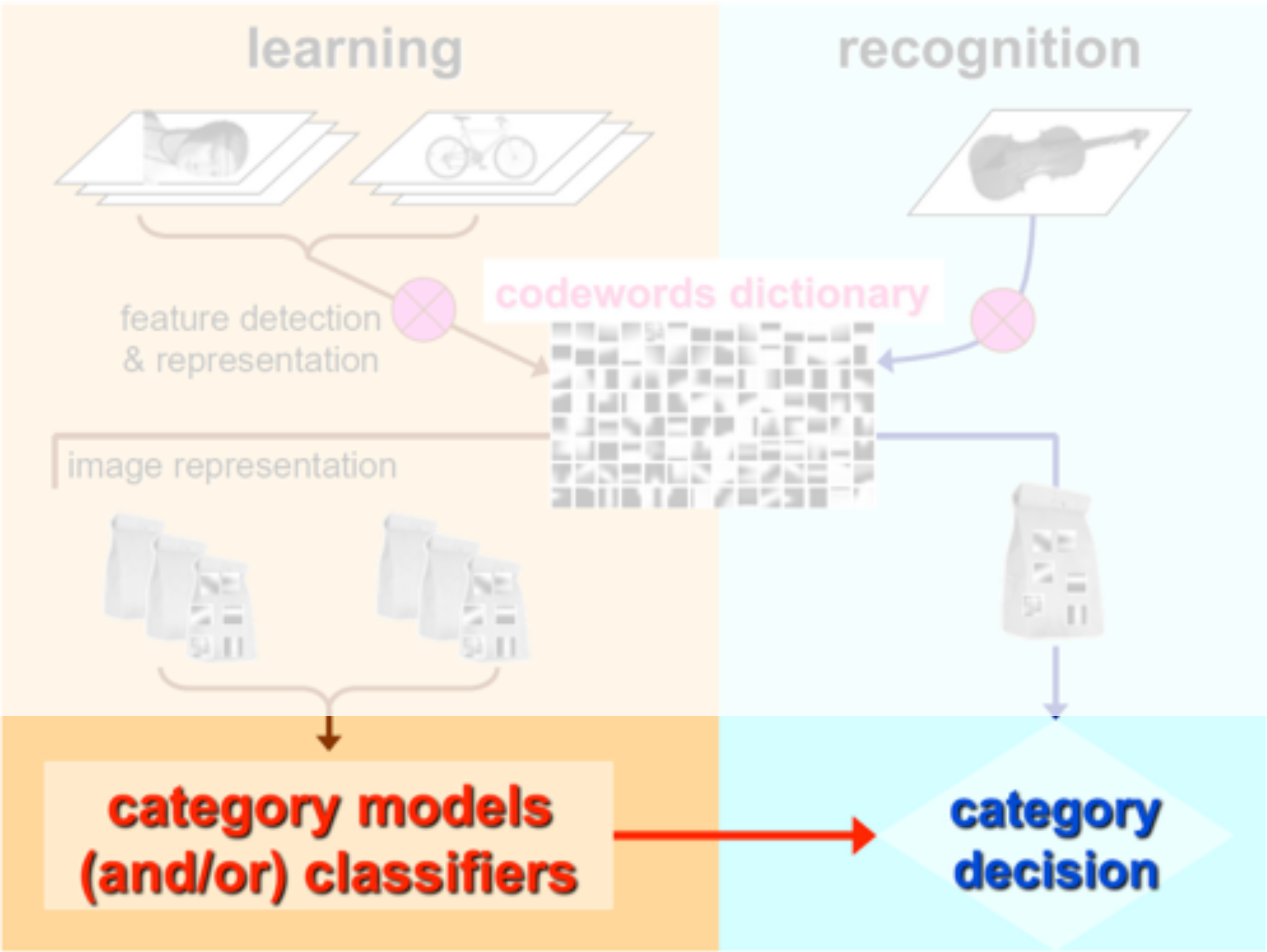
- Histogram of features assigned to each cluster



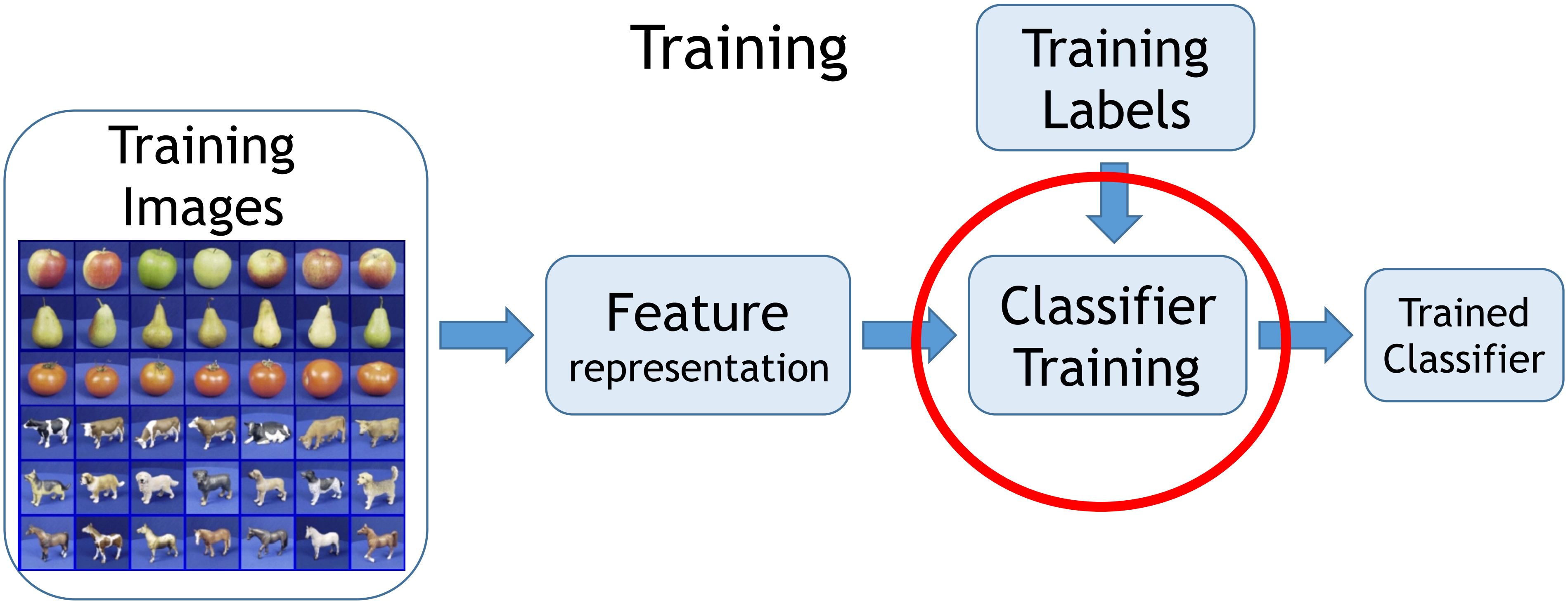
Roadmap (this lecture)

- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching

Bag of Words - Overview

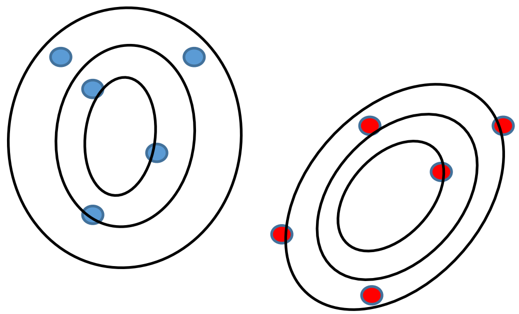


Classifiers

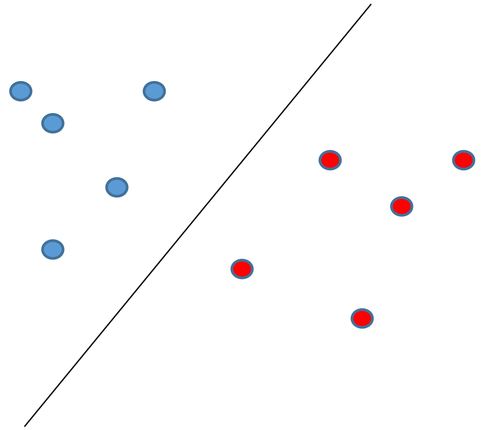


Two approaches

Generative approach:
models distributions



Discriminative function:
models decision function



Generative vs. Discriminative

Generative

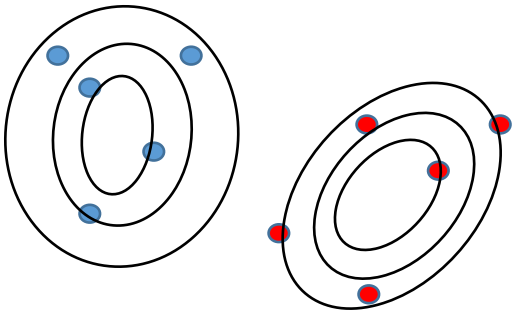
- Training
  - Maximize joint likelihood of data and labels
  - Assume (or learn) probability distribution and dependency structure
  - Can impose priors
- Testing
  - P(y=1, x) / P(y=0, x) > t?
- Examples
  - Foreground/background GMM
  - Naïve Bayes classifier
  - Bayesian network

Discriminative

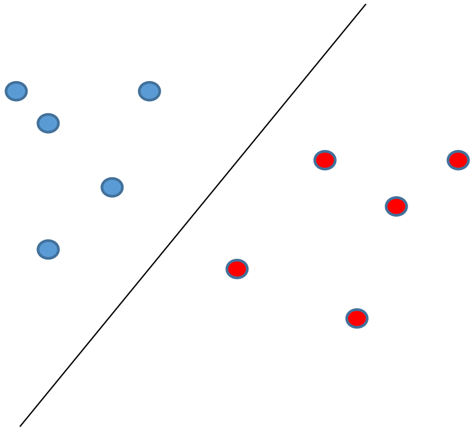
- Training
  - Learn to directly predict the labels from the data
  - Assume form of boundary
  - Margin maximization or parameter regularization
- Testing
  - $f(x) > t$; e.g., $w^T x > t$
- Examples
  - Logistic regression
  - SVM
  - Boosted decision trees

