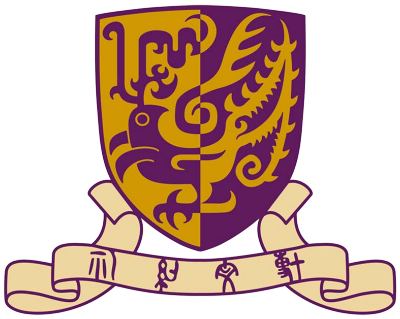
EIE4512 - Digital Image Processing
Histogram of Gradient (HOG) and Object Detection



香 港 中 文 大 學 (深 圳)
The Chinese University of Hong Kong, Shenzhen

Zhen Li
lizhen@cuhk.edu.cn
School of Science and Engineering
The Chinese University of Hong Kong, Shen Zhen
March 26 - 28, 2019

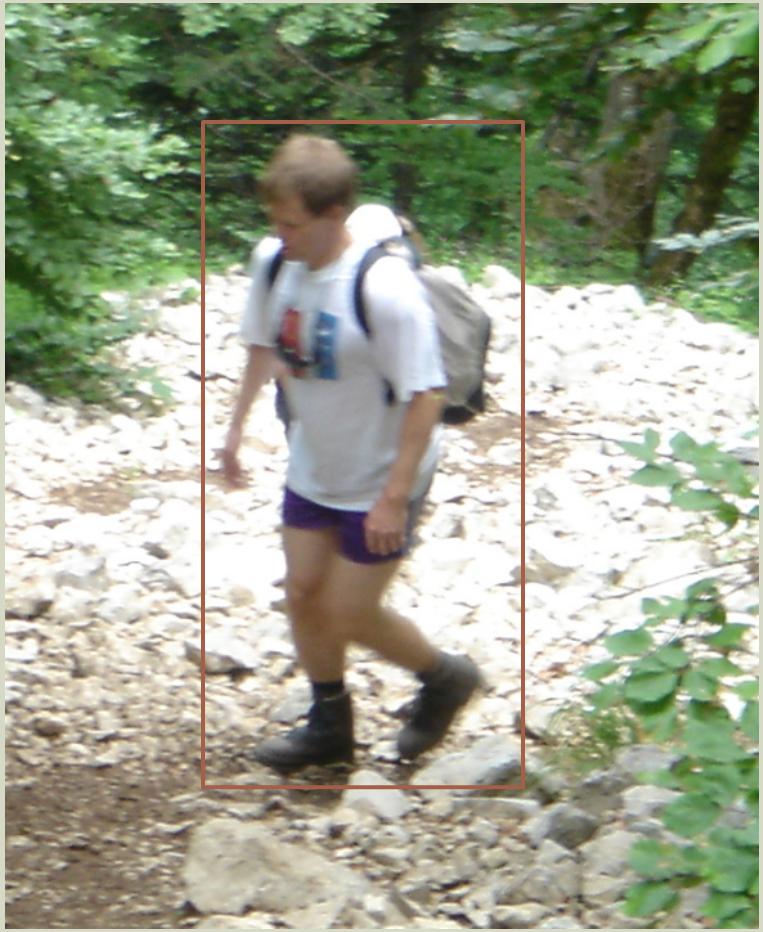
HISTOGRAMS OF ORIENTED GRADIENTS FOR HUMAN DETECTION

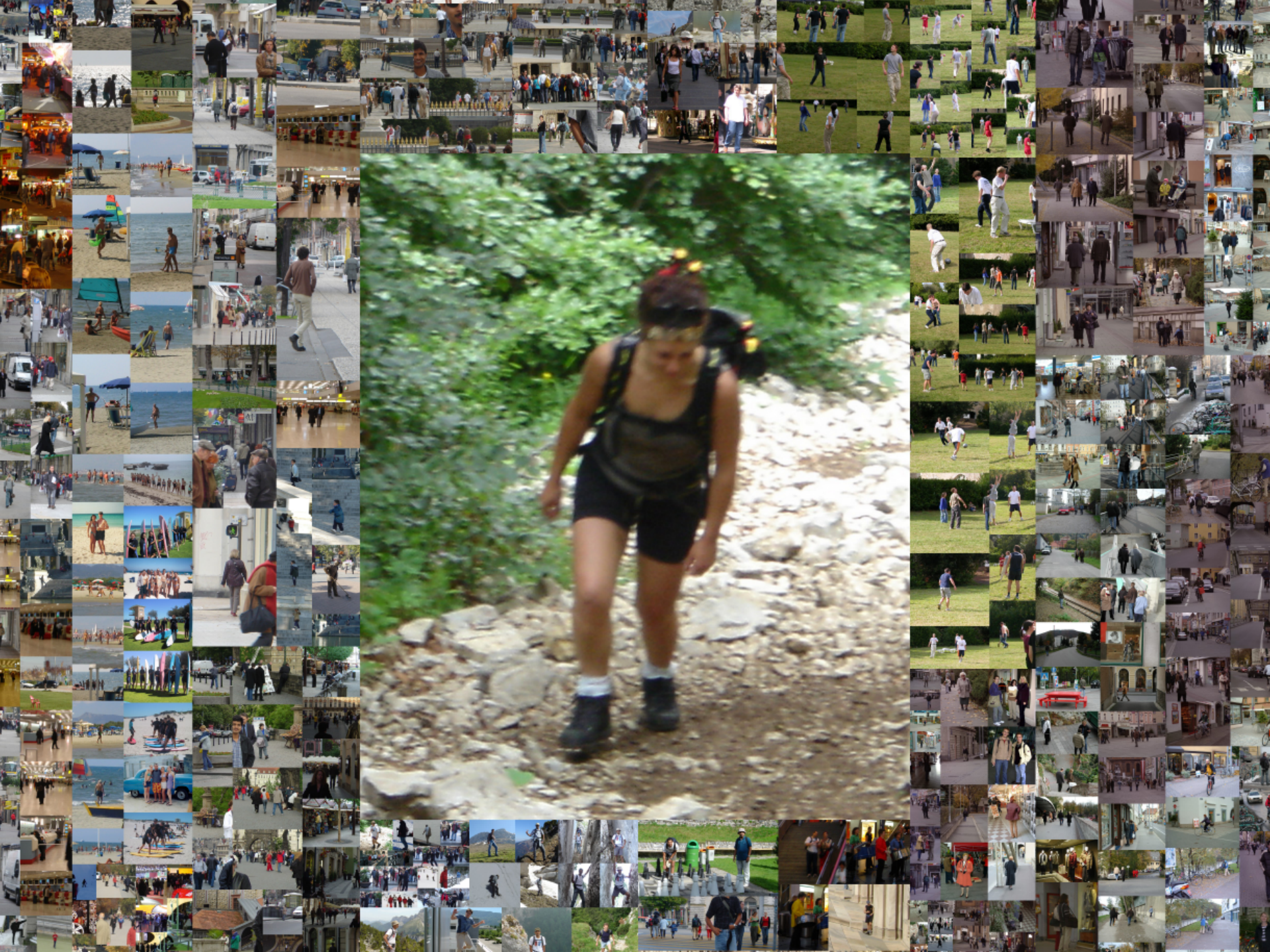
(NAVNEET DALAL AND BILL TRIGGS)

Rafael Cosman

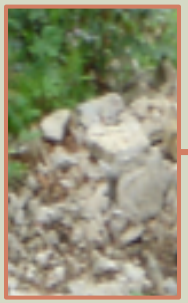
Tao Wang

HUMAN DETECTION





TRAINING SVM



HOG/SIFT/SURF...

Descriptor

$(x_0, x_1 \dots x_n)$

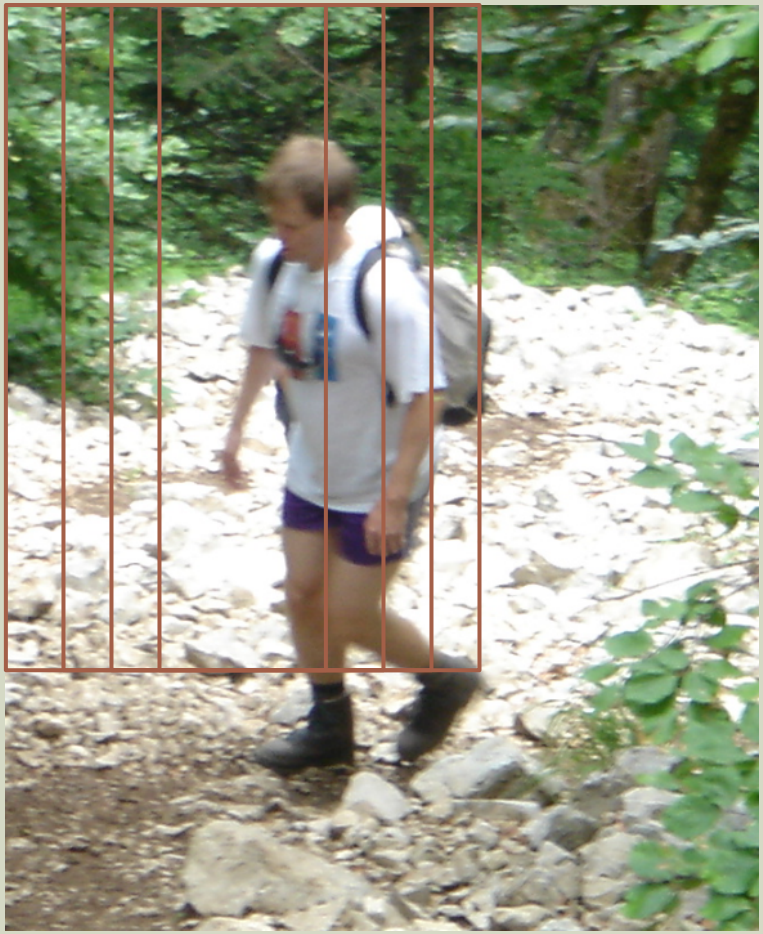
$(x_0, x_1 \dots x_n)$

Class “Person”

SVM

Class “No Person”

TESTING: SCANNING WINDOW



TESTING SVM

$(x_0, x_1 \ldots x_n)$

SVM

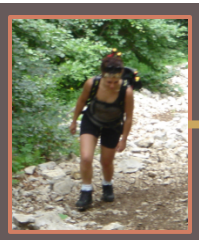
Result

"Person"

"No Person"

MOTIVATION

- Very simple to implement
- Performs as well or better than many descriptors
- Cited over 5000 times
  - Basis for Deformable Parts Model

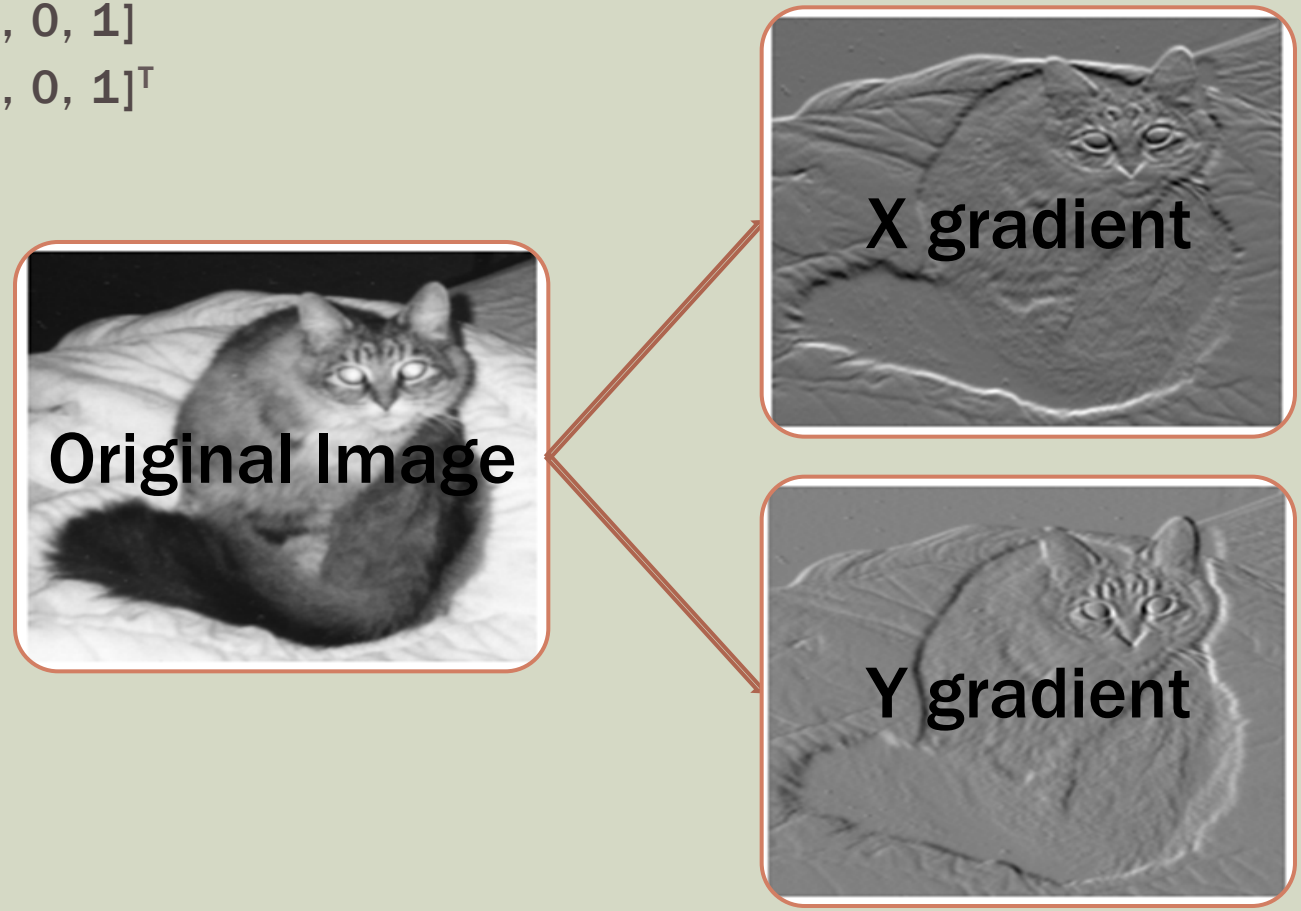


HOG

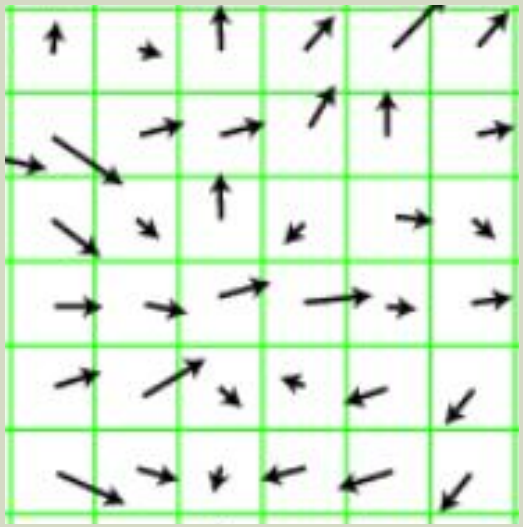
$(x_0, x_1 \ldots x_n)$

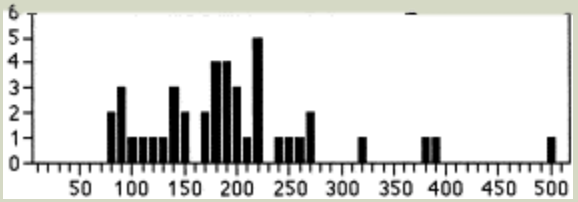
Image → Gradients → Cells → Blocks → Descriptor

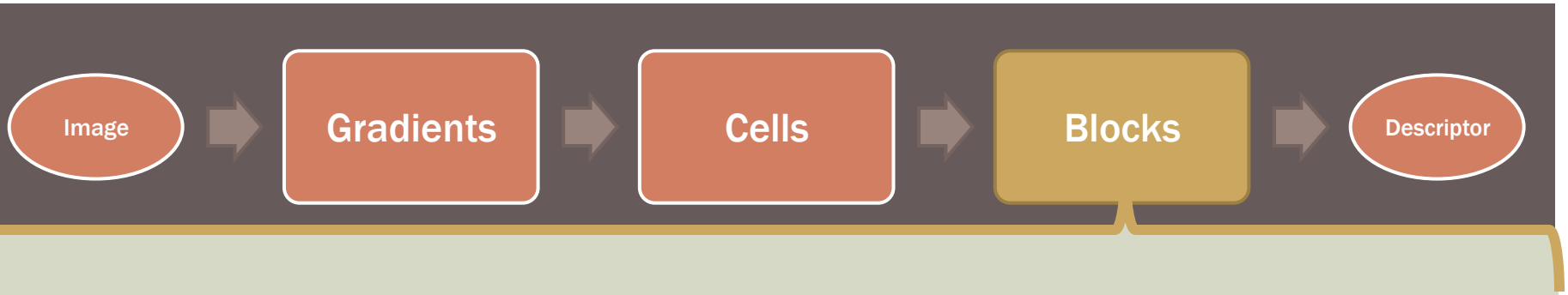
- Convolve the image with discrete derivative mask
  - [-1, 0, 1]
  - $[-1, 0, 1]^T$

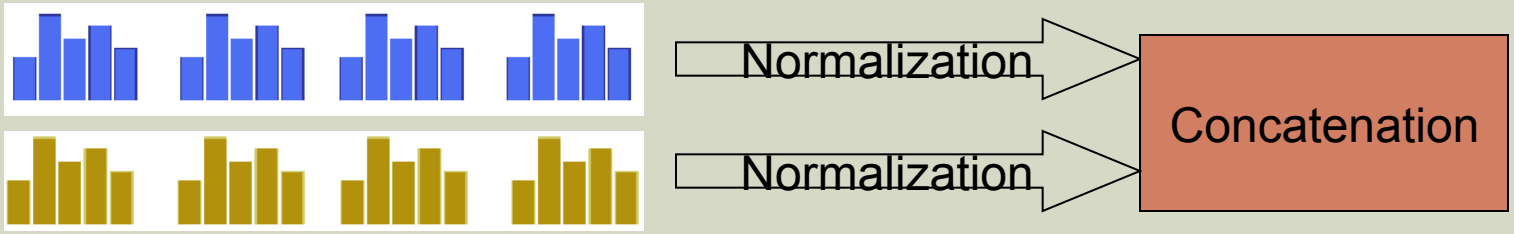


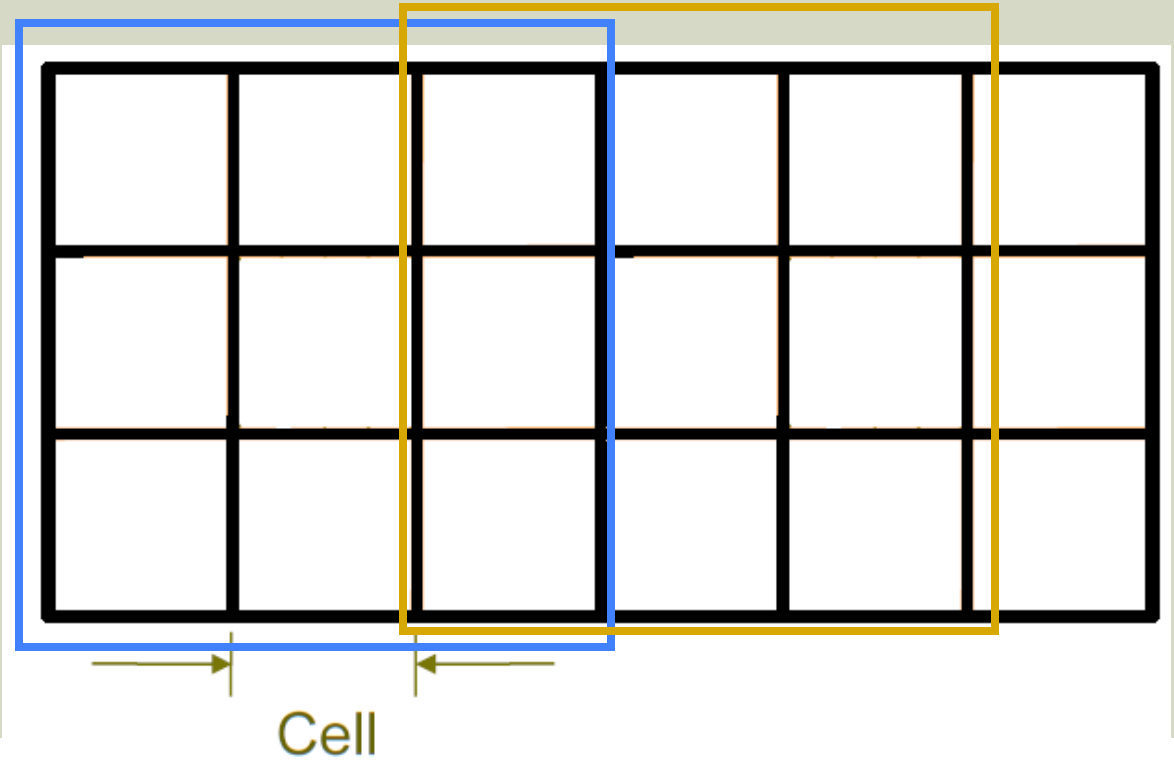
- Now we count up the gradient angles in 8x8 cells
  - Vote weight = magnitude = $\sqrt{dx\hat{}2 + dy\hat{}2}$
  - Who you vote for ~ angle = $\arctan(dy/dx)$