Two approaches

Generative approach:
models distributions



Discriminative function:
models decision function

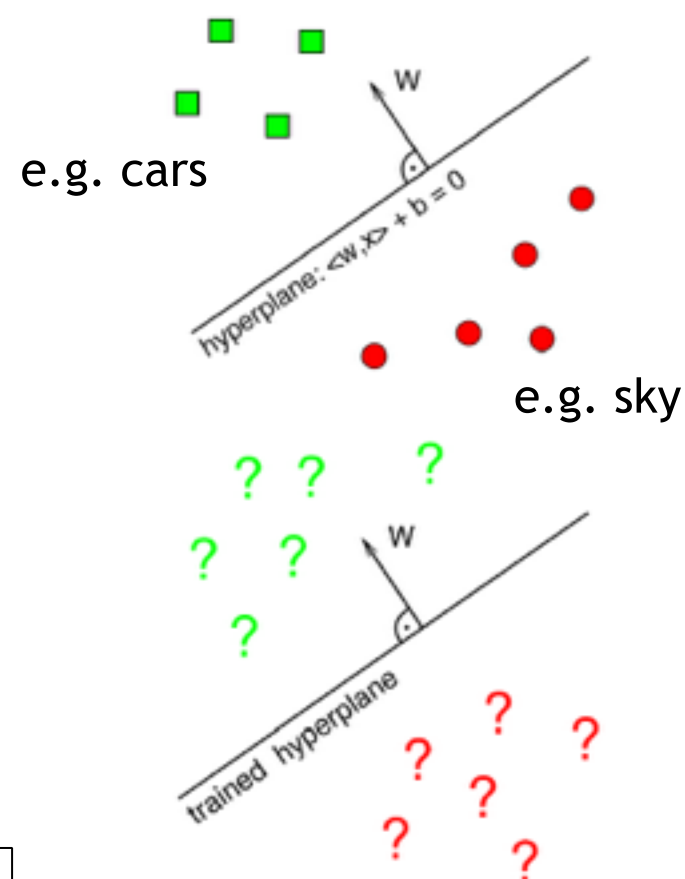


Discriminative functions

"2D space (two codewords)"

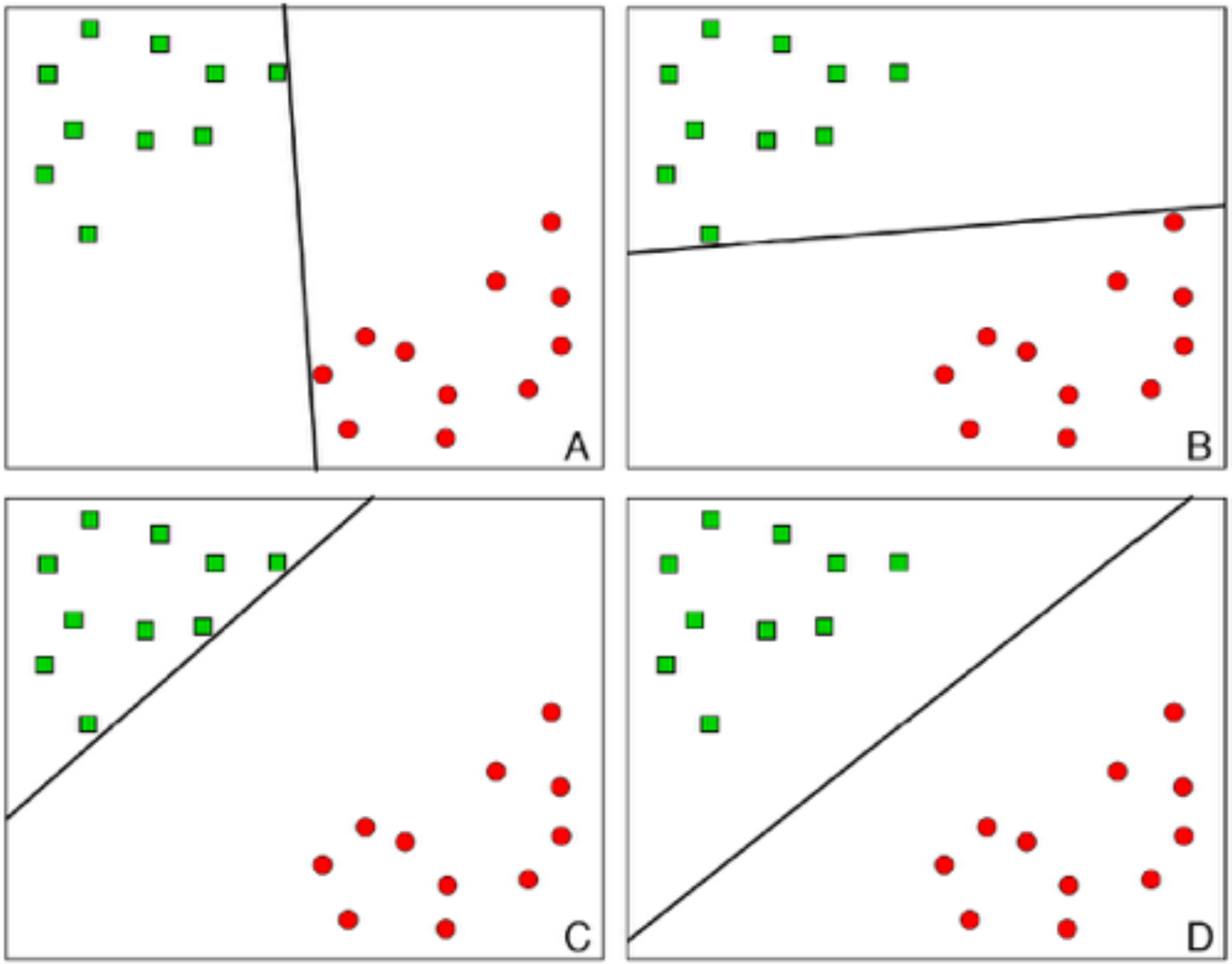
- **Linear discriminant function:**
  - Linear hyperplane:
    $$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b$$
  - trained on samples of both classes C_1 (■) and C_2 (●)

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



Support Vector Machine is the optimal classifier
-> see Machine Learning 1

Which Hyperplane is best and why?



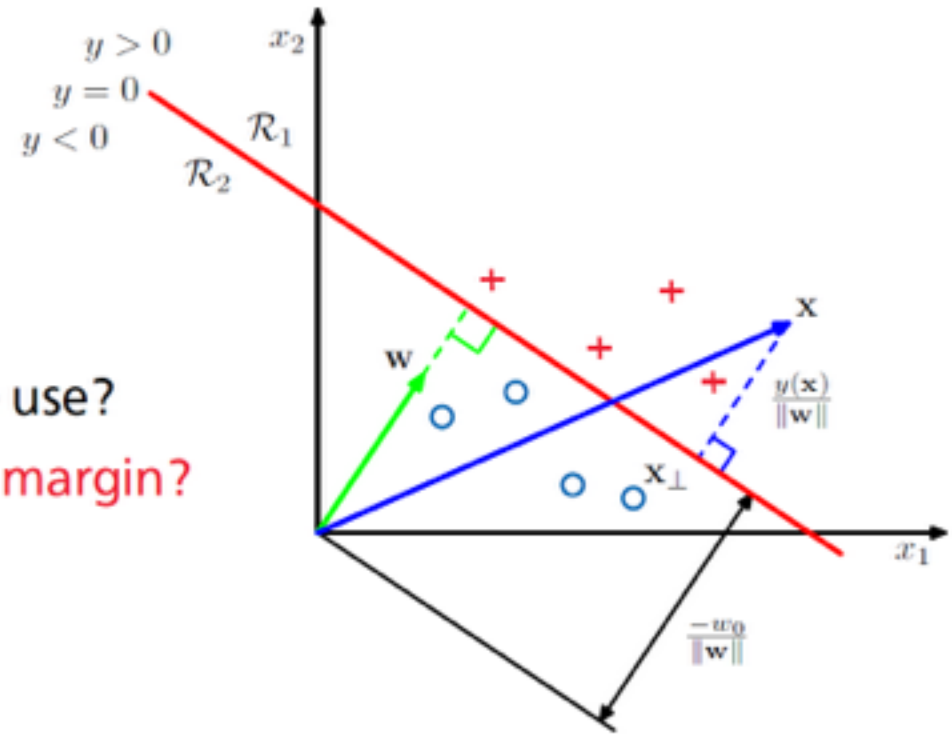
SVM classifier: Max-Margin behavior
– best generalization

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

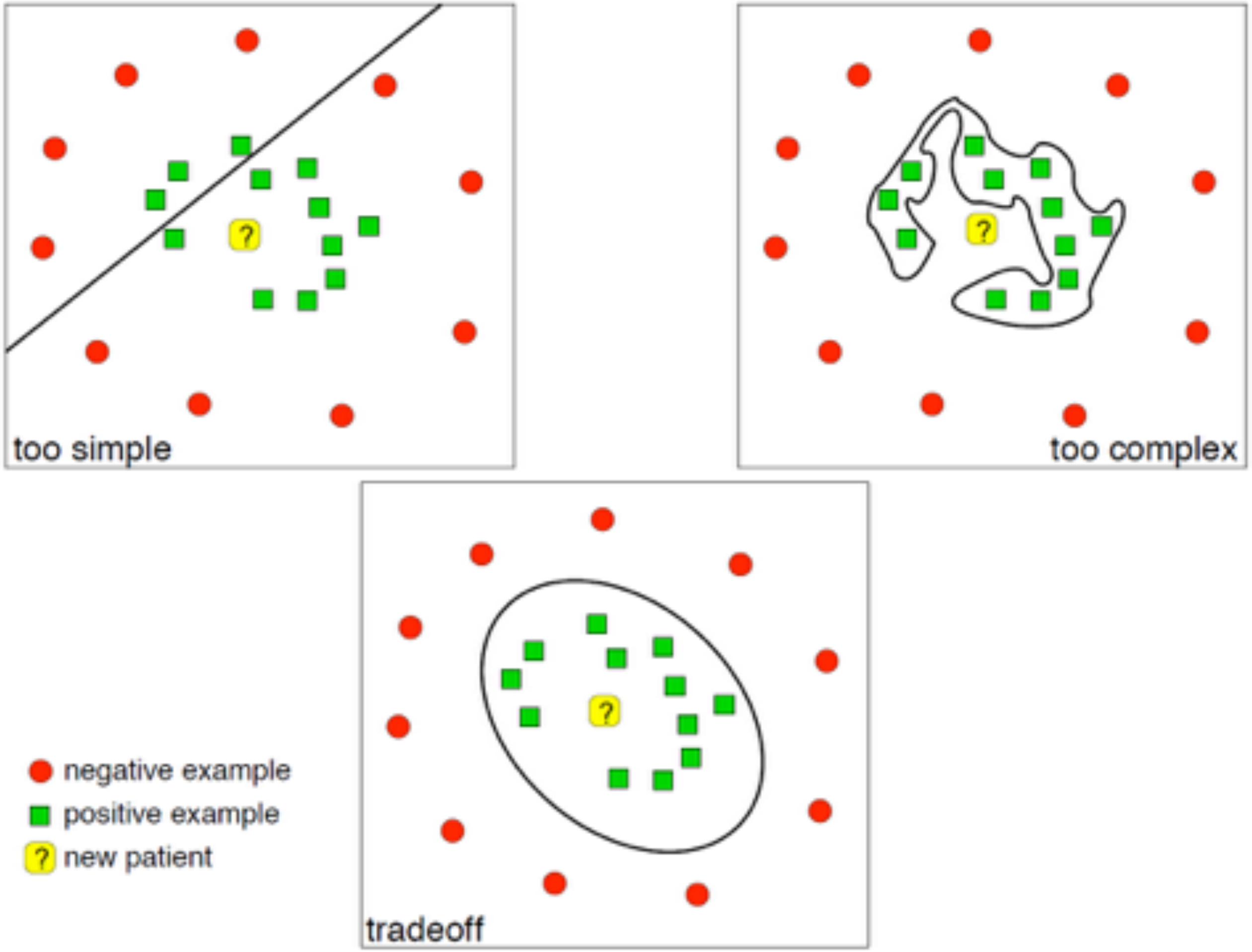
  - Hyperplane that separates the data:

    $$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

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



Simpler decision functions are better



Best generalization

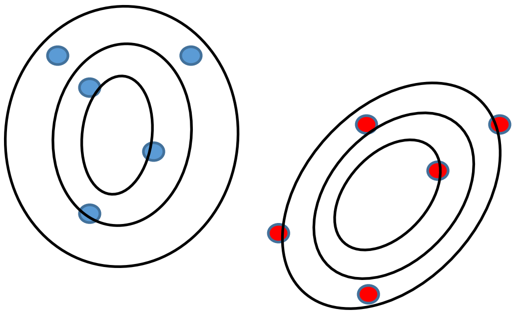
[Florian Markowetz]

Half way slide

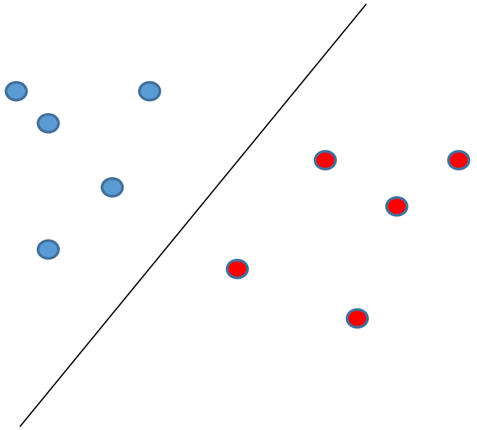
3 Minutes break

Two approaches

Generative approach:
models distributions



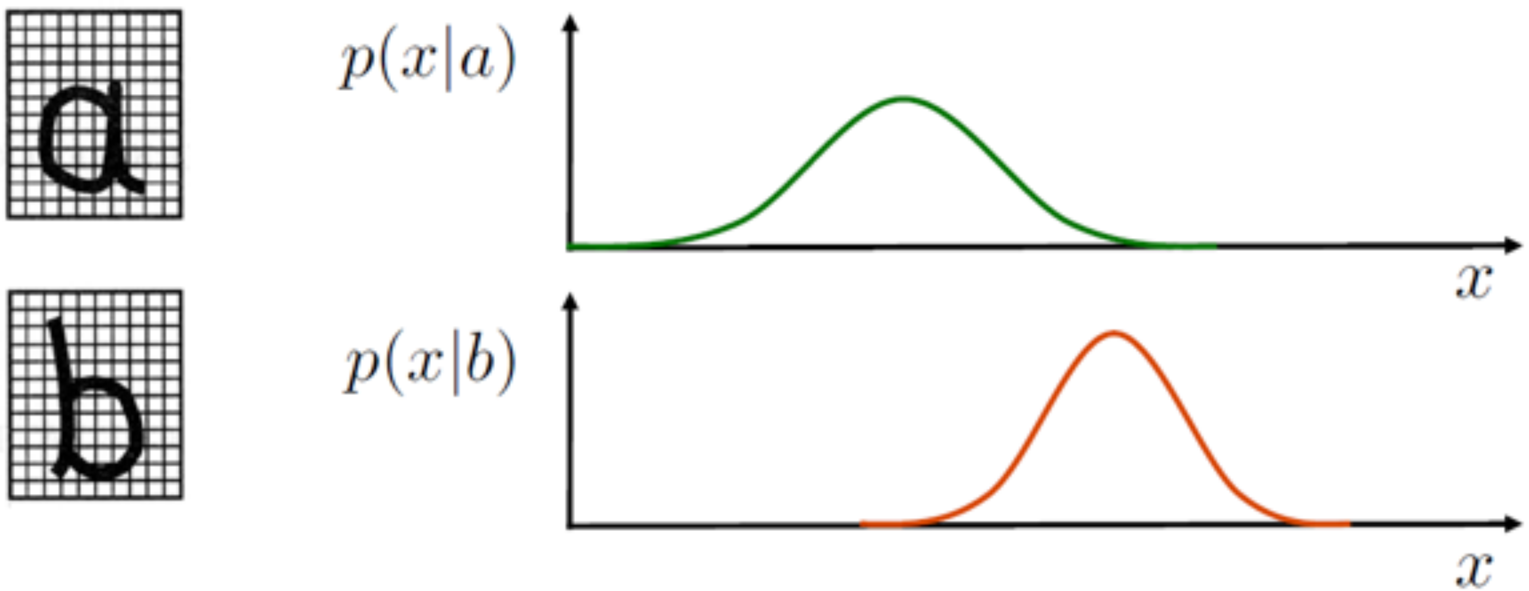
Discriminative function:
models decision function



- 1st concept: Class conditional probabilities
  - 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.

$$\boxed{p(x|C_k)}$$

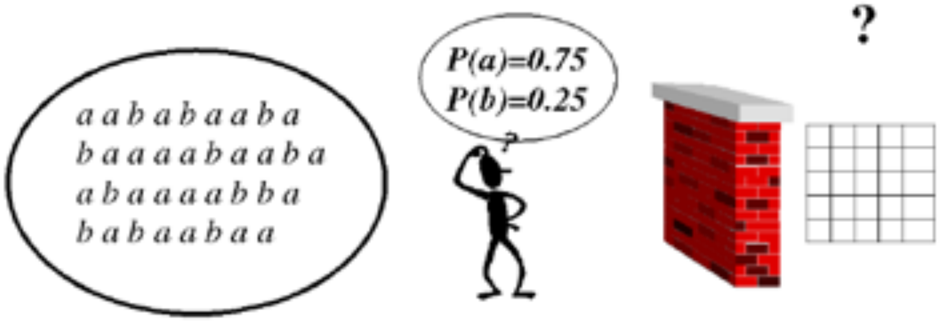
(Likelihood)



- 2nd concept: Class priors $\boxed{p(C_k)}$

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

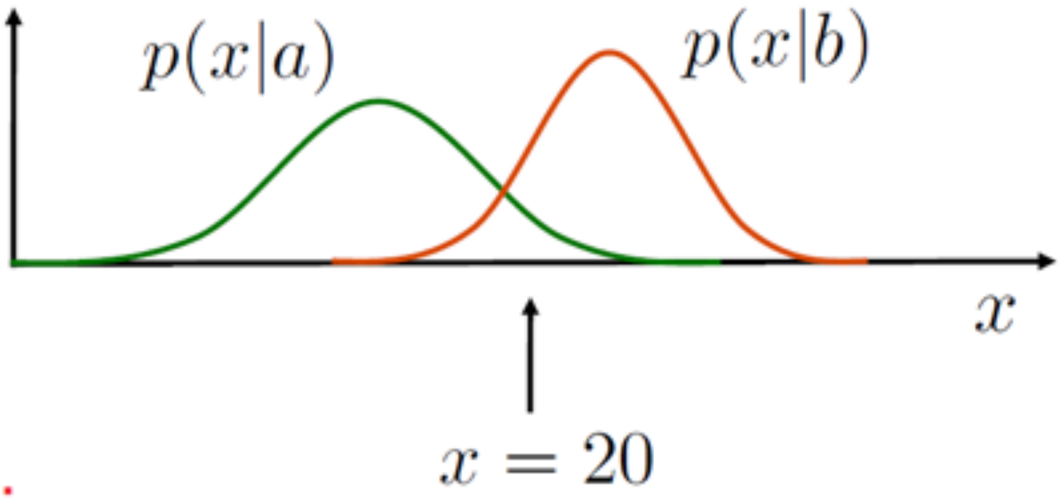
  - Example:



$$C_1 = a \qquad p(C_1) = 0.75$$
$$C_2 = b \qquad p(C_2) = 0.25$$

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

- 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 may decide class a.

Bayesian Decision Theory

- 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 posterior

class-conditional probability (likelihood)