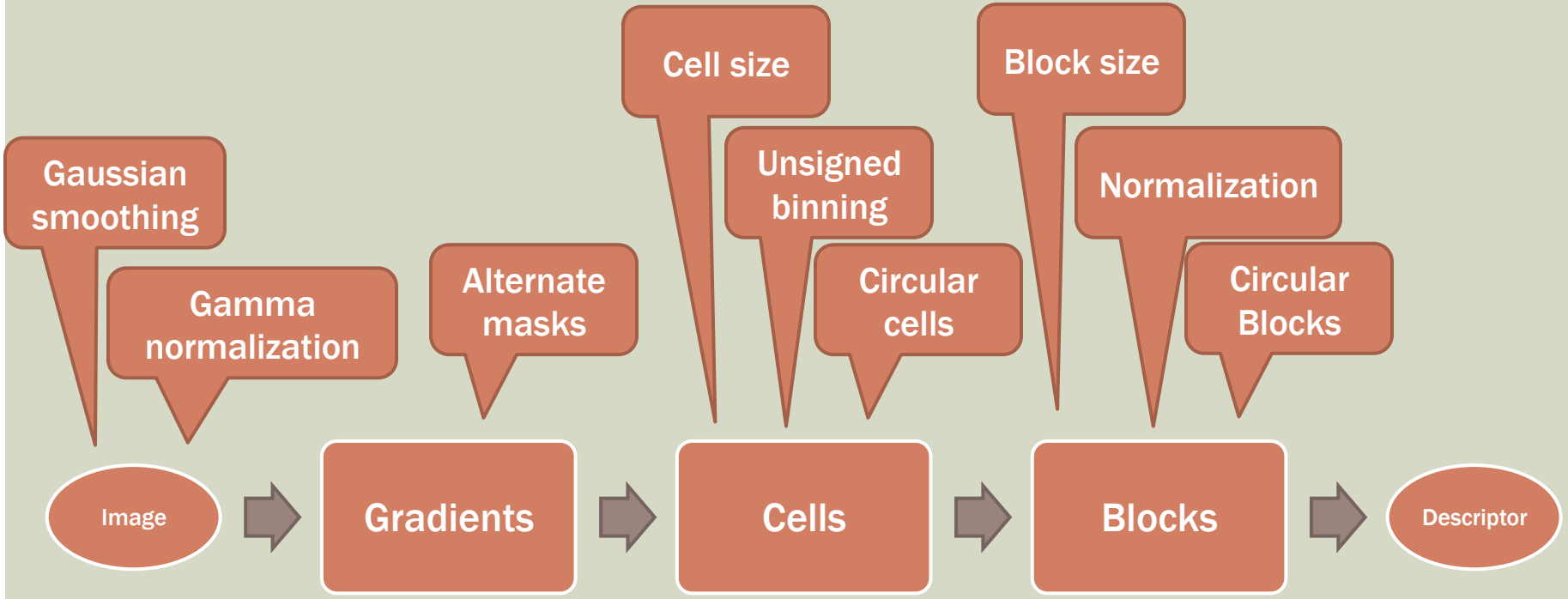




Overlapping blocks yields better results!

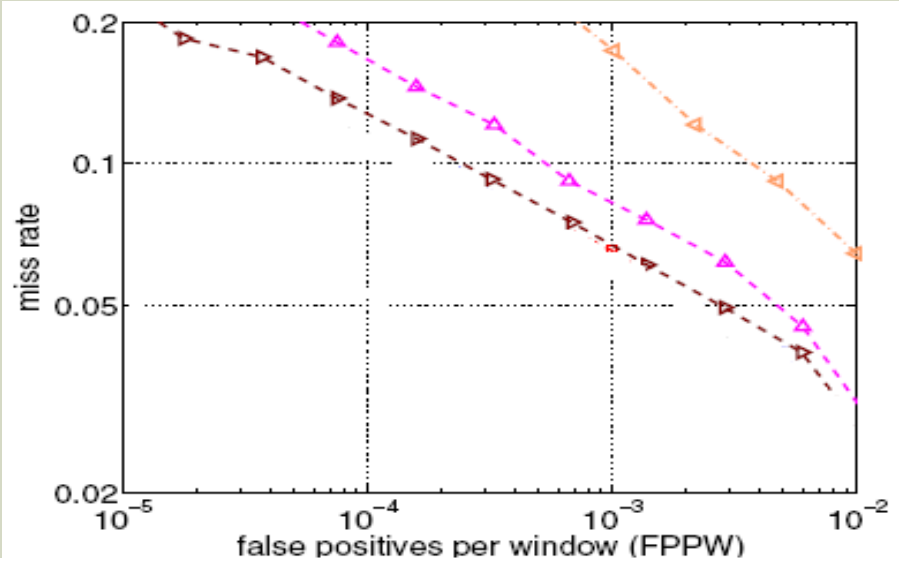
(Overcomplete features)

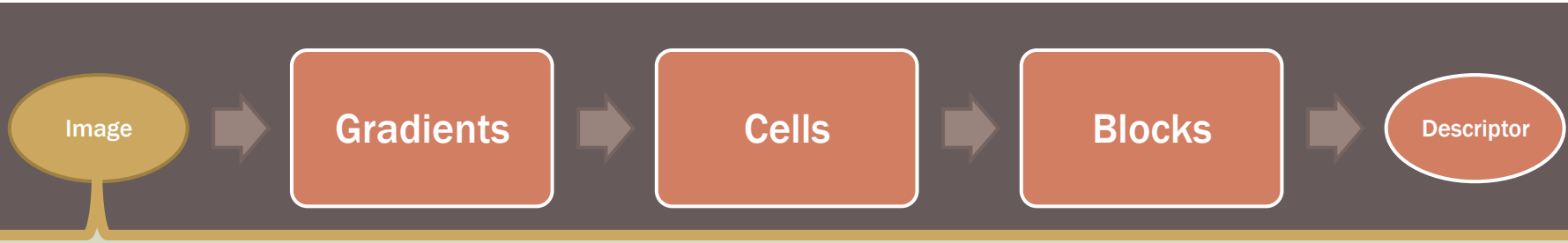
VARIATIONS


Gaussian smoothing
Gamma normalization
Alternate masks
Cell size
Unsigned binning
Circular cells
Block size
Normalization
Circular Blocks
Image
Gradients
Cells
Blocks
Descriptor

EVALUATION METRIC

- Detection task, so a single accuracy value does not make sense
  - Low threshold – Low miss rate, but many false positives
  - High threshold – Few false positives, but misses a lot
- **Detection Error Tradeoff (DET) Curves**
  - Vertical Axis: Miss Rate = 1-Recall = (1 – True Pos / Total Pos in GT)
  - Horizontal Axis: False Positive Per Window (FPPW)

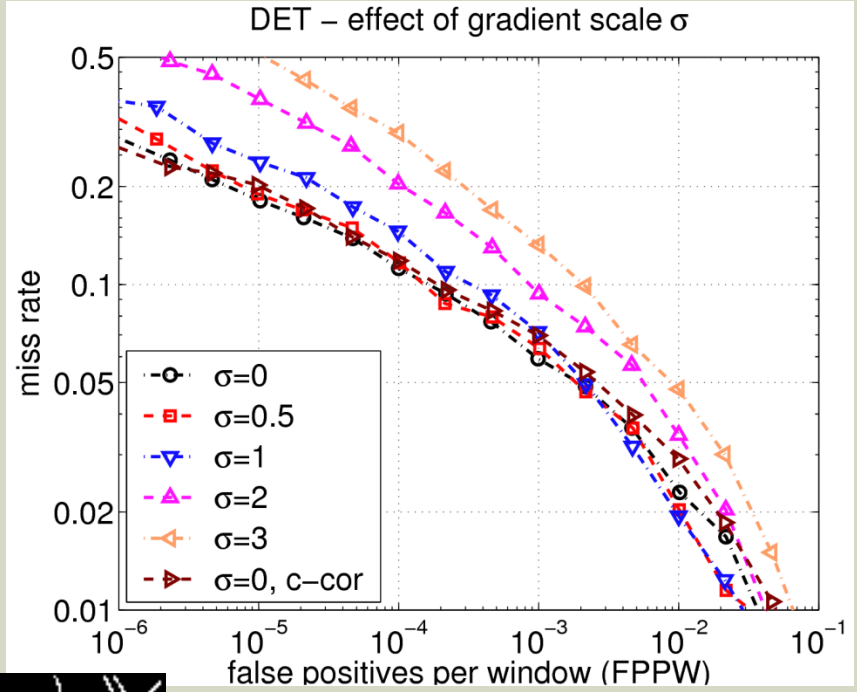




- Using Grayscale/Color
  - Compute gradients in all color channels, pick the highest one
  - Using color gives slightly better results

- Gamma/Color Normalization
  - As a preprocessing step
  - Has little effect on results

- Gaussian smoothing
  - Reduces performance
  - Fine-grained edge detection is crucial




raw image

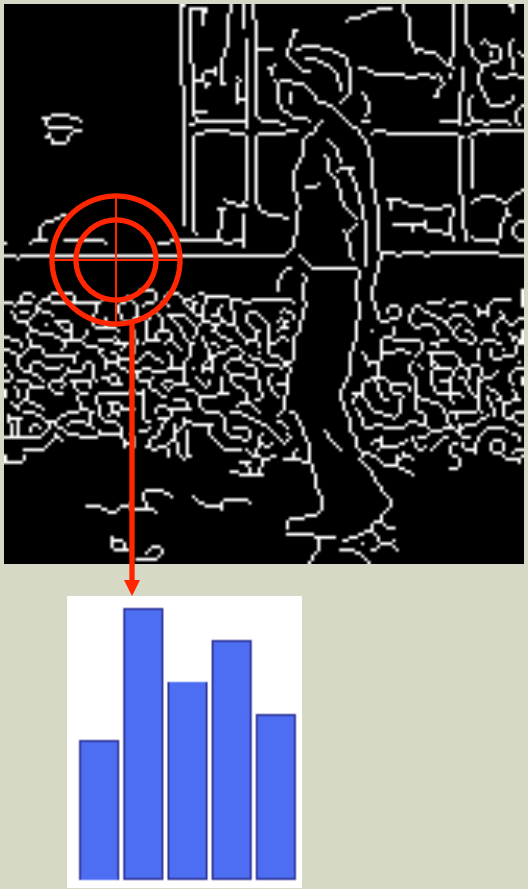

σ>0


σ=0

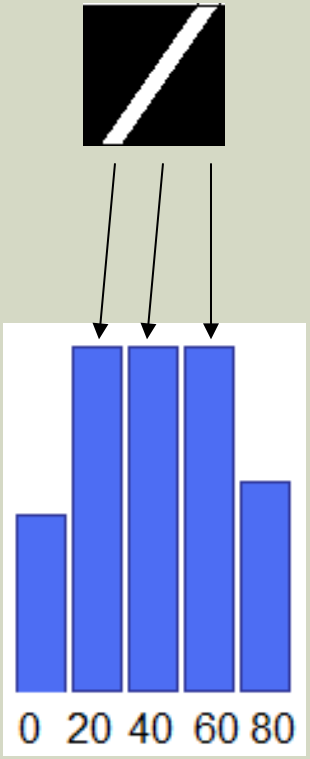
- Choice of masks
  - 1-D centered: $[-1, 0, 1]$, $[-1, 0, 1]^T$
  - 1-D uncentered: $[-1, 1]$, $[-1, 1]^T$
  - 1-D Cubic corrected: $[1, -8, 0, 8, -1]$, $[1, -8, 0, 8, -1]^T$
  - 2-D Sobel mask

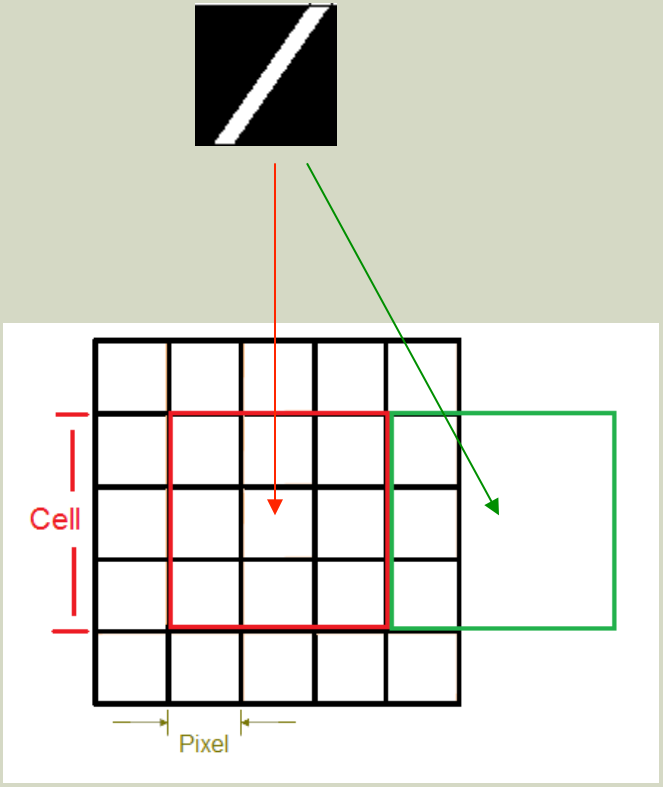
- 1-D centered $[-1, 0, 1]$, $[-1, 0, 1]^T$ with no Gaussian smoothing works best

- Rectangular vs Circular cells

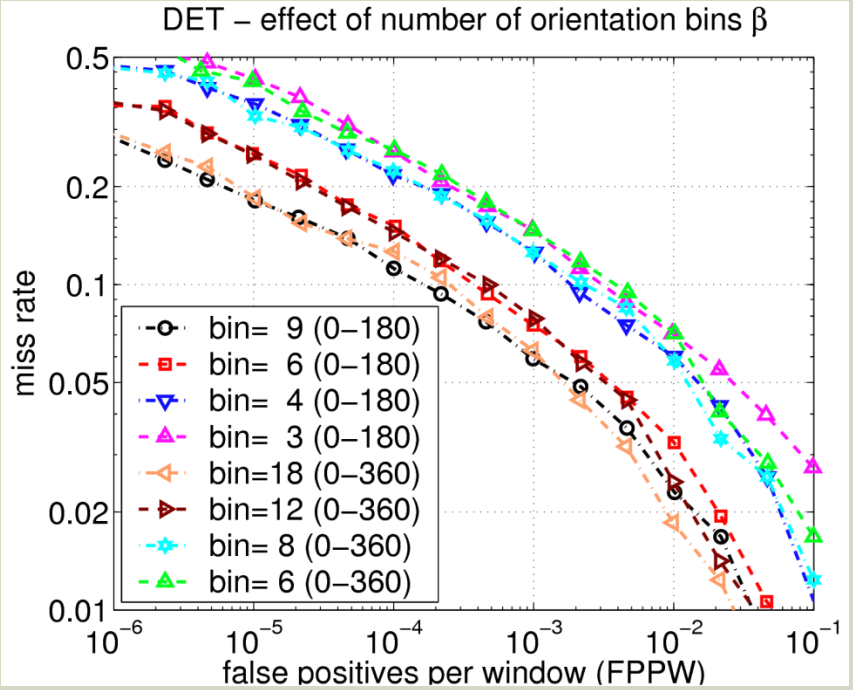


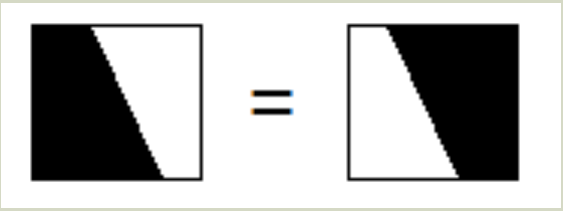
- Bilinear interpolation to reduce aliasing
- Across both orientations and locations (weighted by distance in pixels)





- # of orientation bins
  - Increasing orientation bins from 4 to 9 decreases false positives by 10 times
- Unsigned cells
  - 0-180 degrees instead of 0-360 degrees
  - Actually improves performance slightly!
    - Why?





- Motivation
  - Gradient magnitude lacks invariance to changes in illumination and foreground background contrast.