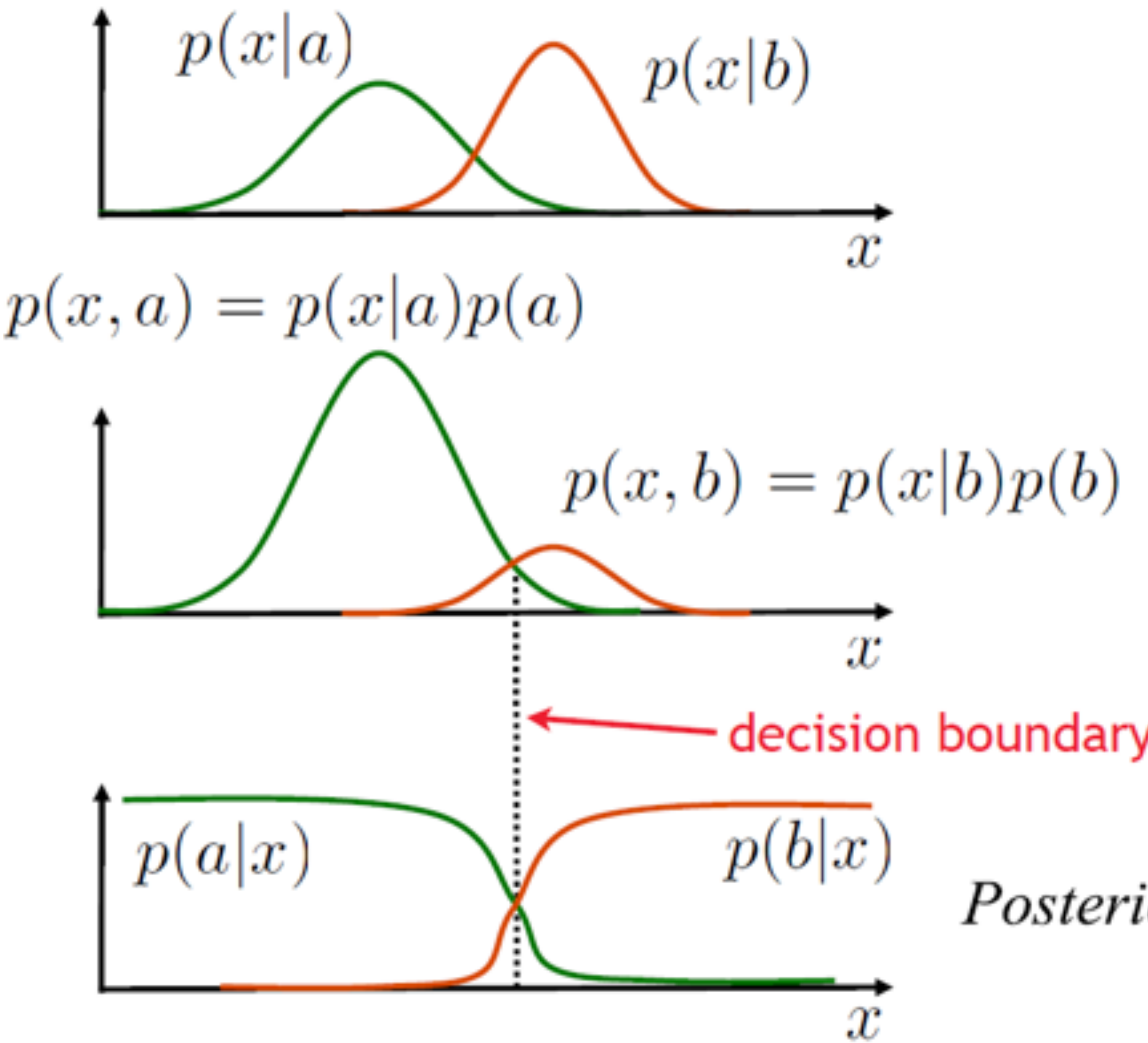
class prior

normalization term

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

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

Bayesian Decision Theory



Likelihood

$$\textit{Likelihood} \times \textit{Prior}$$

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

Bayesian Decision Theory

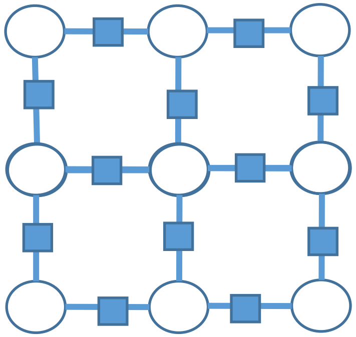
- **Decision rule:**
  - Decide C_1 if $$p(C_1|x) > p(C_2|x)$$
  - This is equivalent to $$p(x|C_1)p(C_1) > p(x|C_2)p(C_2)$$

We do not need the normalization!

- **MAP classifier:**
  - A classifier obeying this rule is called a MAP classifier (sometimes called Bayes optimal classifier)

Relation to previous lectures

- Image gets a label (class): K labelings

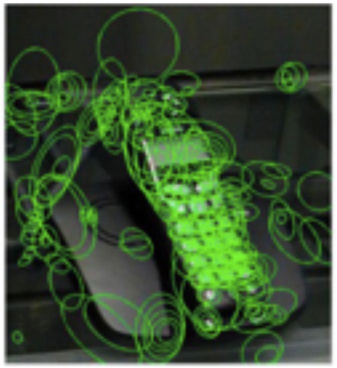


- Each pixel gets a label (class): K^n labelings
- Pixels are structured

Naive Bayes Classifier

A **naive Bayes classifier** is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions

- Encode each image as a feature vector $\boldsymbol{x} = (x^1, \dots, x^n)$ where n is the number of interest points.
- $x^j \in \{w_1, \dots w_m\}$. Here m visual words.



~200 interest points

Interest points for codewords (visual words)

- Naive Bayes Classifier assumes that visual words are conditionally independent given object class: $P(x|c) = \prod_j P(x^j|c)$ (which is rarely true in practice)
- Naive Bayes Classifier:
  $c^* = argmax_c \; P(c|x) = argmax_c \; P(c) \, P(x|c) = argmax_c \; P(c) \prod_j P(x^j|c)$

Image Classification with Naive Bayes

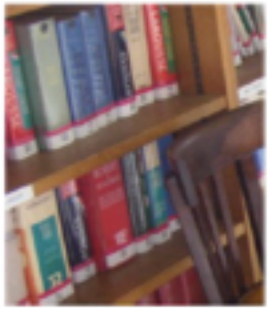
- Image dataset: 7 object categories, arbitrary views, partial occlusions

Image Classification with Naive Bayes

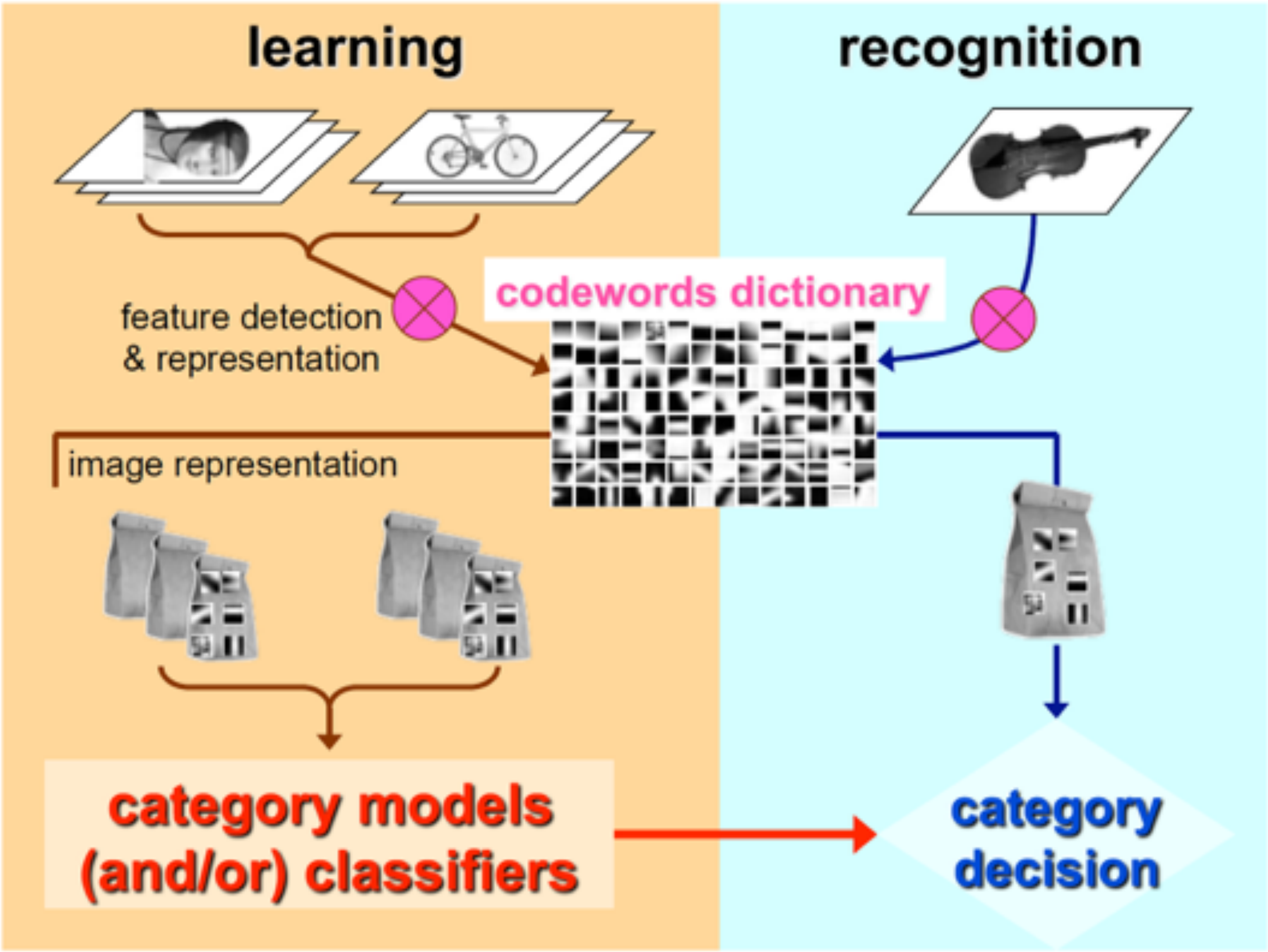
Table 1. Confusion matrix and the mean rank for the best vocabulary (*k=1000*).

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:

Bag of words – Done!



Summary and Discussion

- **Bag of words representation:**
  - Sparse representation of object categories
  - Many Machine learning techniques can be applied (here naïve Bayes and SVM)
  - Robust to occlusion
  - Allows sharing of representation between multiple classes (via codeword dictionary)

- **Problems:**
  - Spatial distribution of visual works is not modelled.

Roadmap (this lecture)

- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching

Spatial Pyramid Matching

- Add spatial information to the bag-of-features

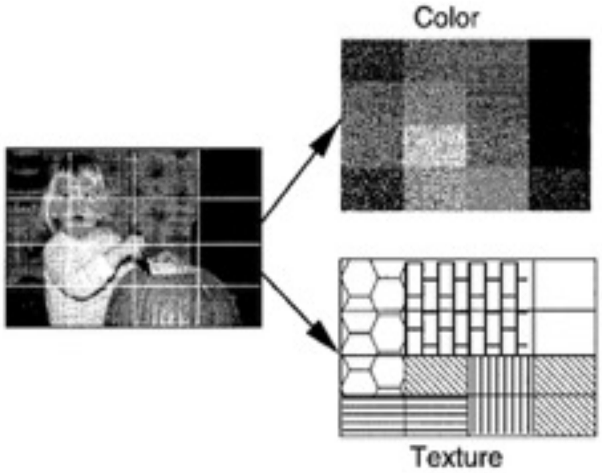
- Perform matching in 2D image space

[Lazebnik, Schmid & Ponce, CVPR 2006]

Spatial Pyramid Matching

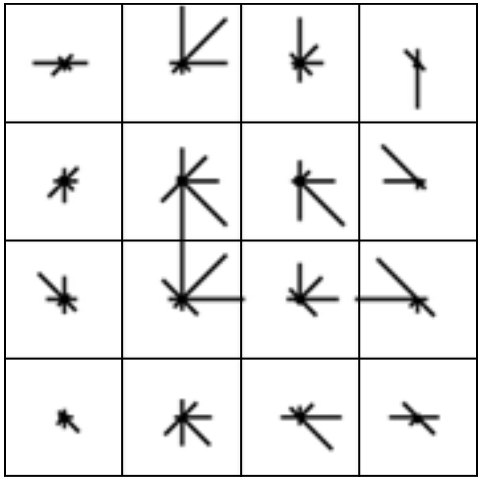
Similar approaches:

- Subblock description [Szummer & Picard, 1997]
- SIFT [Lowe, 1999]
- GIST [Torralba et al., 2003]



Szummer & Picard (1997)

SIFT



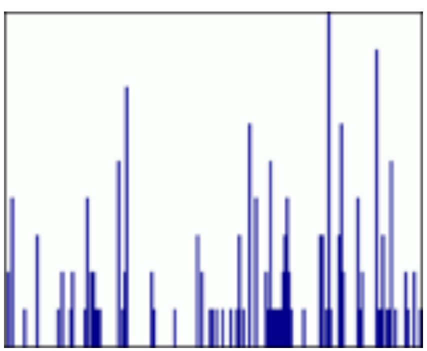
Lowe (1999, 2004)

Gist

Torralba et al. (2003)

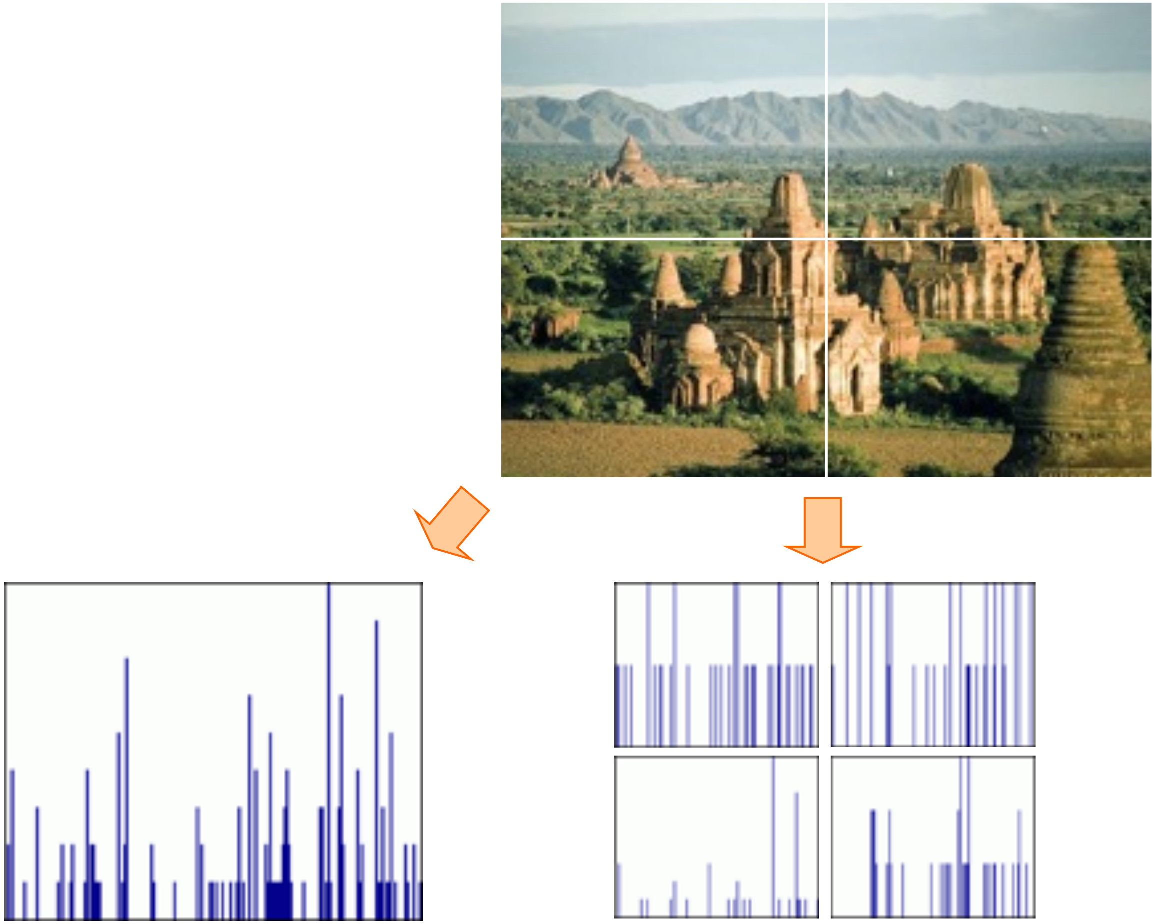
Spatial pyramid representation

Locally orderless representation at several levels of spatial resolution



level 0

Spatial pyramid representation

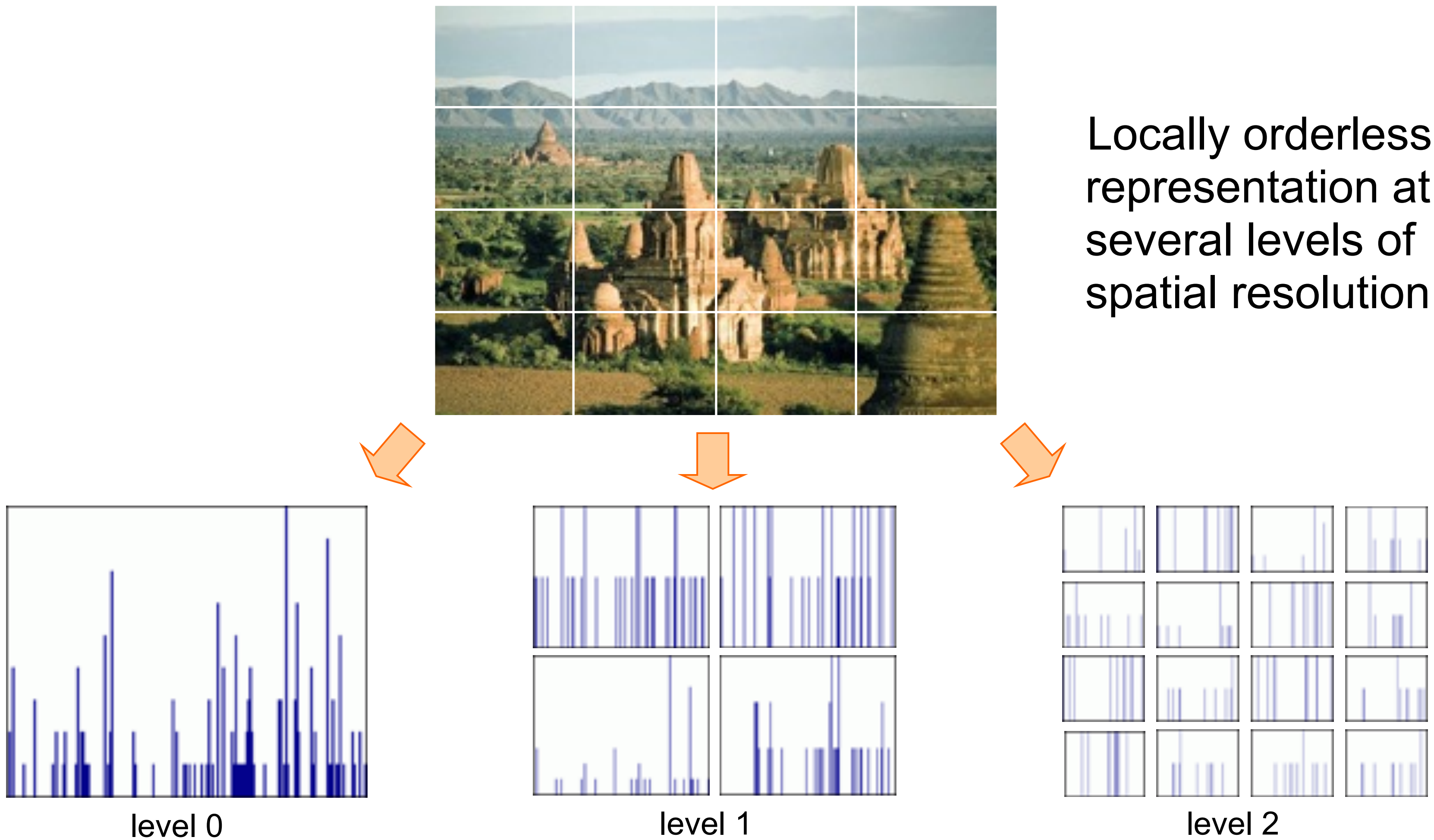


Locally orderless representation at several levels of spatial resolution

level 0

level 1

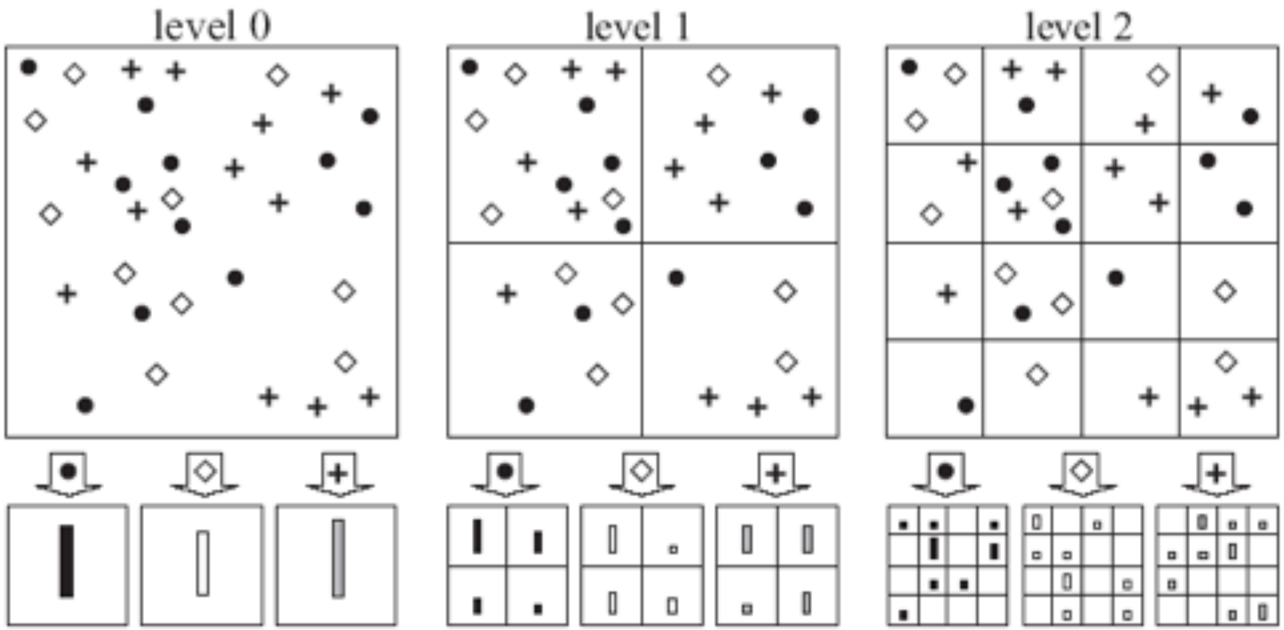
Spatial pyramid representation



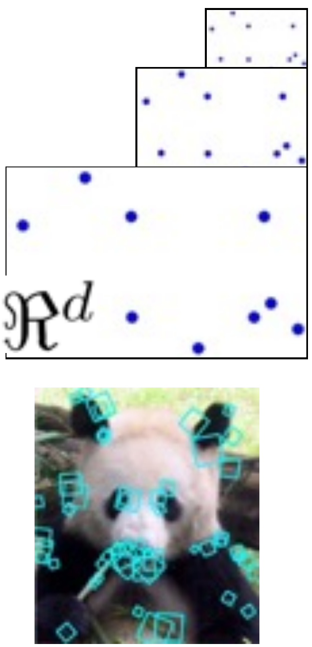
Locally orderless representation at several levels of spatial resolution