- Need local contrast normalization to find the “true” weight of an edge

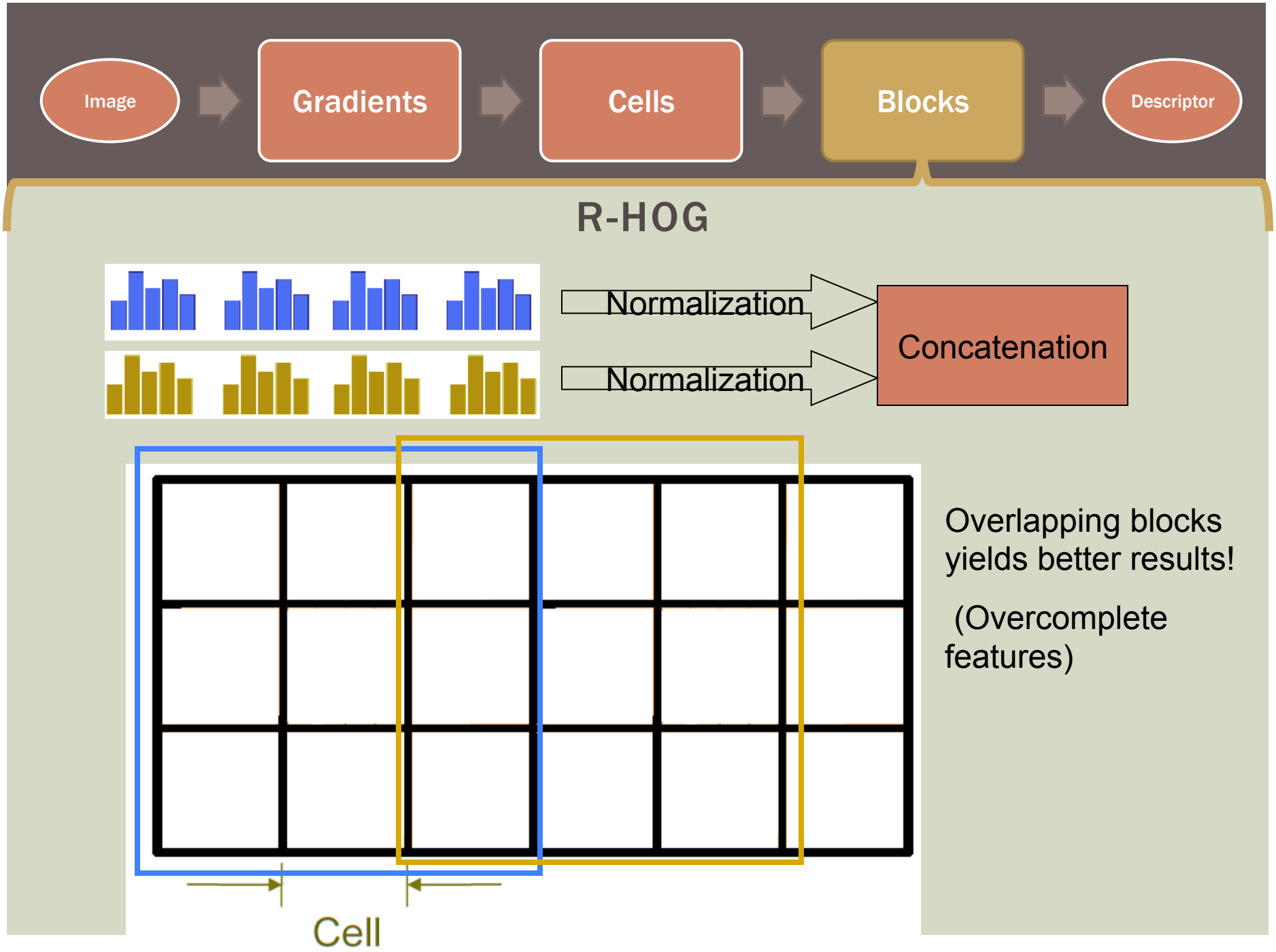
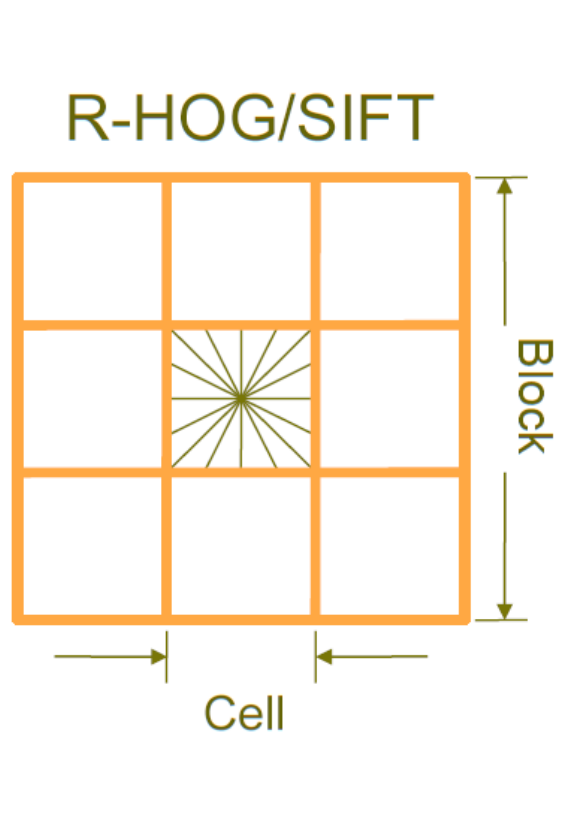
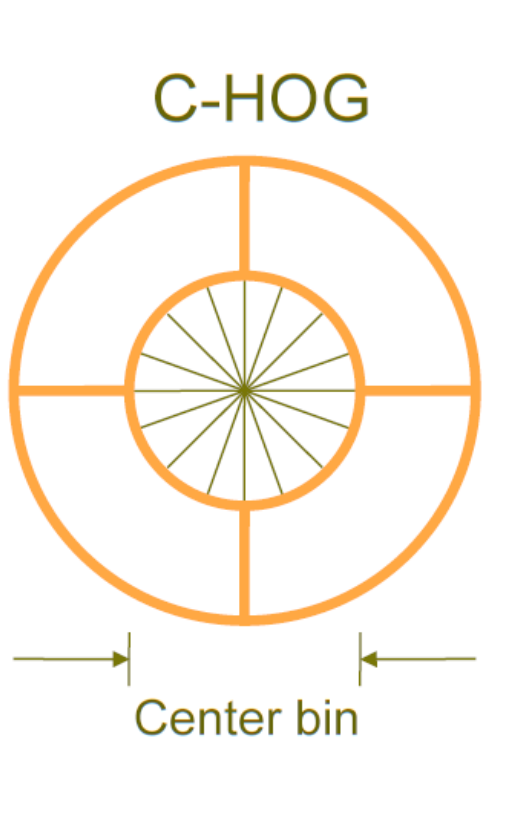


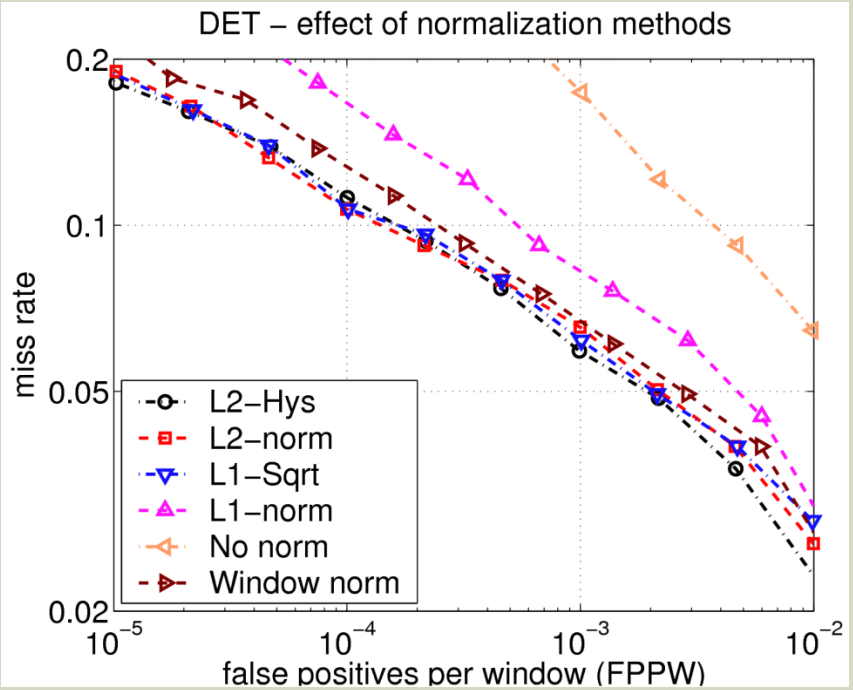
Image
Gradients
Cells
Blocks
Descriptor
R-HOG
Normalization
Normalization
Concatenation
Cell
Overlapping blocks yields better results!
(Overcomplete features)

R-HOG versus C-HOG

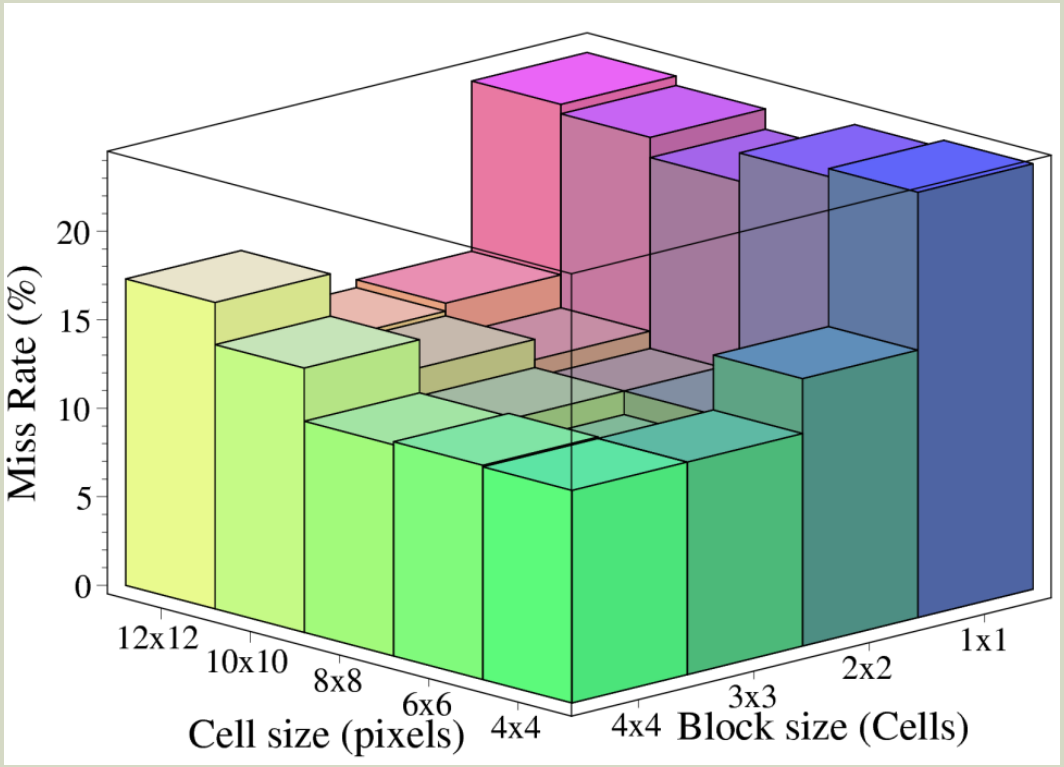




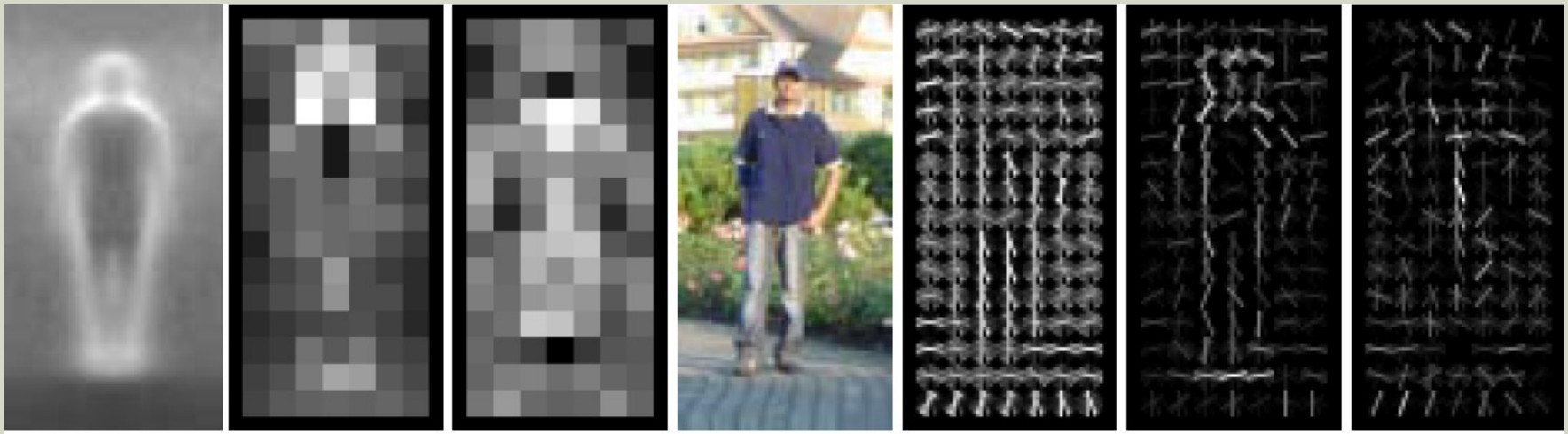
- Normalizations
  - L2-norm
  - L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing
  - L1-norm:
  - L1-sqrt:



- Best cell & block sizes:
  - Cell size of 6x6
  - Block size of 3x3

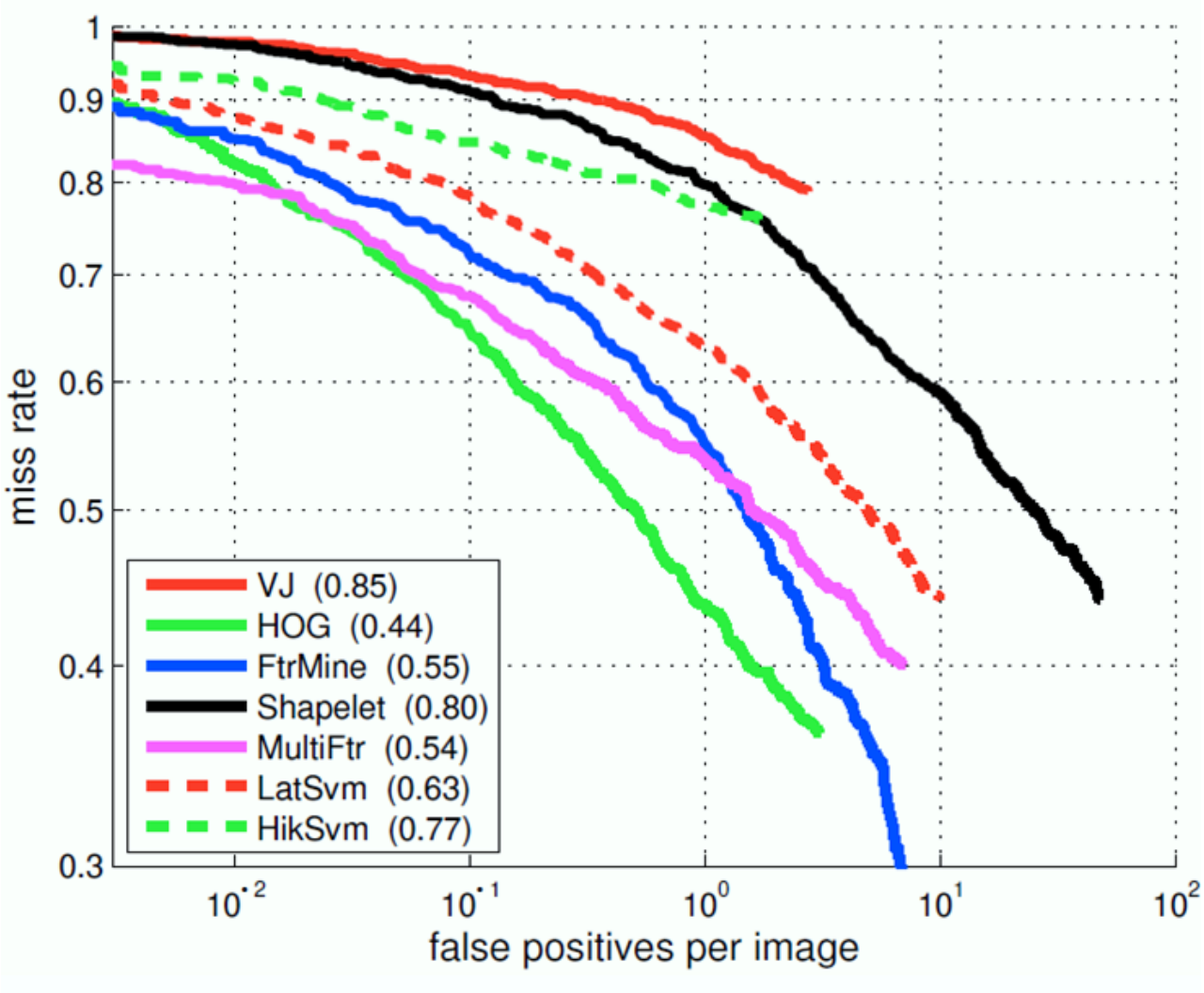


VISUALIZATION AND INSIGHTS

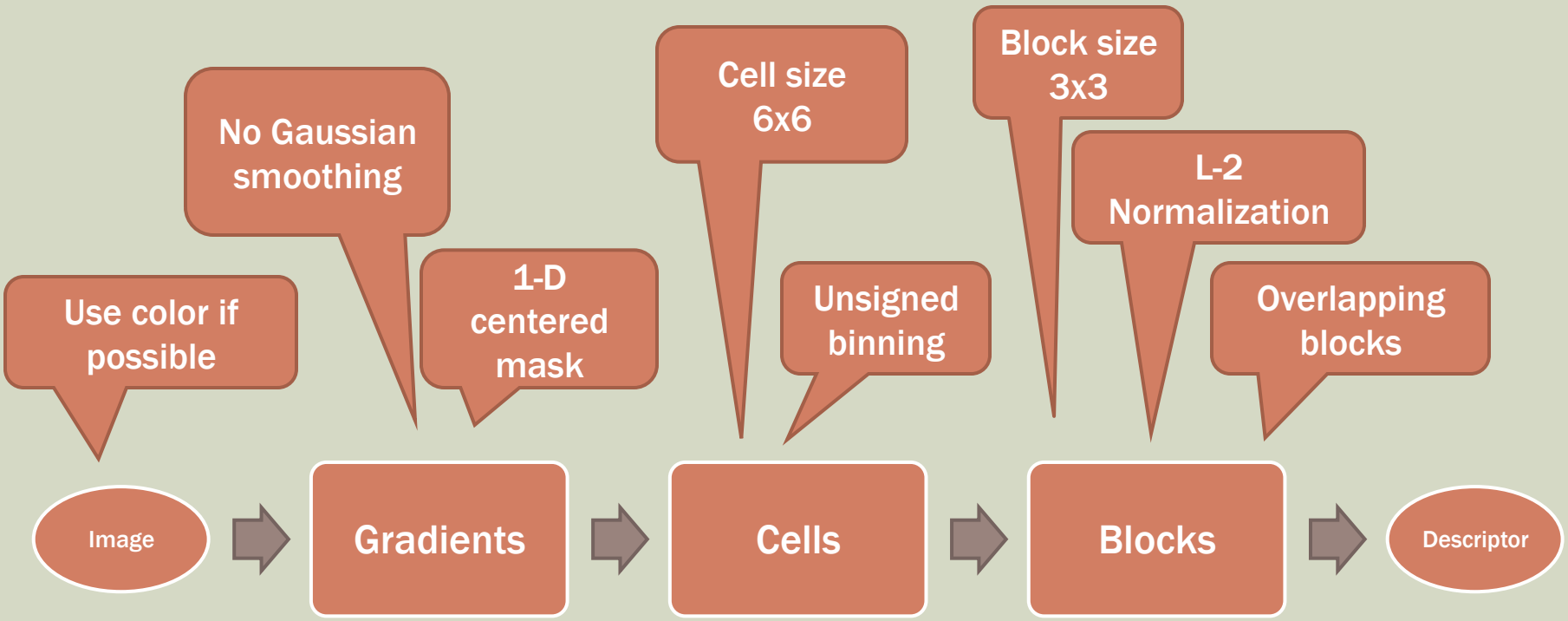


a. Average gradient over positive examples
b. Maximum positive SVM weight in each block
c. Maximum negative SVM weight in each block
d. A test image
e. It's R-HOG descriptor
f. R-HOG descriptor weighted by positive SVM weights
g. R-HOG descriptor weighted by negative SVM weights

HOG COMPARED TO OTHERS



BEST SETUP


Use color if possible
No Gaussian smoothing
1-D centered mask
Cell size 6x6
Unsigned binning
Block size 3x3
L-2 Normalization
Overlapping blocks
Image
Gradients
Cells
Blocks
Descriptor

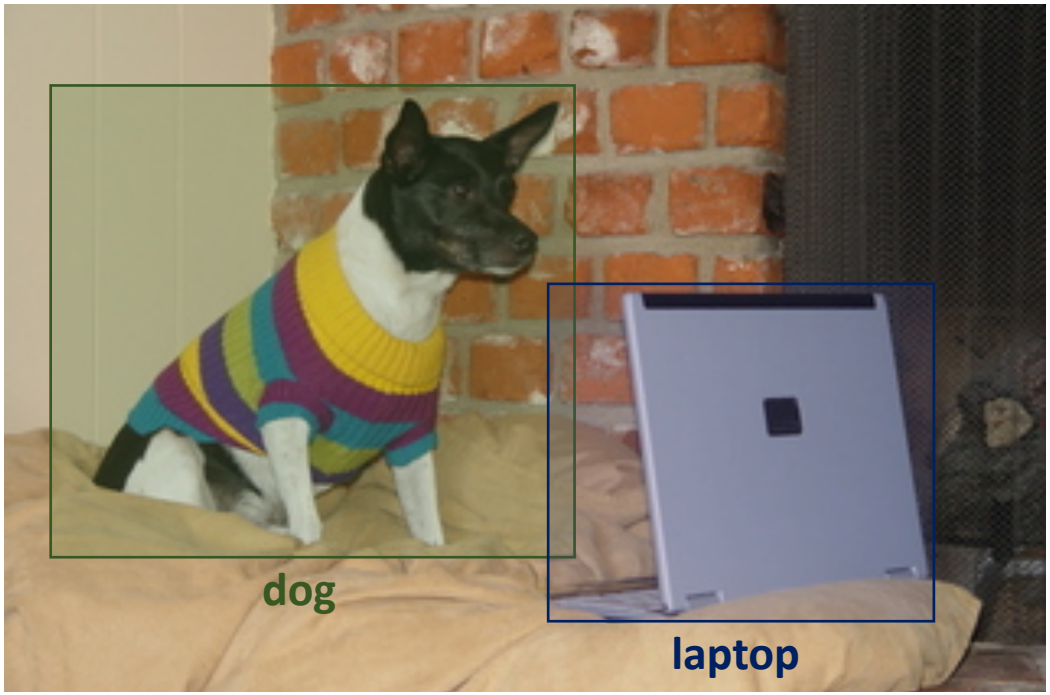
RULES OF THUMB

- Abrupt edges at fine scales are essential
  - No blurring
- Local contrast normalization is essential
- Overlapping blocks w/ “redundant” information improves results significantly.
- Fine orientation quantization is more important than fine spatial orientation

Introduction to other Object Dectection Algorithms

What is Object Recognition

- What would you see from this image?

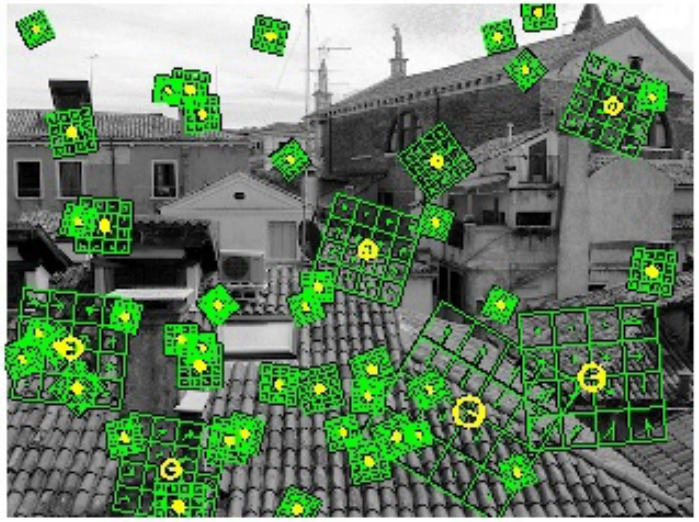


Object recognition: identifying the categories of objects in an image or video.

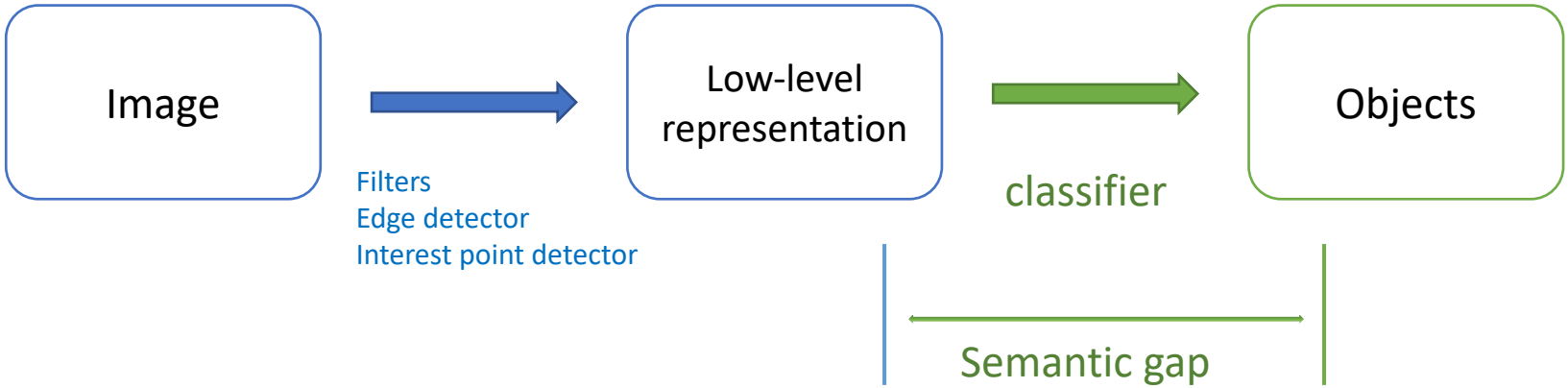
What we learned

Edges

Interest Points

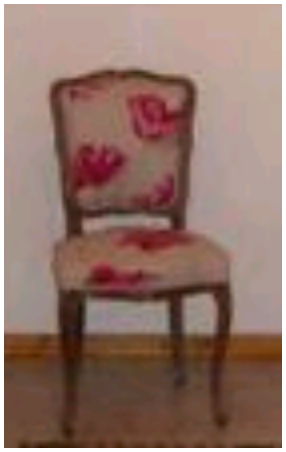


Overall Pipeline

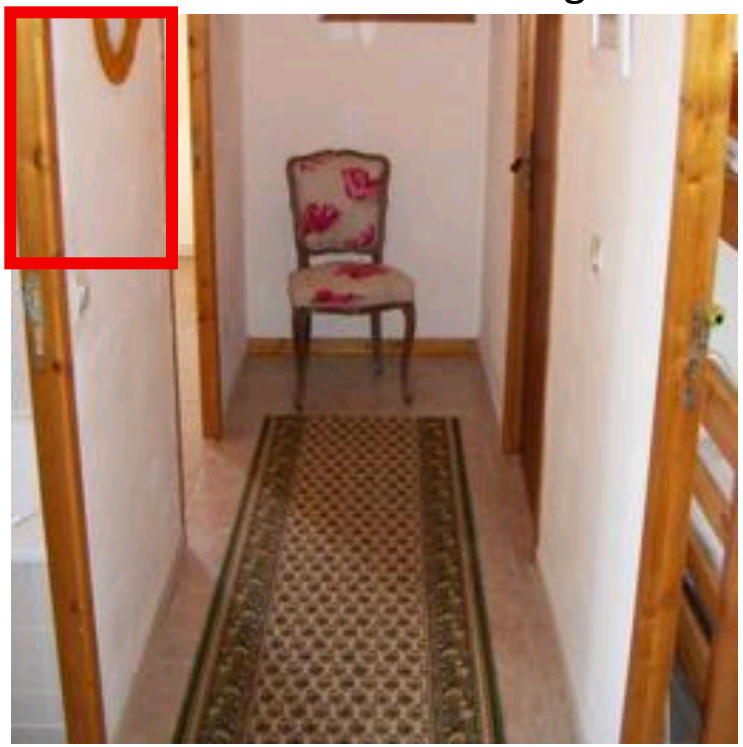


Is Object Recognition Hard?

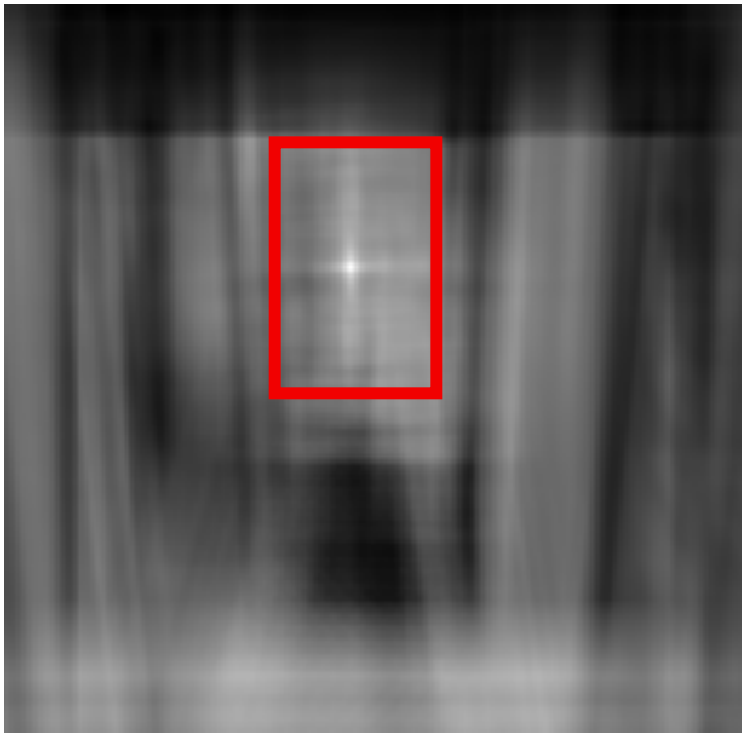
This is a chair



Find the chair in this image

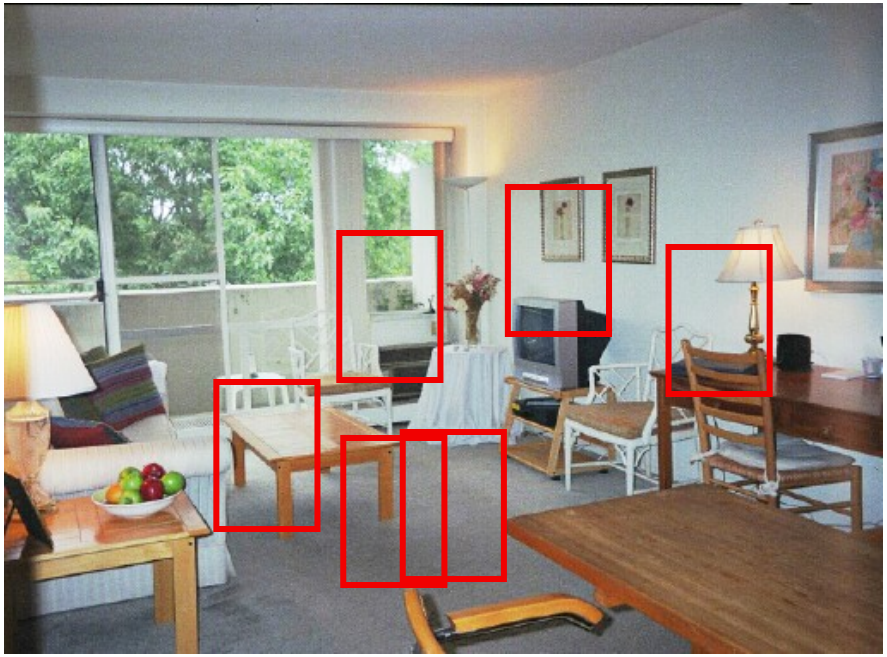


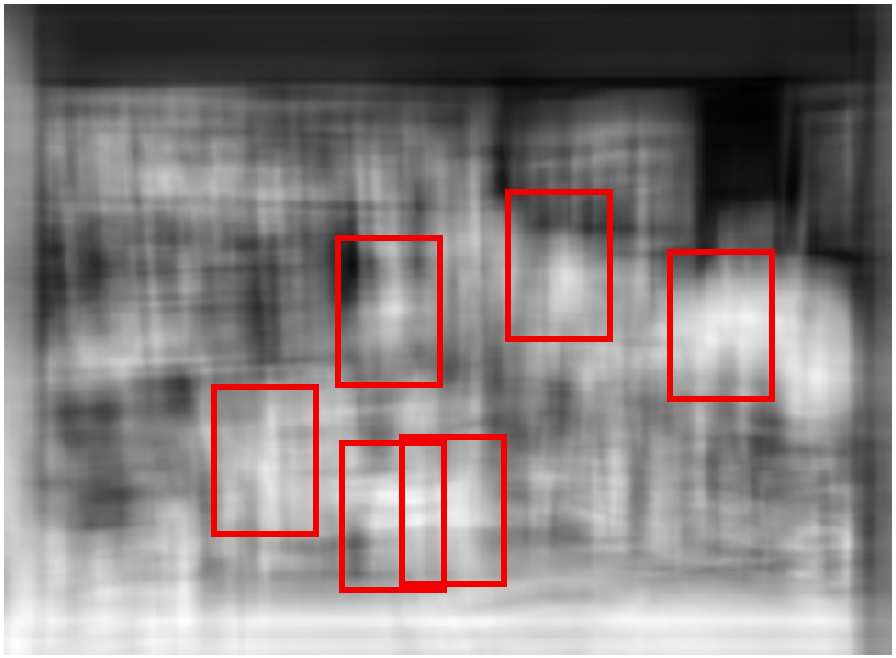
Output of normalized correlation

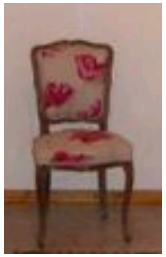


Is Object Recognition Hard?

Find the chair in this image







Pretty much garbage

Simple template matching is not going to make it

Is Object Recognition Hard?

Find the chair in this image

A “popular method is that of template matching, by point to point correlation of a model pattern with the image pattern. These techniques are inadequate for three-dimensional scene analysis for many reasons, such as occlusion, changes in viewing angle, and articulation of parts.” Nivatia & Binford, 1977.

It can get a lot harder

Brady, M. J., & Kersten, D. (2003). Bootstrapped learning of novel objects. J Vis, 3(6), 413-422

Towards Object Recognition

- Develop an image representation
  - Bag of Words (BoW)
  - Part-based models

- Develop a classifier
  - K Nearest Neighbors
  - Metrics to measure distances

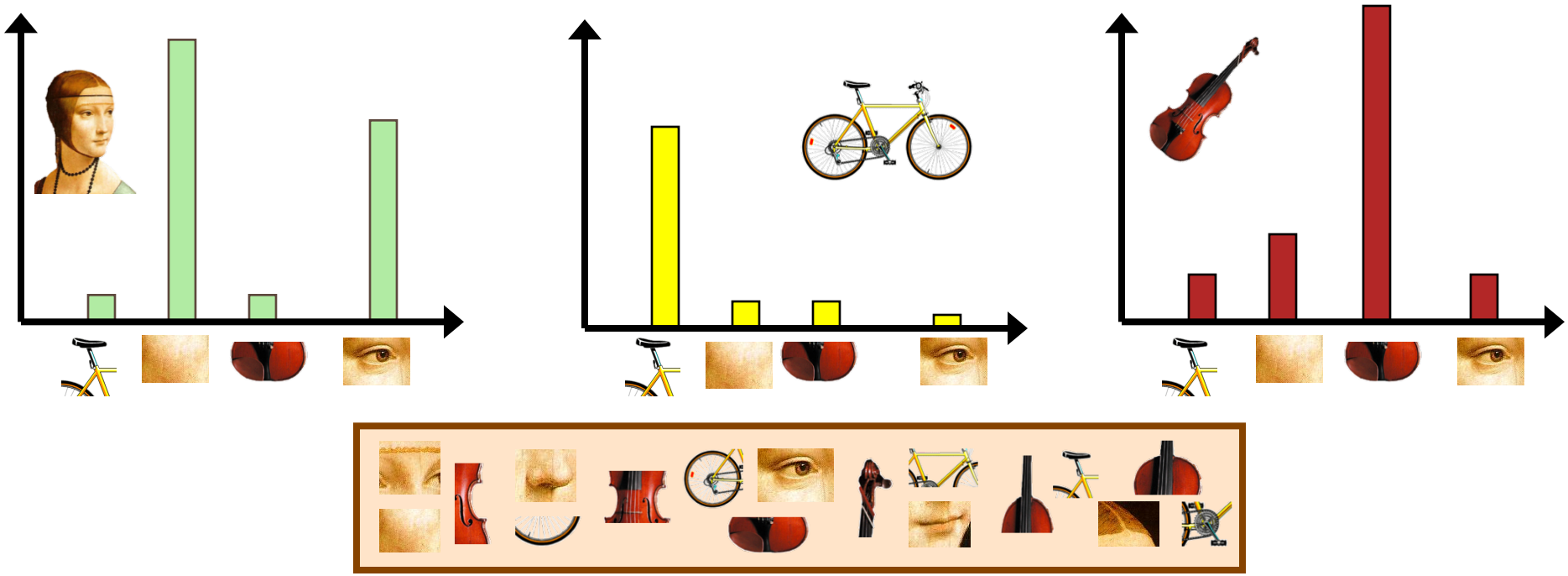
Bag of Words Models

The slides for Bag-of-Words are adapted from those by Rob Fergus (NYU).