Spatial Pyramid Matching

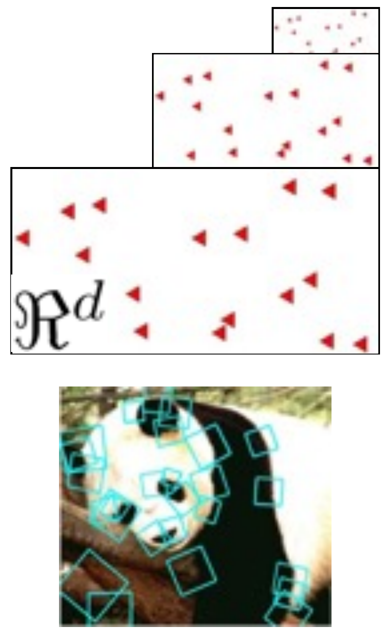
- Combination of spatial levels with pyramid match kernel
  [Grauman & Darell’05]



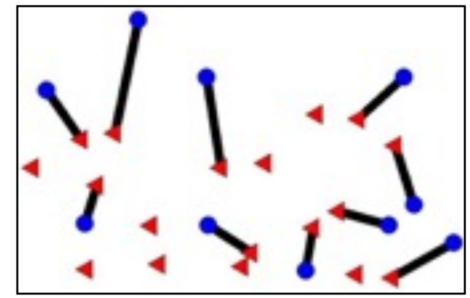
Pyramid Matching Kernel



$\cap$



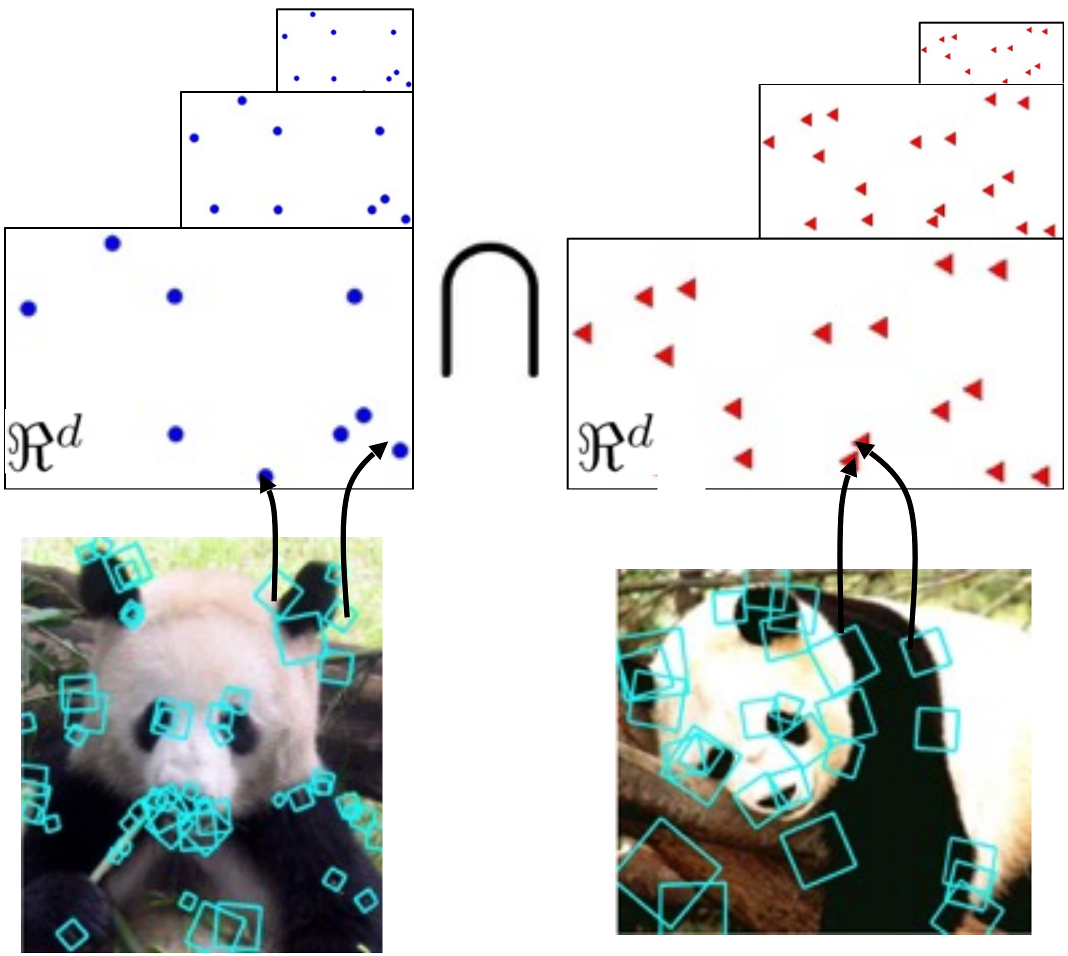
$\approx$



optimal partial matching between sets of features

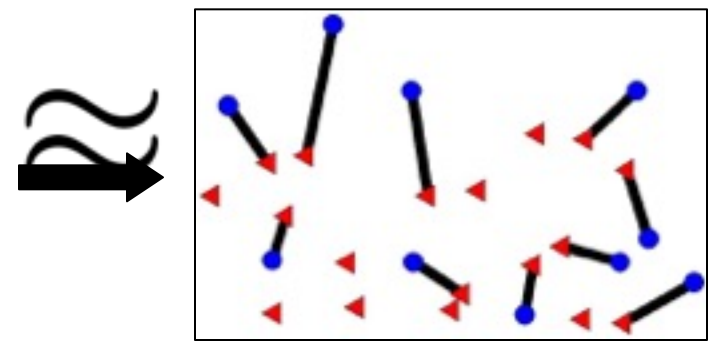
Slides Credit: Kristen Grauman

Pyramid Matching Kernel



$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$
$\vec{\mathbf{x}}_i \in \Re^d$

$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$
$\vec{\mathbf{y}}_i \in \Re^d$



optimal partial matching

$$\max_{\pi:\mathbf{X}\rightarrow\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

Pyramid match overview

Pyramid match kernel measures similarity of a partial matching between two sets:

- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell
- Approximate similarity between matched points with worst case similarity at given level

Pyramid match kernel

Number of newly matched pairs at level *i*

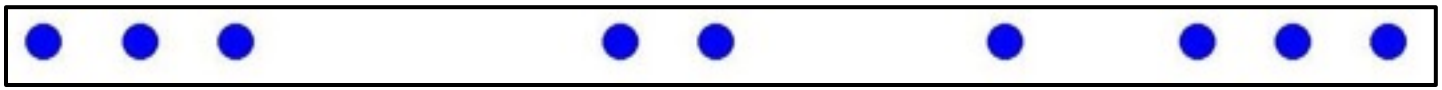
Approximate partial match similarity

$$K_{\Delta} = \sum_{i=0}^{L} w_i N_i$$

Measure of difficulty of a match at level *i*

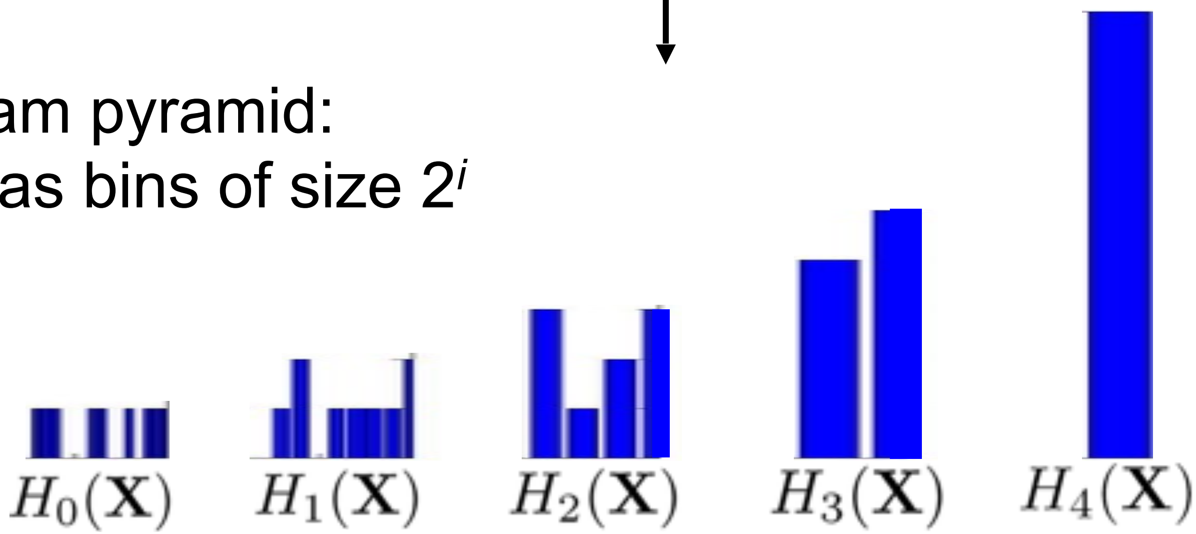
Feature extraction

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}, \quad \vec{\mathbf{x}}_i \in \Re^d$$



$d = 1$

Histogram pyramid:
level *i* has bins of size 2^i

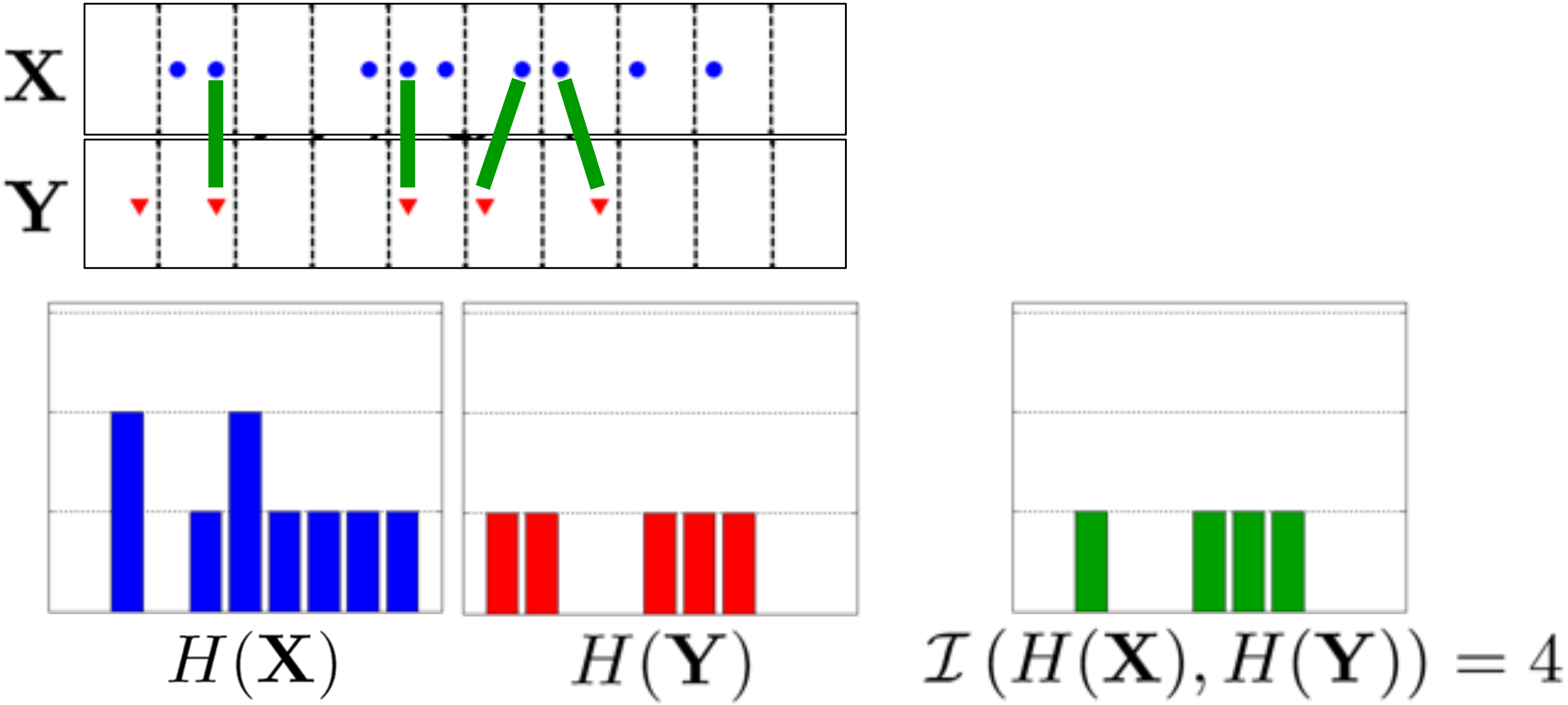


$$\Psi(\mathbf{X}) = [H_0(\mathbf{X}), \ldots, H_L(\mathbf{X})]$$

Counting matches

Histogram intersection

$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



Counting new matches

Histogram intersection

$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$

$$N_i = \underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right)}_{\text{matches at this level}} - \underbrace{\mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{matches at previous level}}$$

Difference in histogram intersections across levels counts *number of new pairs* matched

Pyramid match kernel

$$K_\Delta\left(\overbrace{\Psi(\mathbf{X}), \Psi(\mathbf{Y})}^{\text{histogram pyramids}}\right) =$$

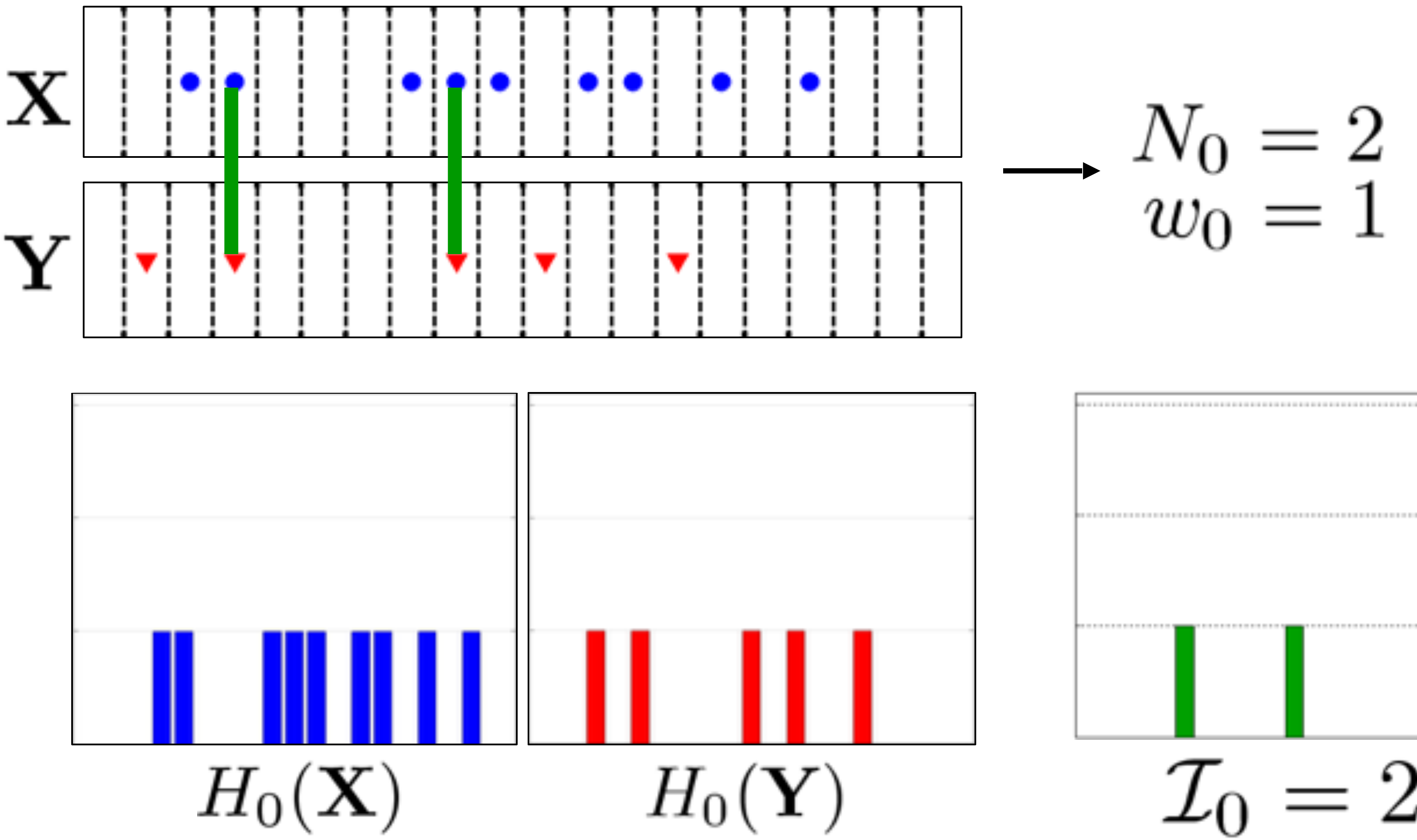
$$\sum_{i=0}^{L} \frac{1}{2^i}\Big(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{number of newly matched pairs at level } i}\Big)$$