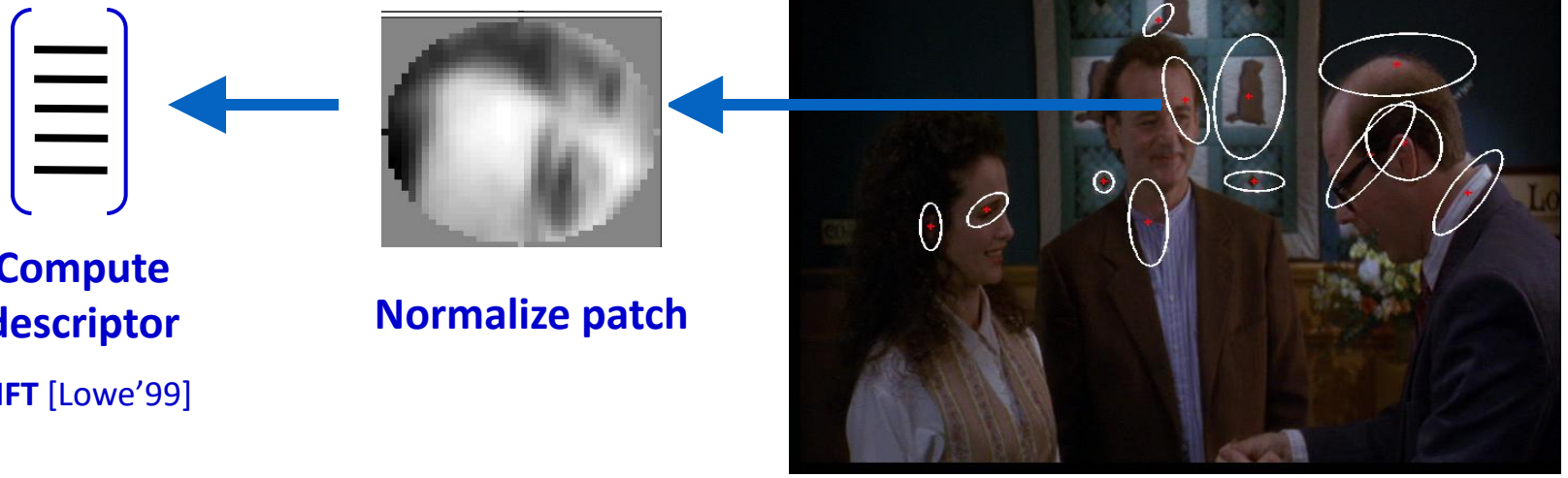
Object → Bag of ‘words’

Bag of Words

- Independent features
- Histogram representation



Step 1: Feature **detection and** representation



Compute descriptor

e.g. **SIFT** [Lowe'99]

Normalize patch

Detect patches

[Mikojaczyk and Schmid '02]

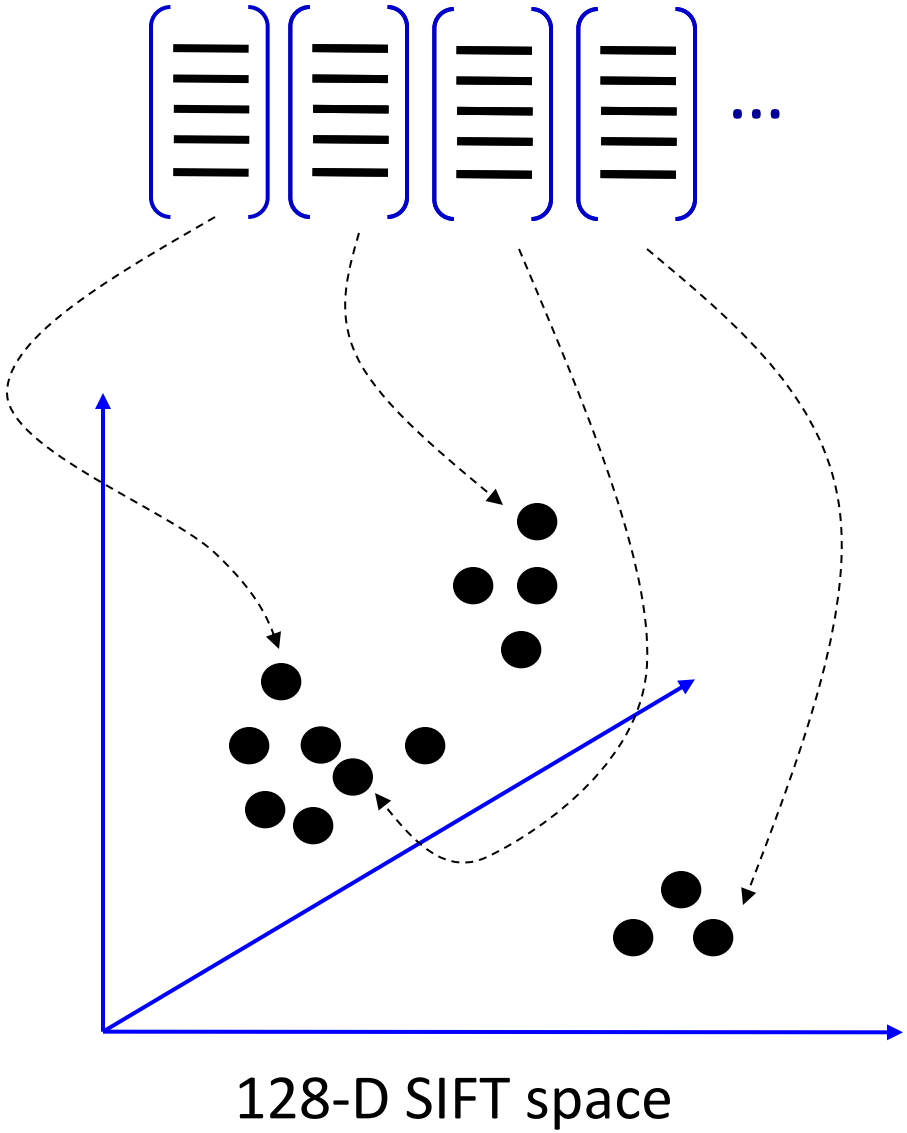
[Mata, Chum, Urban & Pajdla, '02]

[Sivic & Zisserman, '03]

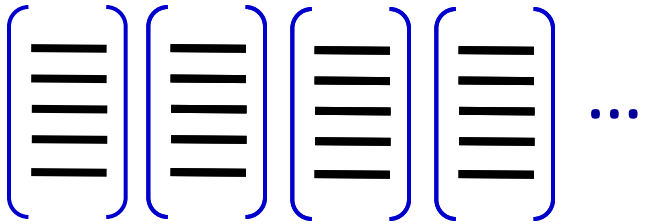
Local interest operator
or
Regular grid

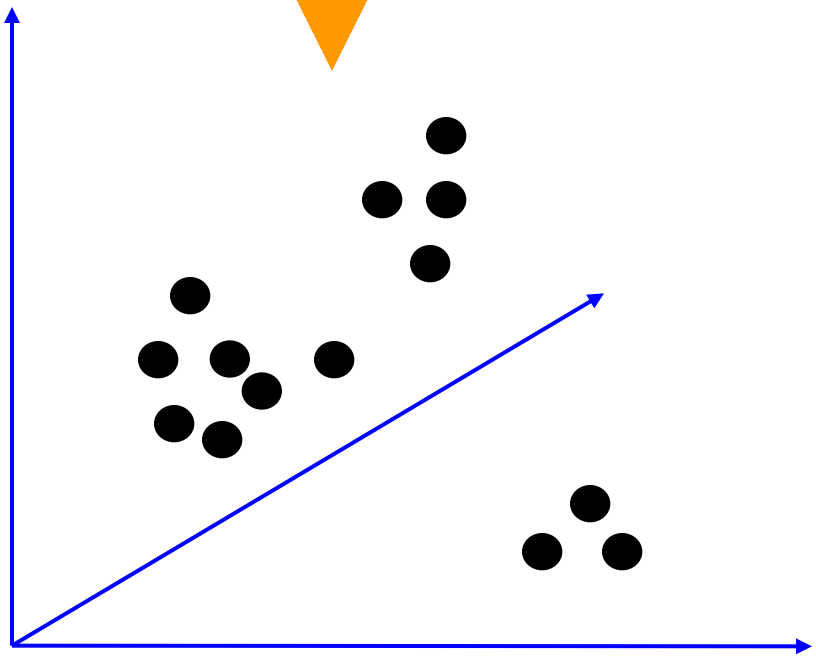
Slide credit: Josef Sivic

Step 2: Codewords dictionary formation

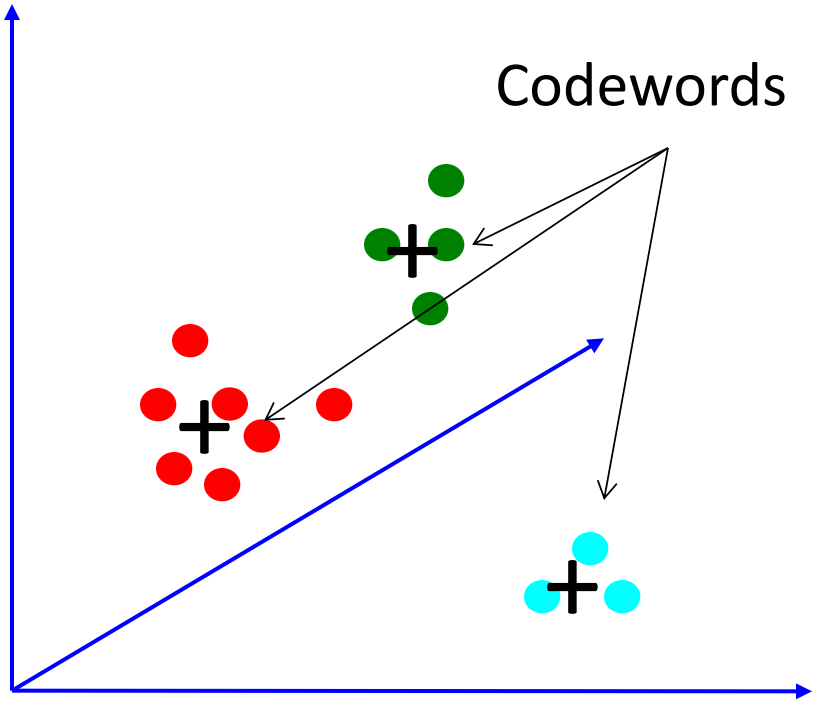


2. Codewords dictionary formation





128-D SIFT space



Vector quantization

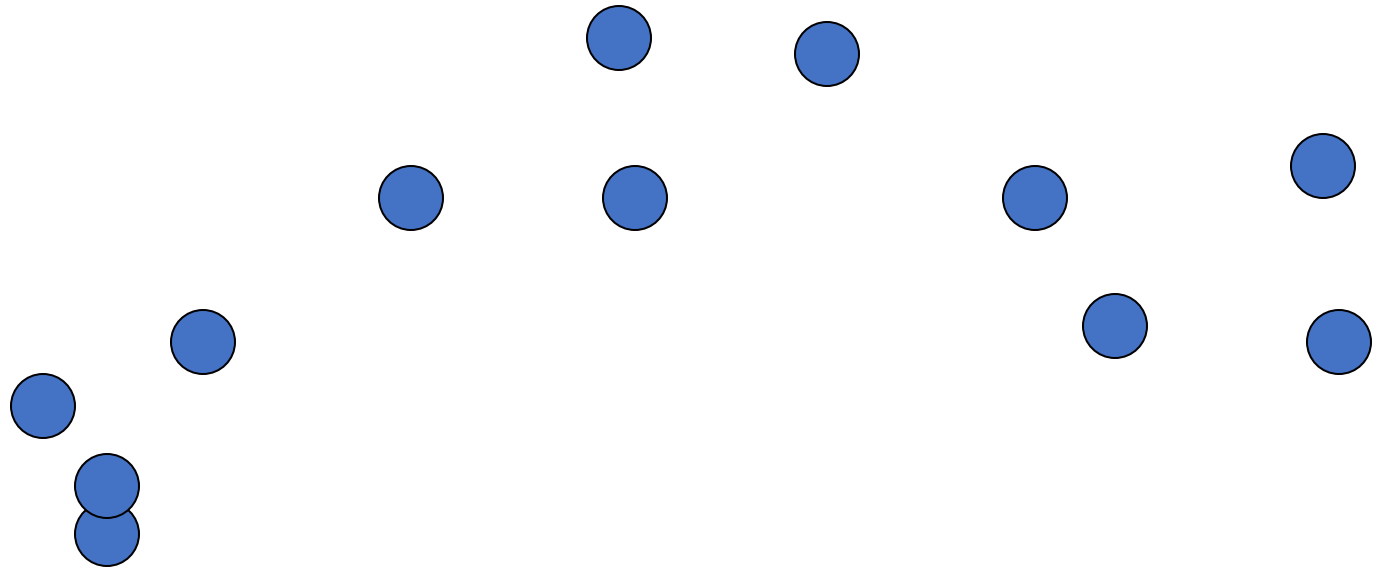
Slide credit: Josef Sivic

K-means

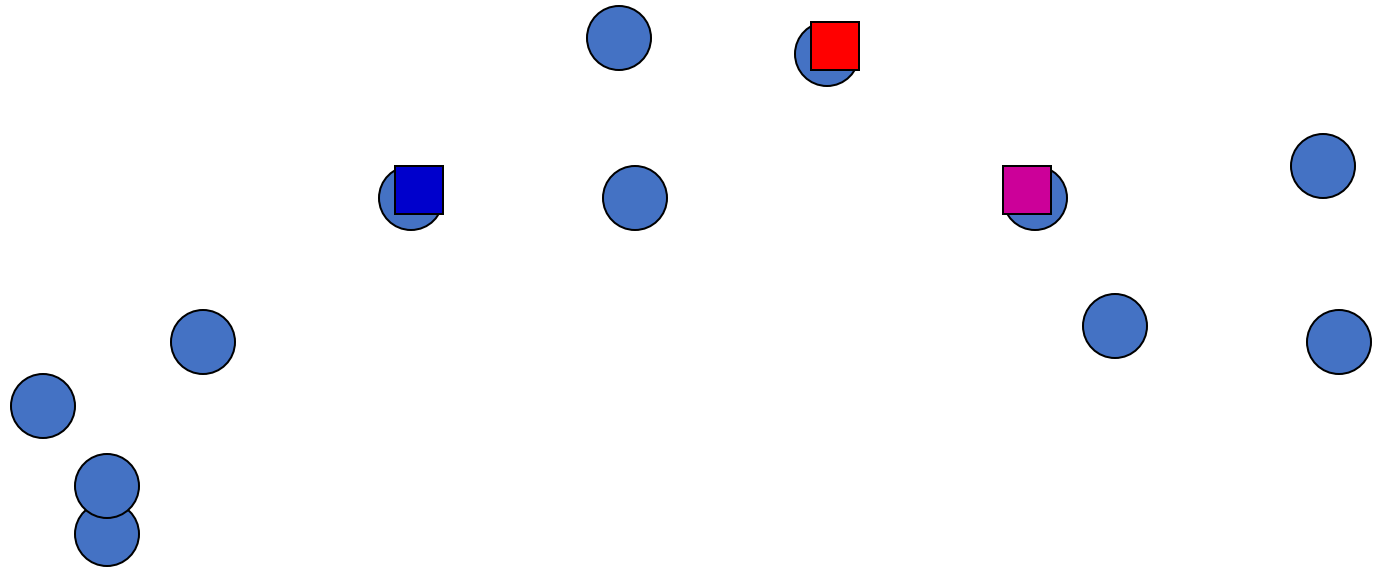
- Most well-known and popular clustering algorithm:
- Start with some initial cluster centers
- Iterate:
  - Assign/cluster each example to closest center
  - Recalculate centers as the mean of the points in a cluster

Credit: The slides for K-means are adapted from David Kauchak, CS 451 – Fall 2013

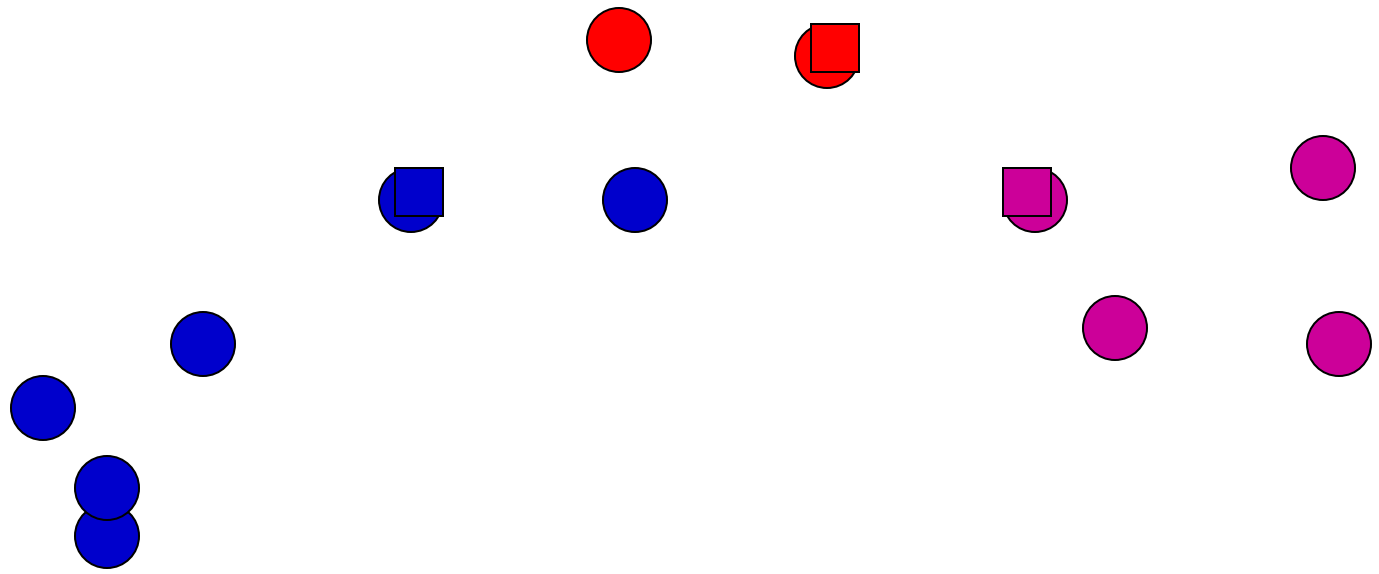
K-means: an example



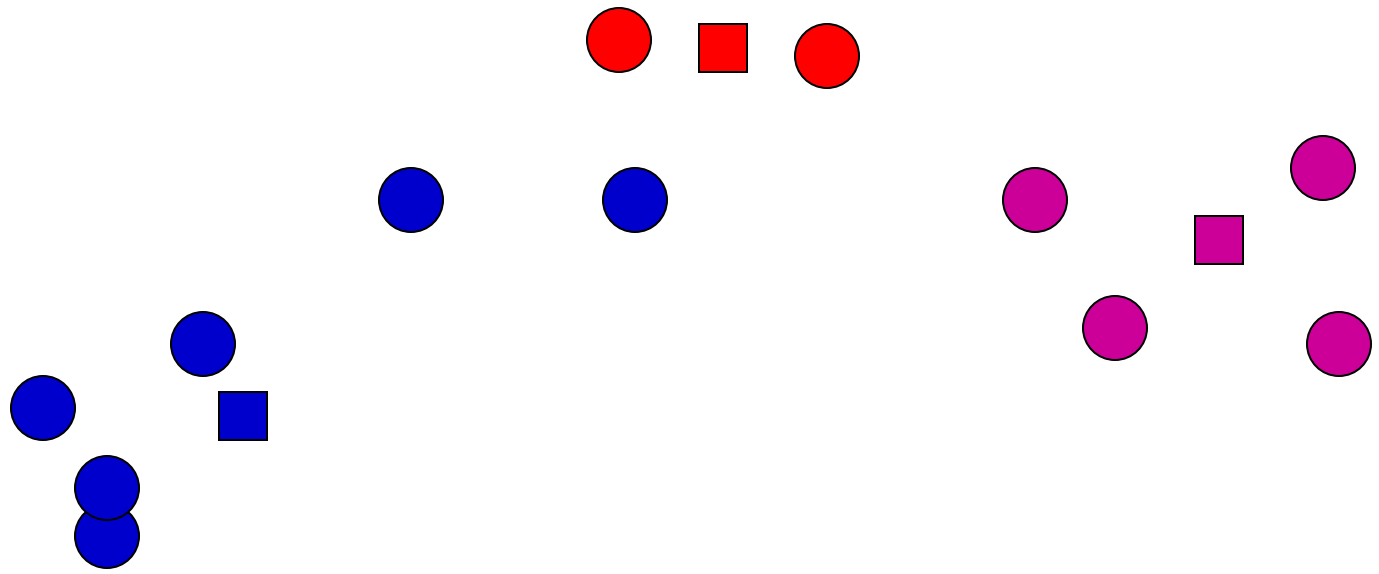
K-means: Initialize centers randomly



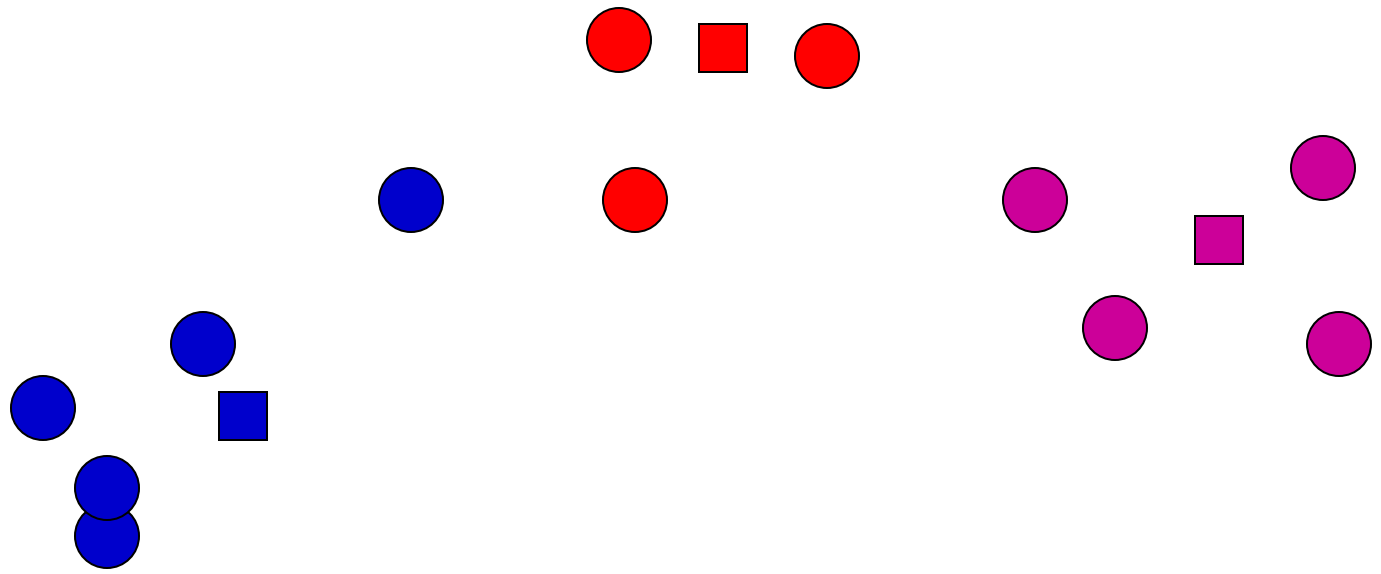
K-means: assign points to nearest center



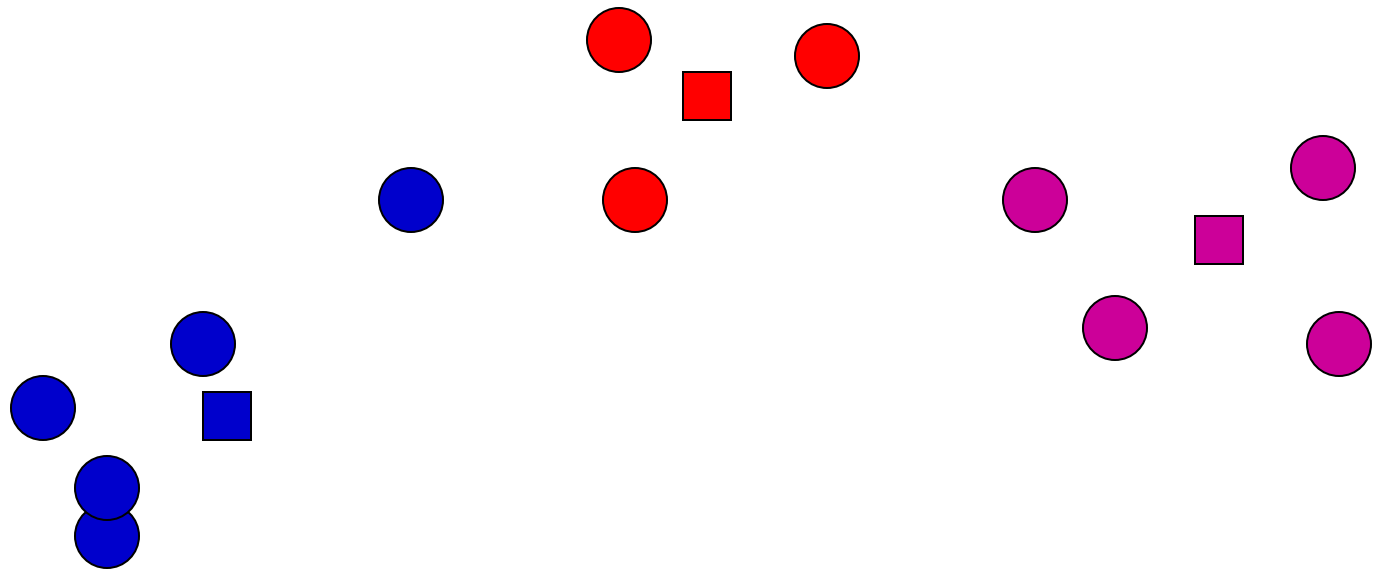
K-means: readjust centers



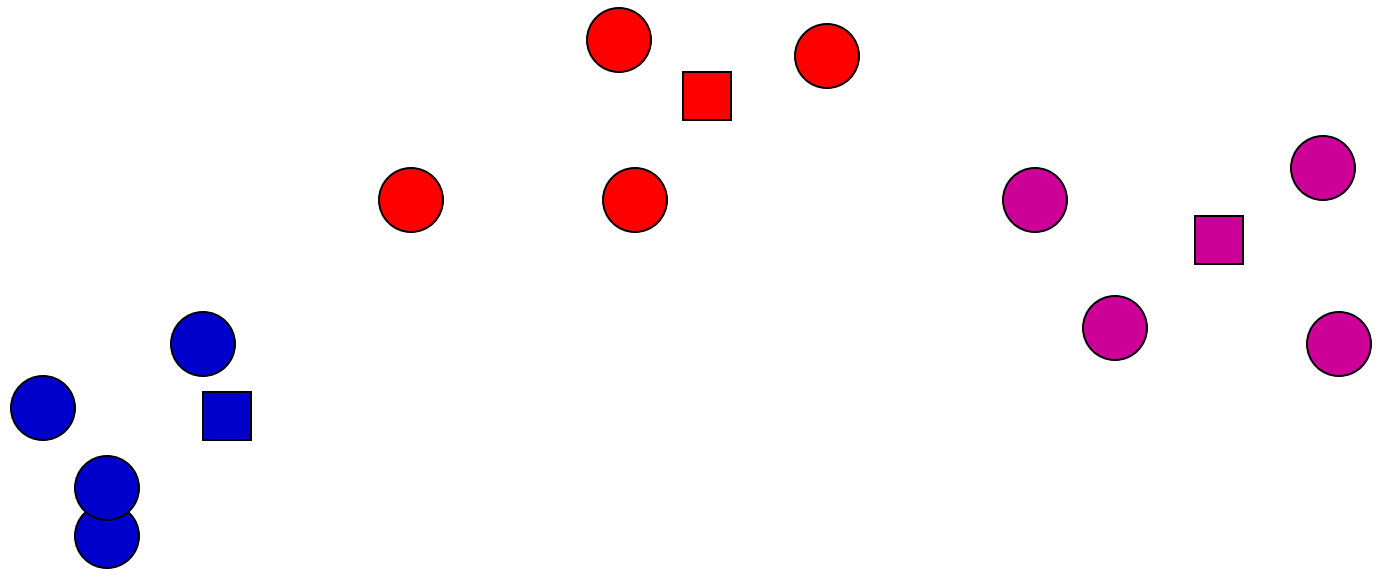
K-means: assign points to nearest center



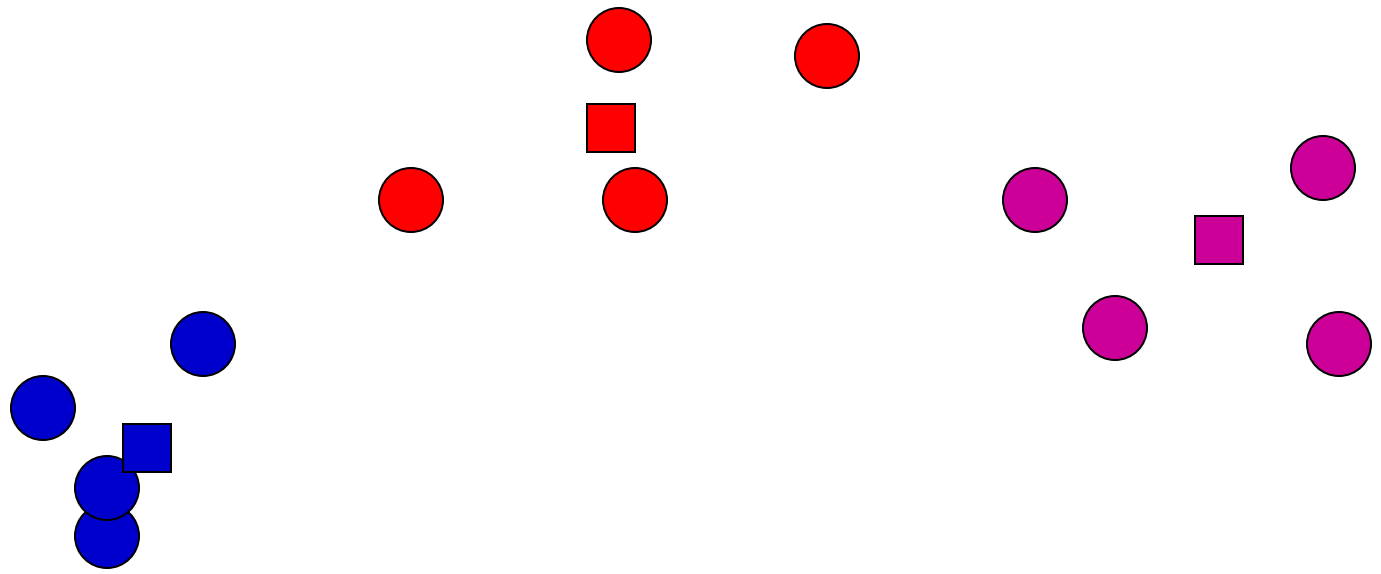
K-means: readjust centers



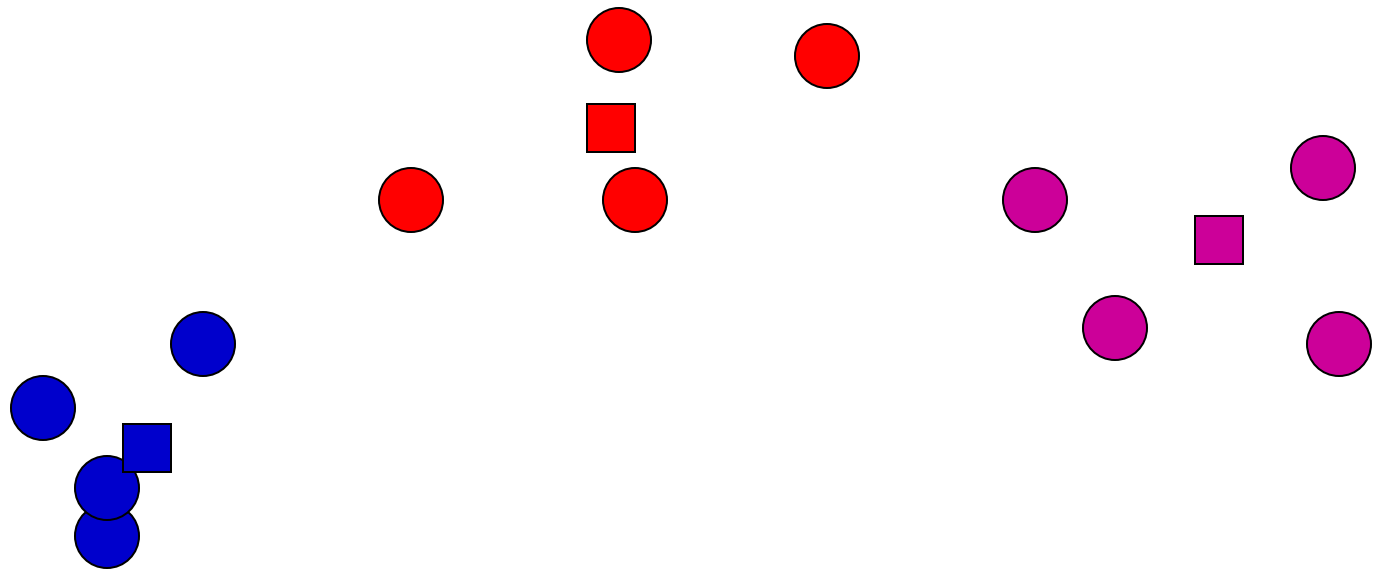
K-means: assign points to nearest center



K-means: readjust centers

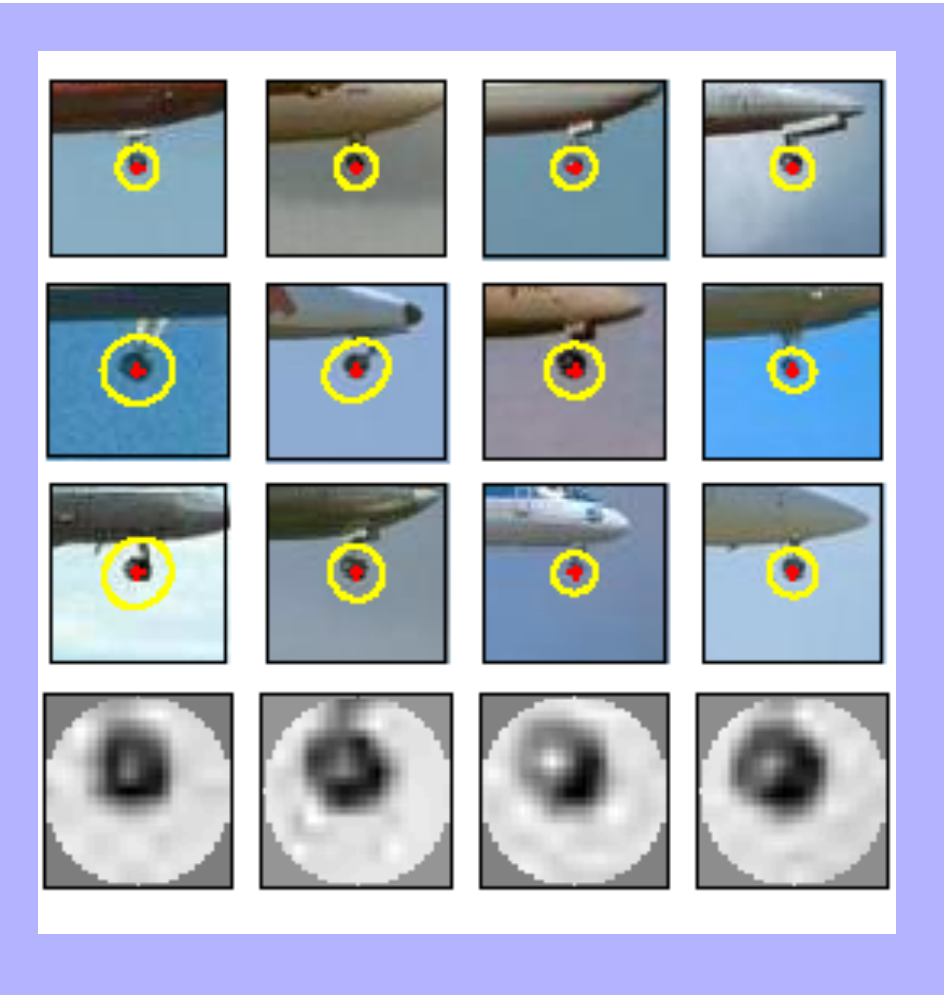


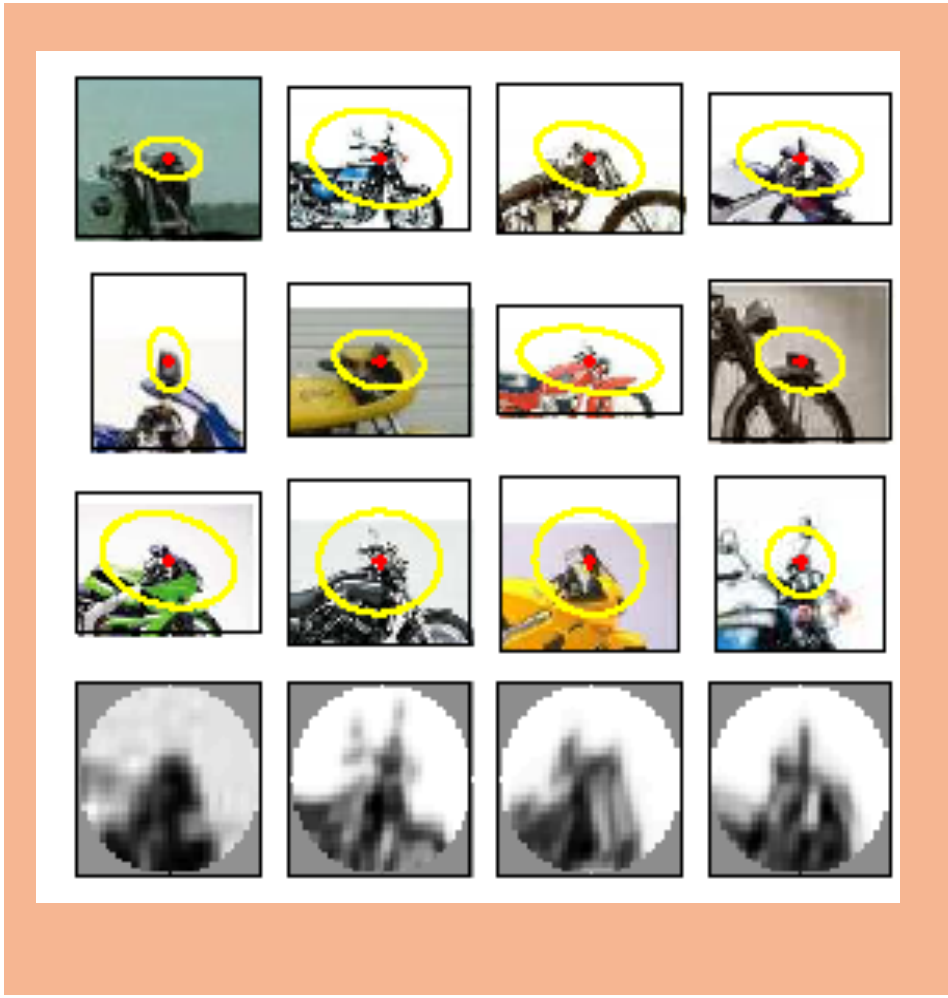
K-means: assign points to nearest center