measure of difficulty of a match at level *i*

- Weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

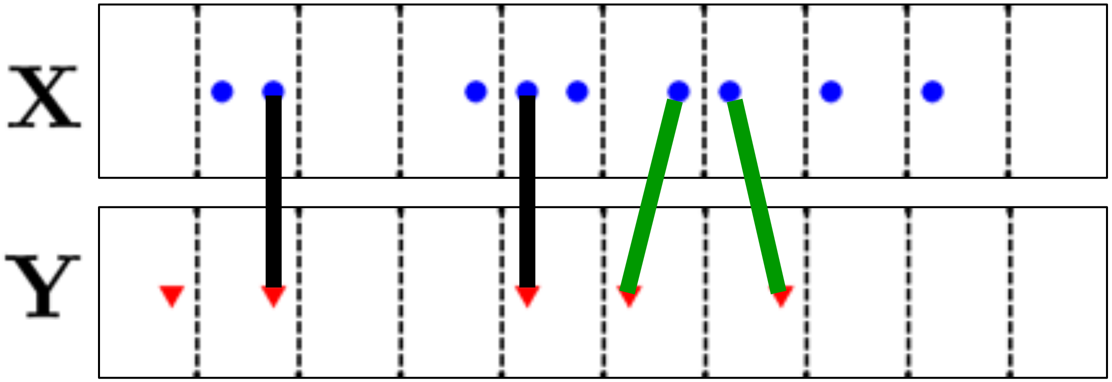
Example pyramid match

Level 0

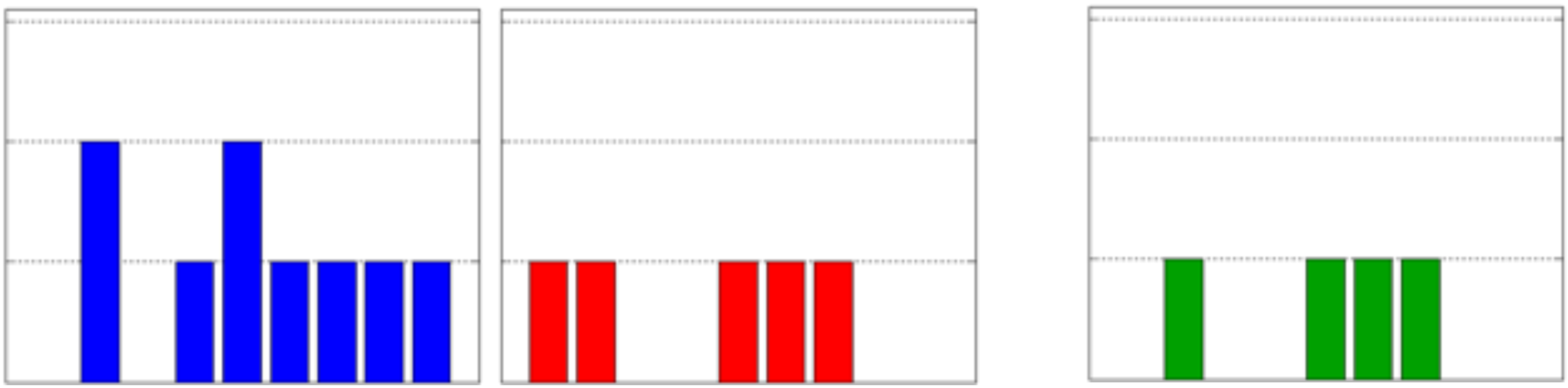


Example pyramid match

Level 1



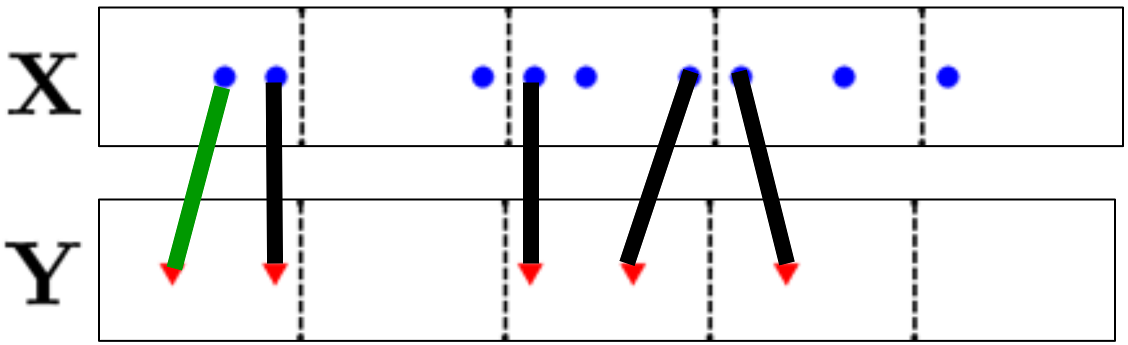
$$N_1 = 4 - 2 = 2$$
$$w_1 = \frac{1}{2}$$



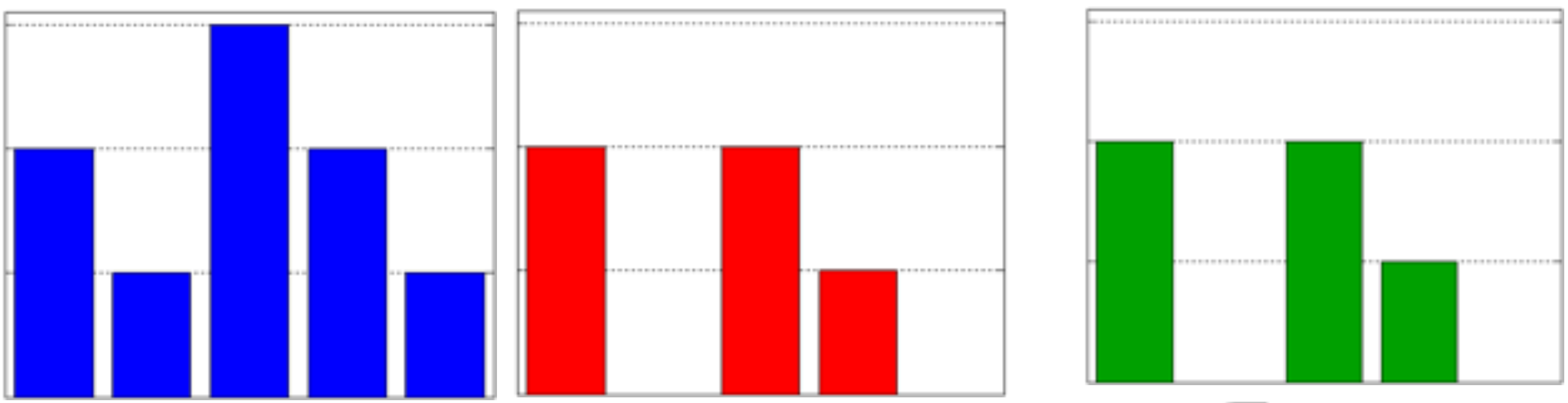
$H_1(\mathbf{X})$ $H_1(\mathbf{Y})$ $\mathcal{I}_1 = 4$

Example pyramid match

Level 2



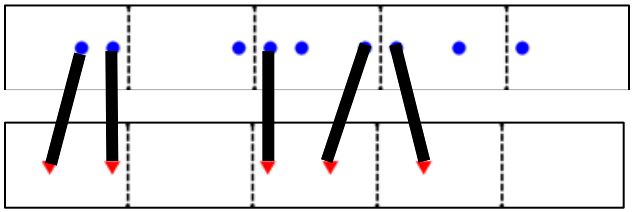
$$N_2 = 5 - 4 = 1$$
$$w_2 = \frac{1}{4}$$



$H_2(\mathbf{X})$ $H_2(\mathbf{Y})$ $\mathcal{I}_2 = 5$

Example pyramid match

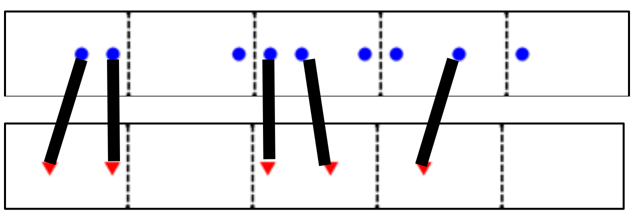
pyramid match



$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

$$= 1(2) + \tfrac{1}{2}(2) + \tfrac{1}{4}(1) = 3.25$$

optimal match



$$K = \max_{\pi : \mathbf{X} \to \mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$

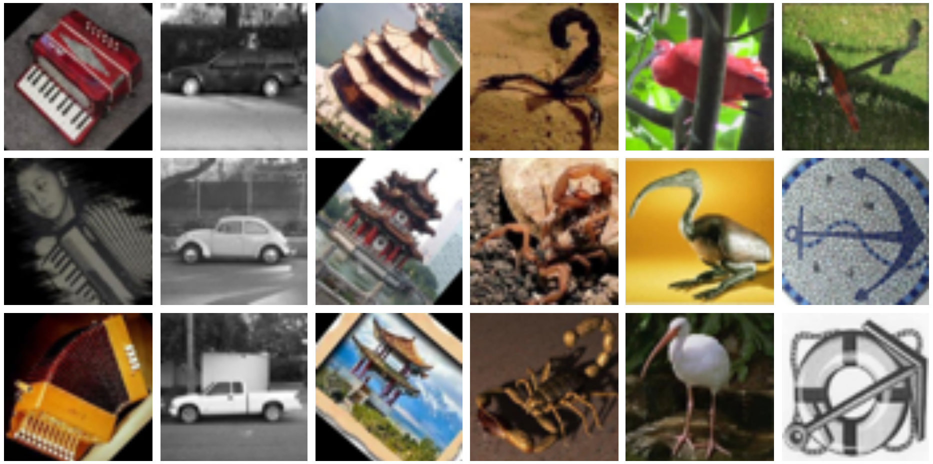
$$= 1(2) + \tfrac{1}{2}(3) = 3.5$$

Scene Classification

L	Single-level	Pyramid
0(1x1)	72.2±0.6	
1(2x2)	77.9±0.6	79.0 ±0.5
2(4x4)	79.4±0.3	*81.1 ±0.3*
3(8x8)	77.2±0.4	80.7 ±0.3

Retrieval Examples

Category classification - CalTech101

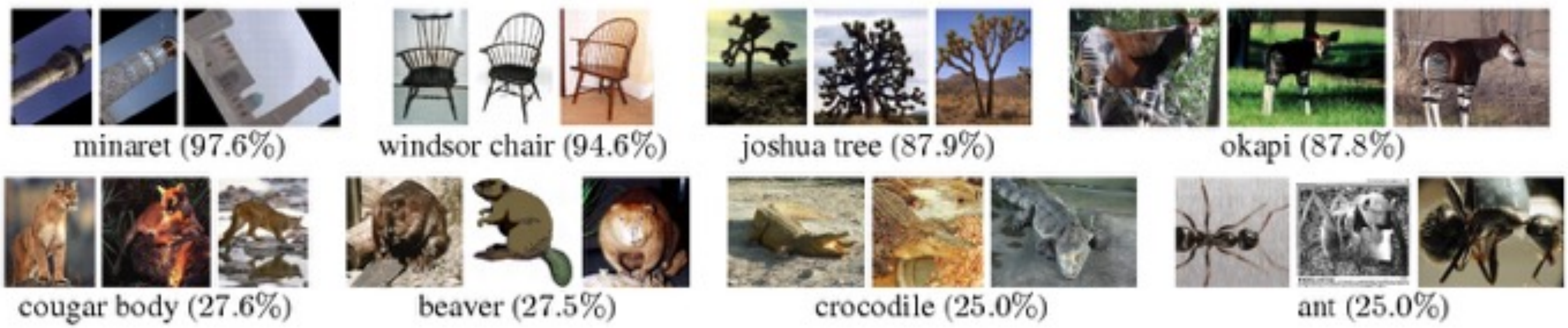


L	Single-level	Pyramid
0(1x1)	**41.2±1.2**	
1(2x2)	**55.9±0.9**	**57.0 ±0.8**
2(4x4)	**63.6±0.9**	***64.6 ±0.8***
3(8x8)	**60.3±0.9**	**64.6 ±0.7**

Bag-of-words approach by Zhang et al.'07: 54 %

CalTech101

Easiest and hardest classes



- Sources of difficulty:
  - Lack of texture
  - Camouflage
  - Thin, articulated limbs
  - Highly deformable shape

Discussion

- Summary
  - Spatial pyramid representation: appearance of local image patches + coarse global position information
  - Substantial improvement over bag of features
  - Depends on the similarity of image layout
- Extensions
  - Integrating different types of features, learning weights, use of different grids
  - Flexible, object-centered grid

Roadmap (this lecture)

- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching

Roadmap (this lecture)

- Image Categorization
- Bag-of-Words (BOW)
- Generative vs. Discriminative Approach
- Spatial Pyramid Matching
- Application: Remote Sensing Image Classification