No changes: Done

Image patch examples of codewords

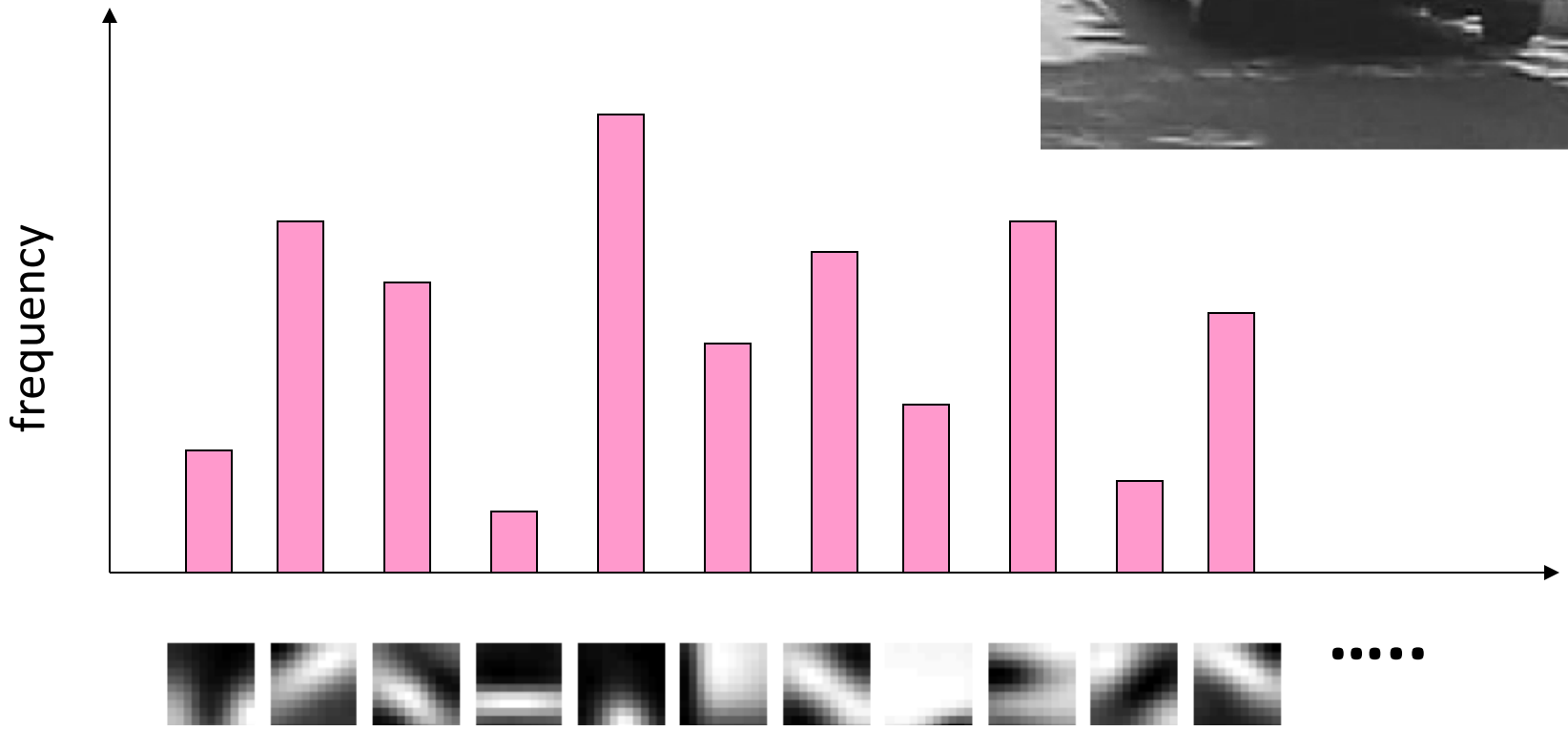




Sivic et al. 2005

Image representation

Histogram of features
assigned to each cluster



BoW as input to classifier

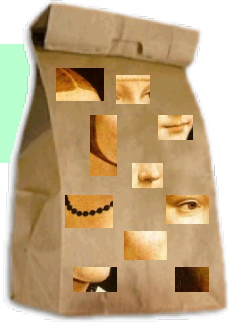
- Shared **dictionary** across images.
- SVM for object classification
  - Csurka, Bray, Dance & Fan, 2004

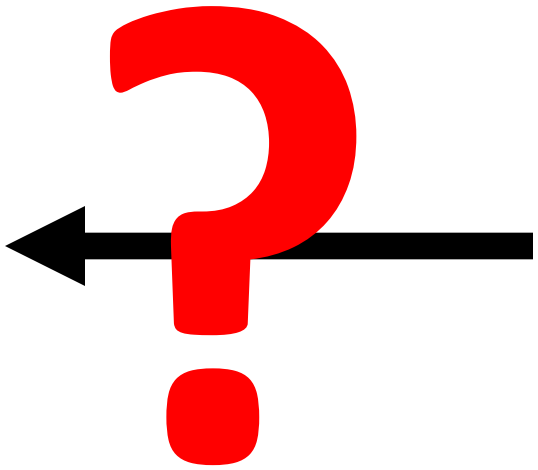
- Naïve Bayes
  - See 2007 edition of this course

More related work

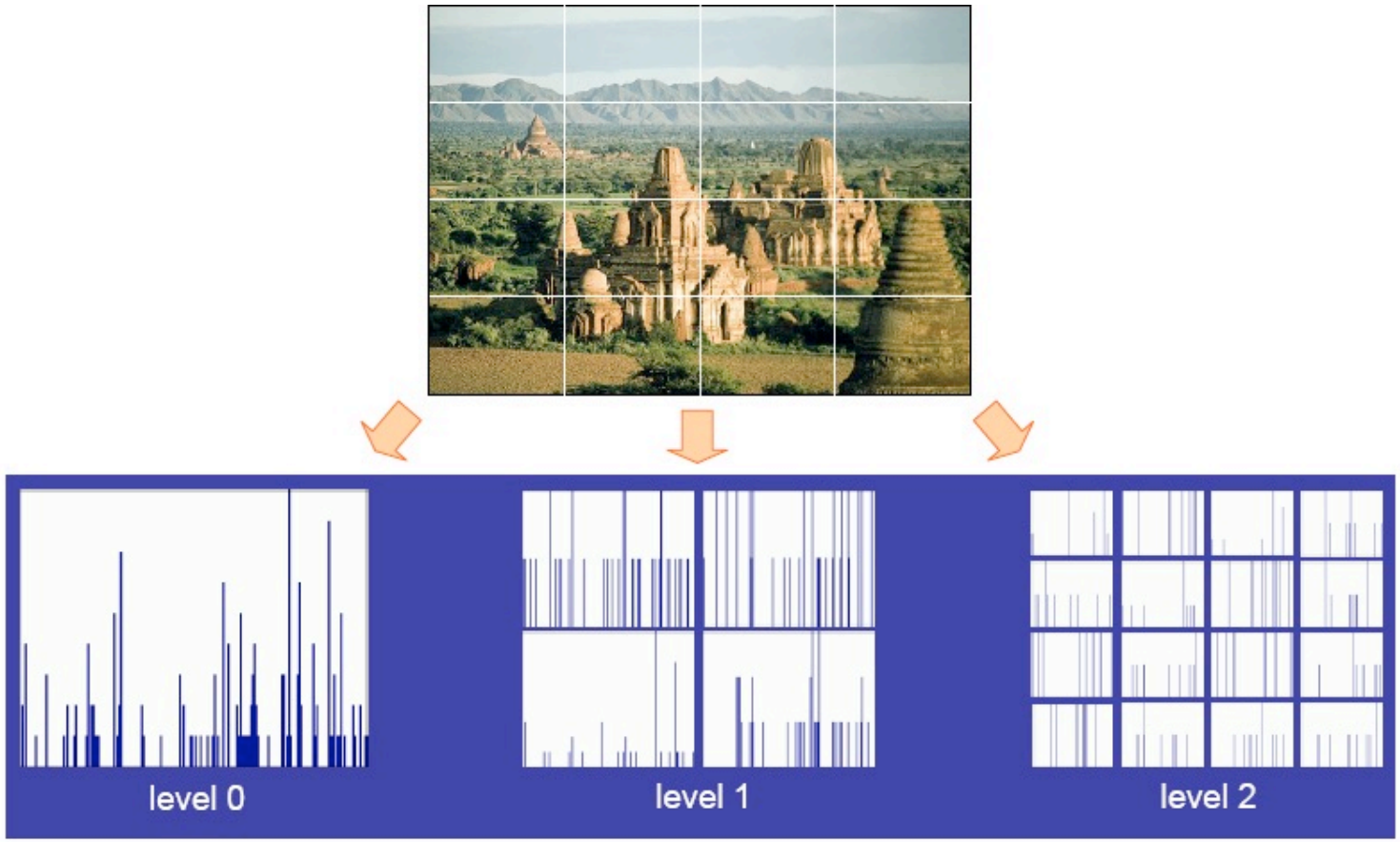
- Early “bag of words” models: mostly texture recognition
  - Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003
- Hierarchical Bayesian models for documents (pLSA, LDA, etc.)
  - Hoffman 1999; Blei, Ng & Jordan, 2004; Teh, Jordan, Beal & Blei, 2004
- Object categorization
  - Csurka, Bray, Dance & Fan, 2004; Sivic, Russell, Efros, Freeman & Zisserman, 2005; Sudderth, Torralba, Freeman & Willsky, 2005;
- Natural scene categorization
  - Vogel & Schiele, 2004; Fei-Fei & Perona, 2005; Bosch, Zisserman & Munoz, 2006

What about spatial info?

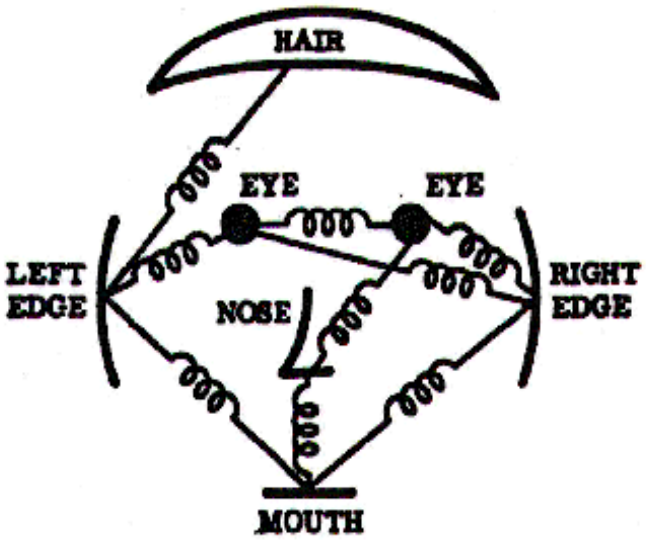




Spatial Pyramid Representation



Lazebnik, Schmid & Ponce, 2006



Part-based Models

Representation

- 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

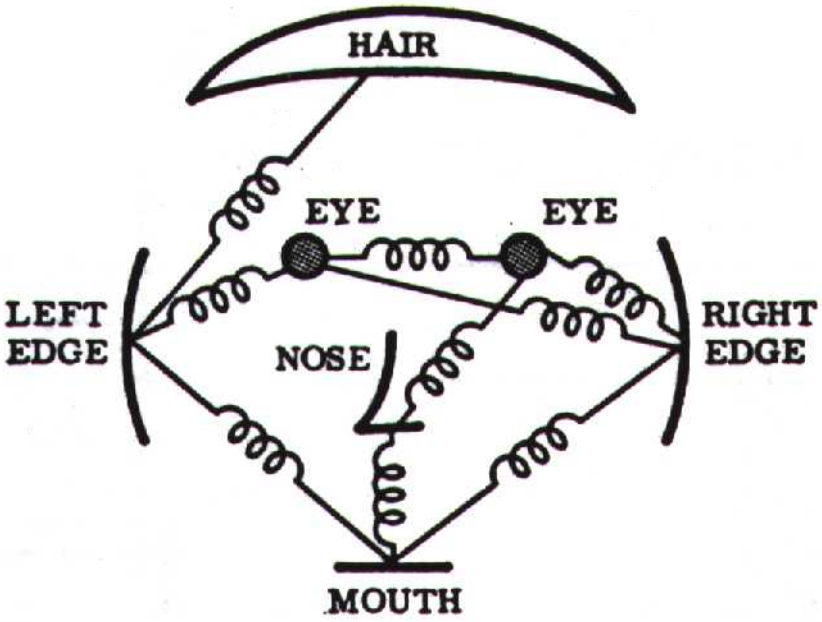


Figure from [Fischler & Elschlager 73]

Sparse representation

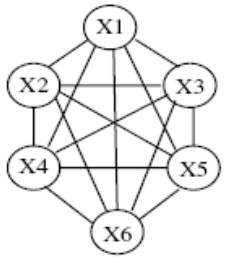
+ Computationally tractable (10^5 pixels → 10^1 -- 10^2 parts)
+ Generative representation of class
+ Avoid modeling global variability
+ Success in specific object recognition

- Throw away most image information
- Parts need to be distinctive to separate from other classes

Different connectivity structures

Fergus et al. ’03
Fei-Fei et al. ‘03

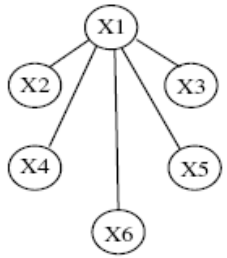
$O(N^6)$



a) Constellation [13]

Crandall et al. ‘05
Fergus et al. ’05

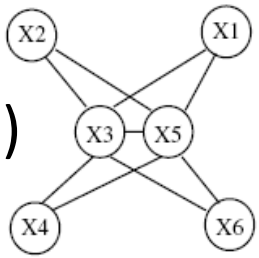
$O(N^2)$



b) Star shape [9, 14]

Crandall et al. ‘05

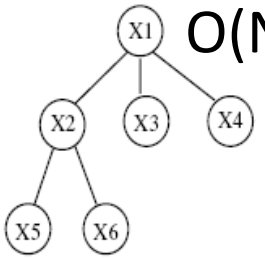
$O(N^3)$



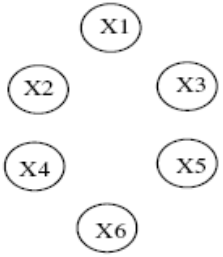
c) k-fan ($k = 2$) [9]

Felzenszwalb & Huttenlocher ‘00

$O(N^2)$

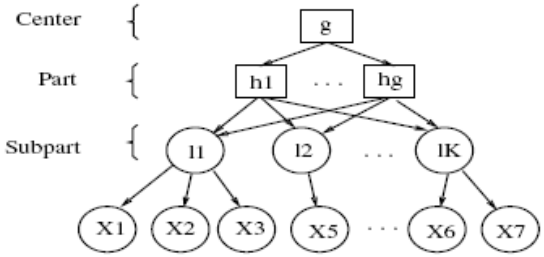


d) Tree [12]



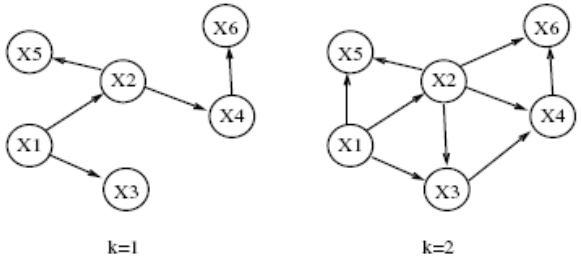
e) Bag of features [10, 21]

Csurka ’04
Vasconcelos ‘00



f) Hierarchy [4]

Bouchard & Triggs ‘05



g) Sparse flexible model

Carneiro & Lowe ‘06

from Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

Towards Object Recognition

- Develop an image representation
  - Bag of Words (BoW)
  - Part-based models

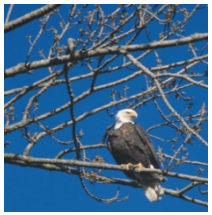
- Develop a classifier
  - K Nearest Neighbors
  - Metrics to measure distances

Classification Problems

- Given input: $\mathbf{x} = (x_1, x_2, \ldots, x_d)$
- Predict the output (class label) $y \in \mathcal{Y}$
  - Binary classification: $\mathcal{Y} = \{-1, +1\}$
  - Multi-class classification: $\mathcal{Y} = \{1, 2, \ldots, C\}$
- Learn a classification function: $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathcal{Y}$
- Regression: $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$

Classification Problem

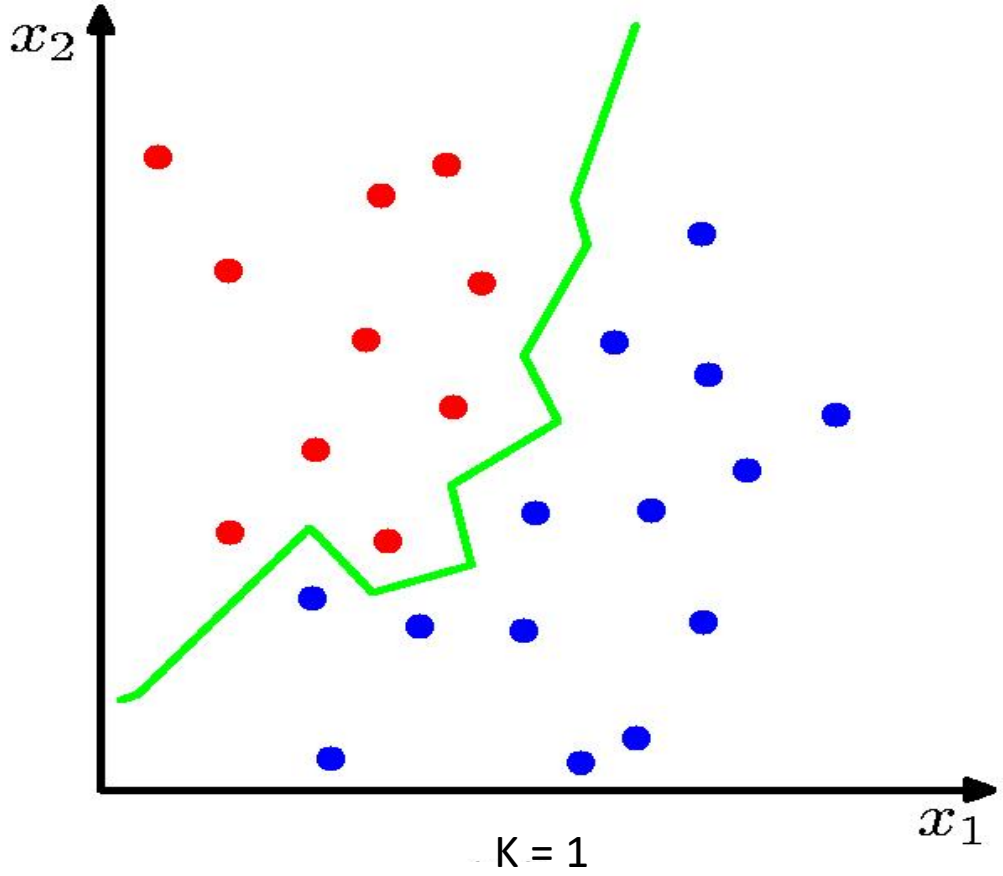
- Image Classification:


Which images are birds, which are not?

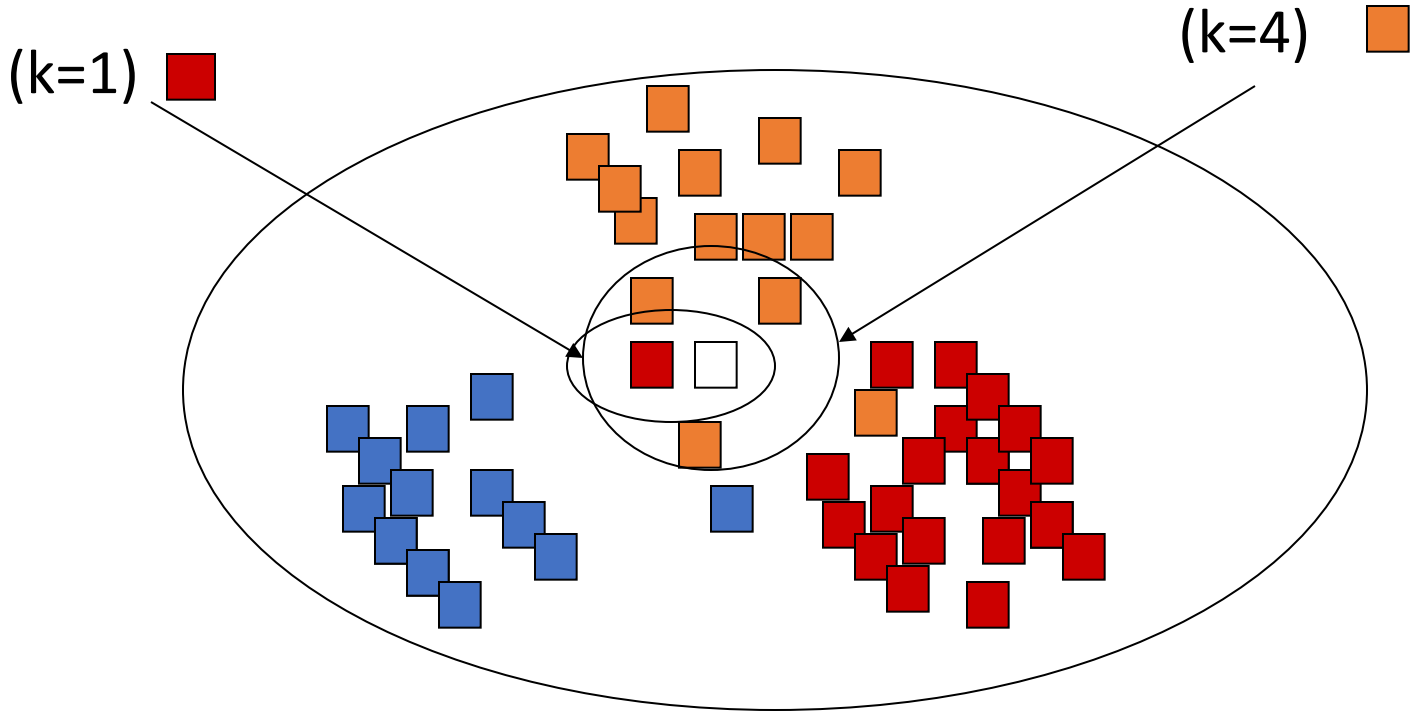
- Input features $\mathbf{x}$
- Class label y
  - 'bird image': $y = +1$
  - 'non-bird image': $y = -1$

K Nearest Neighbour (*k*NN)

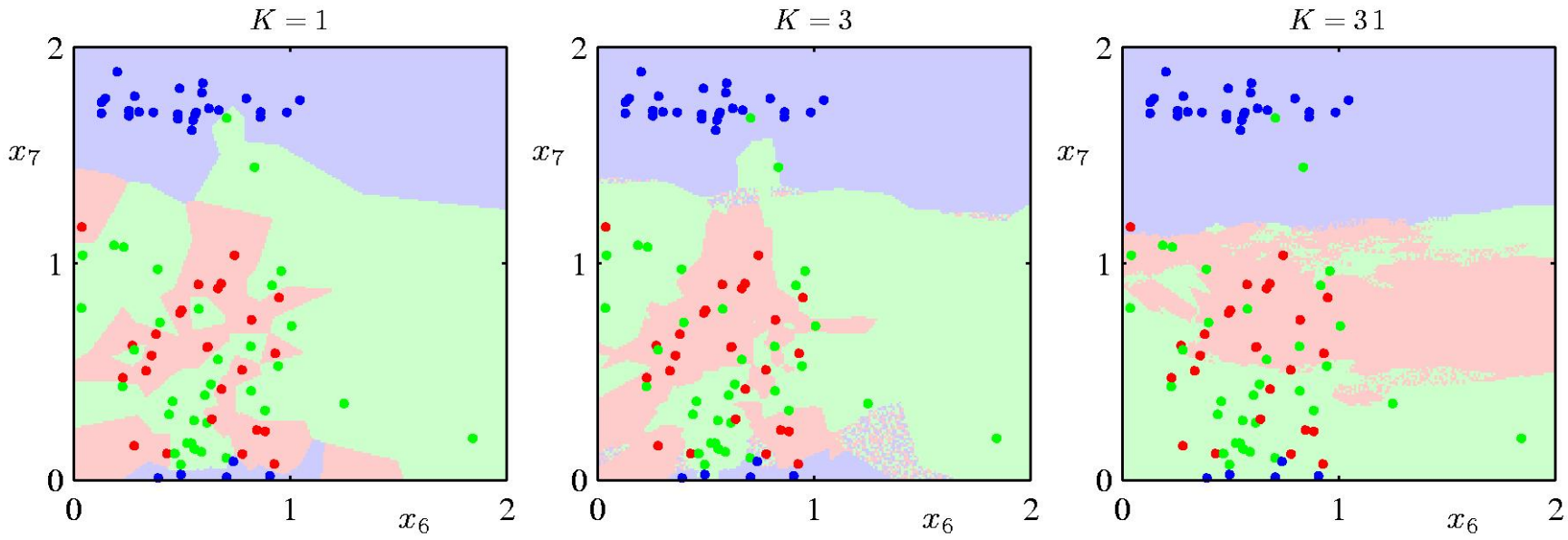


K Nearest Neighbour (*k*NN)

How many neighbors should we count ?



K Nearest Neighbour (*k*NN)

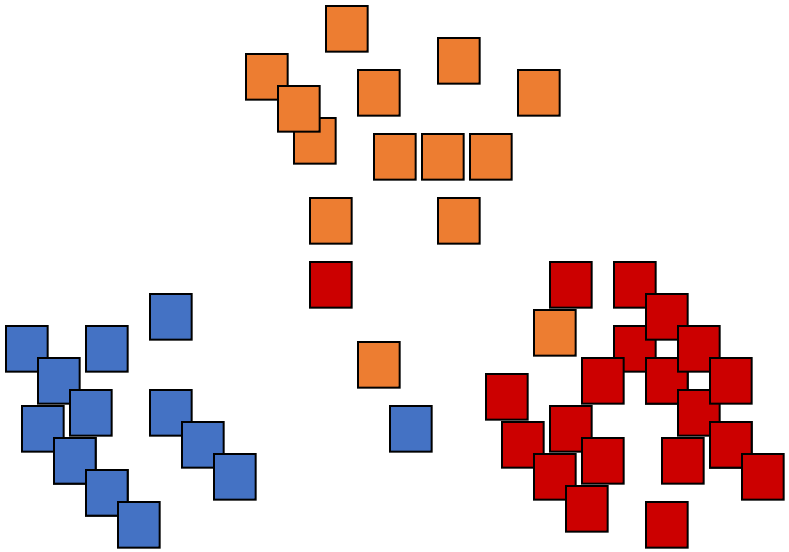


- K acts as a smother

Cross Validation

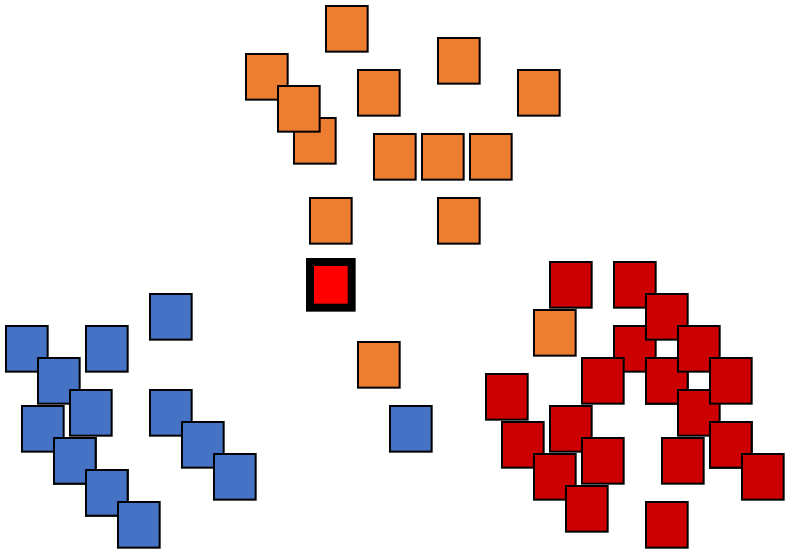
- Divide training examples into two sets
  - A training set (80%) and a validation set (20%)
- Predict the class labels for validation set by using the examples in training set
- Choose the number of neighbors *k* that maximizes the classification accuracy

Leave-One-Out Method



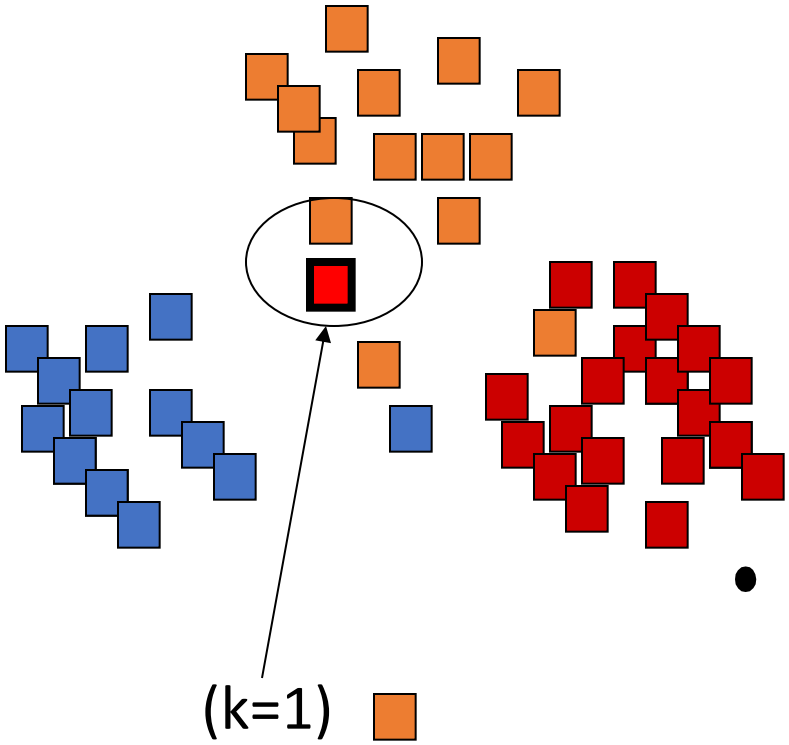
- For $k = 1, 2, \ldots, K$
  - $err(k) = 0$
  - For $i = 1, 2, \ldots, n$
    - Predict the class label $\widehat{y}_i$ for $\mathbf{x}_i$ using the remaining data points
    - $err(k) = err(k) + 1$ if $\widehat{y}_i \neq y_i$
- Output $k^* = \arg\min_{1 \leq k \leq K} err(k)$

Leave-One-Out Method



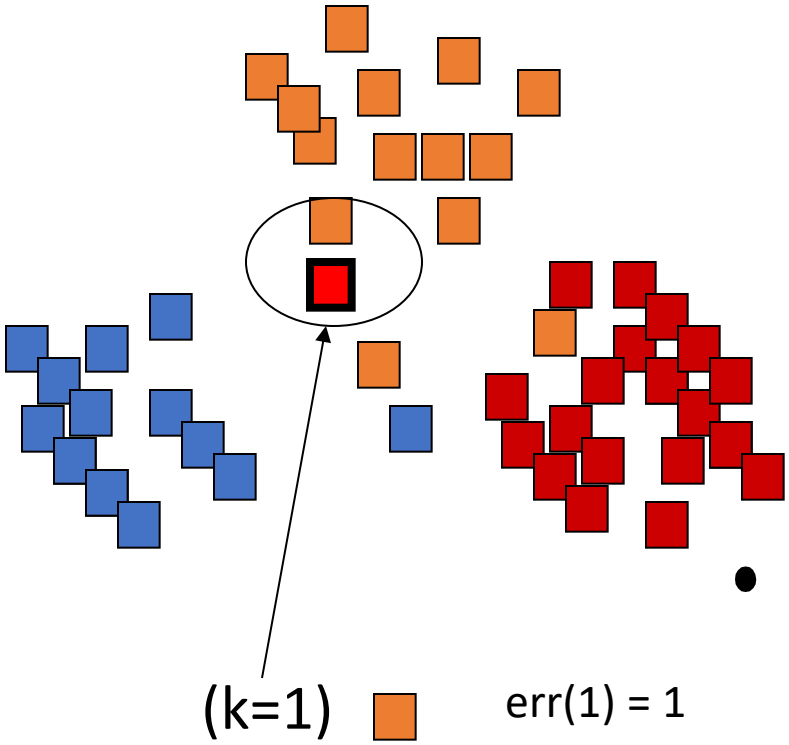
- For $k = 1, 2, \ldots, K$
  - $err(k) = 0$
  - For $i = 1, 2, \ldots, n$
    - Predict the class label $\widehat{y}_i$ for $\mathbf{x}_i$ using the remaining data points
    - $err(k) = err(k) + 1$ if $\widehat{y}_i \neq y_i$
- Output $k^* = \arg\min_{1 \leq k \leq K} err(k)$

Leave-One-Out Method



- For $k = 1, 2, \ldots, K$
  - $err(k) = 0$
  - For $i = 1, 2, \ldots, n$
    - Predict the class label $\widehat{y}_i$ for $\mathbf{x}_i$ using the remaining data points
    - $err(k) = err(k) + 1$ if $\widehat{y}_i \neq y_i$
- Output $k^* = \arg\min_{1 \leq k \leq K} err(k)$

Leave-One-Out Method



- For $k = 1, 2, \ldots, K$
  - $err(k) = 0$
  - For $i = 1, 2, \ldots, n$
    - Predict the class label $\widehat{y}_i$ for $\mathbf{x}_i$ using the remaining data points
    - $err(k) = err(k) + 1$ if $\widehat{y}_i \neq y_i$
- Output $k^* = \arg\min_{1 \leq k \leq K} err(k)$

Weighted kNN

- Weight the contribution of each close neighbor based on their distances
- Weight function

$$w(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\lambda |\mathbf{x} - \mathbf{x}_i|_2^2\right)$$

- Prediction

$$\Pr(y|\mathbf{x}) = \frac{\sum_{i=1}^{n} w(\mathbf{x}, \mathbf{x}_i)\delta(y, y_i)}{\sum_{i=1}^{n} w(\mathbf{x}, \mathbf{x}_i)}$$

$$\delta(y, y_i) = \begin{cases} 1 & y = y_i \\ 0 & y \neq y_i \end{cases}$$

When to Consider Nearest Neighbor ?

- Lots of training data
- Less than 20 attributes per example
- Advantages:
  - Training is very fast
  - Learn complex target functions
  - Don’t lose information
- Disadvantages:
  - Slow at query time
  - Easily fooled by irrelevant attributes

Metrics between sample points

- Plays a foundational role in classification

- Many different metrics
  - Euclidean
  - Minkowski (L1, L2, Linf)
  - Distances between histograms
  - Cosine

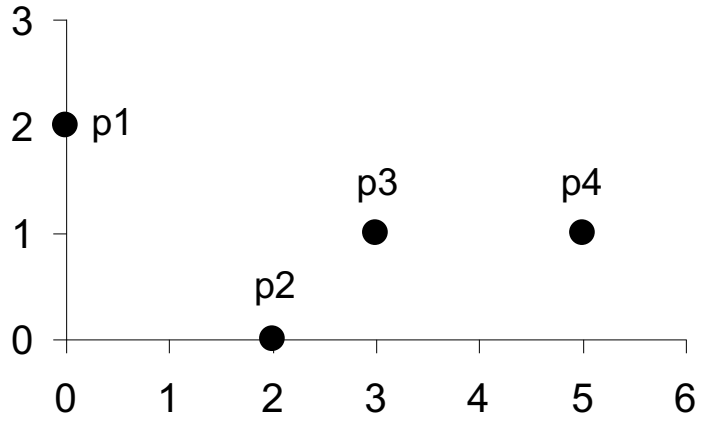
Similarity and Dissimilarity

- Similarity
  - Numerical measure of how alike two data objects are.
  - Is higher when objects are more alike.
  - Often falls in the range [0,1]
- Dissimilarity
  - Numerical measure of how different are two data objects
  - Lower when objects are more alike
  - Minimum dissimilarity is often 0
  - Upper limit varies
- Proximity refers to a similarity or dissimilarity

Euclidean Distance

$$dist = \sqrt{\sum_{k=1}^{n} (p_k - q_k)^2}$$

Euclidean Distance



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Minkowski Distance

$$dist = \left(\sum_{k=1}^{n} | p_k - q_k |^r \right)^{\frac{1}{r}}$$

Minkowski Distance: Examples

- $r = 1$. City block (Manhattan, taxicab, L_1 norm) distance.
  - A common example of this is the Hamming distance, which is just the number of bits that are different between two binary vectors

- $r = 2$. Euclidean distance

- $r \rightarrow \infty$. "supremum" (L_{max} norm, L_∞ norm) distance.
  - This is the maximum difference between any component of the vectors
  - Example: L_infinity of (1, 0, 2) and (6, 0, 3) = ??

  - Do not confuse *r* with *n*, i.e., all these distances are defined for all numbers of dimensions.

Minkowski Distance

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Cosine Similarity

- If d_1 and d_2 are two document vectors, then

$$\cos(d_1, d_2) = (d_1 \bullet d_2) / ||d_1|| \, ||d_2|| ,$$

where • indicates vector dot product and || d || is the length of vector d.

- Example:

d_1 = **3 2 0 5 0 0 0 2 0 0**

d_2 = **1 0 0 0 0 0 0 1 0 2**

$d_1 \bullet d_2$= 3*1 + 2*0 + 0*0 + 5*0 + 0*0 + 0*0 + 0*0 + 2*1 + 0*0 + 0*2 = 5

$||d_1||$ = $(3*3+2*2+0*0+5*5+0*0+0*0+0*0+2*2+0*0+0*0)^{0.5}$ = $(42)^{0.5}$ = 6.481

$||d_2||$ = $(1*1+0*0+0*0+0*0+0*0+0*0+0*0+1*1+0*0+2*2)^{0.5}$ = $(6)^{0.5}$ = 2.245

$\cos(d_1, d_2)$ = .3150, distance=1-cos(d1,d2)

Similarity Between Binary Vectors

- Common situation is that objects, *p* and *q*, have only binary attributes
- Compute similarities using the following quantities

  M_{01} = the number of attributes where p was 0 and q was 1

  M_{10} = the number of attributes where p was 1 and q was 0

  M_{00} = the number of attributes where p was 0 and q was 0

  M_{11} = the number of attributes where p was 1 and q was 1

- **Simple Matching** and **Jaccard Coefficients**

  SMC = number of matches / number of attributes

  = $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

  J = number of value-1-to-value-1 matches / number of not-both-zero attributes values

  = $(M_{11}) / (M_{01} + M_{10} + M_{11})$

SMC versus Jaccard: Example

p = 1 0 0 0 0 0 0 0 0 0
q = 0 0 0 0 0 0 1 0 0 1

M_{01} = 2 (the number of attributes where p was 0 and q was 1)
M_{10} = 1 (the number of attributes where p was 1 and q was 0)
M_{00} = 7 (the number of attributes where p was 0 and q was 0)
M_{11} = 0 (the number of attributes where p was 1 and q was 1)

SMC = $(M_{11} + M_{00})/(M_{01} + M_{10} + M_{11} + M_{00})$ = (0+7) / (2+1+0+7) = 0.7

J = (M_{11}) / $(M_{01} + M_{10} + M_{11})$ = 0 / (2 + 1 + 0) = 0

Summary

- Object recognition pipeline
- Image representation
  - Bag of Words
  - Part-based Model
- Classification
  - K-Nearest Neighbors
  - Metrics
  - There are much more (to be covered next)