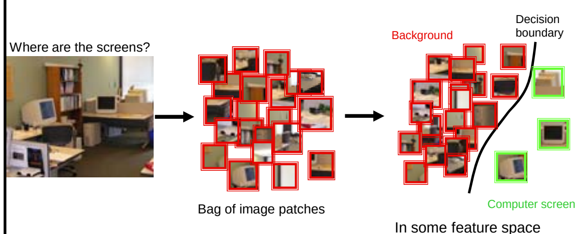
Lecture 10
Discriminative models

Overview of section

- Object detection with classifiers
- Boosting
  - Gentle boosting
  - Weak detectors
  - Object model
  - Object detection
- Nearest-Neighbor methods
- Multiclass object detection
- Context

Discriminative methods

Object detection and recognition is formulated as a classification problem.
The image is partitioned into a set of overlapping windows
… and a decision is taken at each window about if it contains a target object or not.

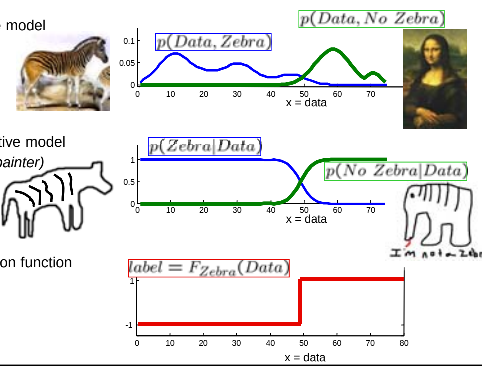


Discriminative vs. generative

- Generative model
  (*The artist*)

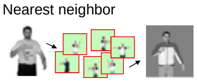
- Discriminative model
  (*The lousy painter*)

- Classification function



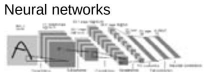
Discriminative methods

Nearest neighbor



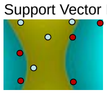
Shakhnarovich, Viola, Darrell 2003
Berg, Berg, Malik 2005
…

Neural networks



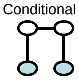
LeCun, Bottou, Bengio, Haffner 1998
Rowley, Baluja, Kanade 1998
…

Support Vector Machines and Kernels



Guyon, Vapnik
Heisele, Serre, Poggio, 2001
…

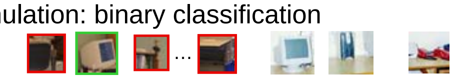
Conditional Random Fields



McCallum, Freitag, Pereira 2000
Kumar, Hebert 2003
…

Formulation

- Formulation: binary classification



Features	x =	x_1	x_2	x_3	…	x_N	x_{N+1}	x_{N+2}	…	x_{N+M}
Labels	y =	-1	+1	-1		-1	?	?		?

Training data: each image patch is labeled as containing the object or background

Test data

- Classification function

$$\hat{y} = F(x)$$ Where $F(x)$ belongs to some family of functions

- Minimize misclassification error

(Not that simple: we need some guarantees that there will be generalization)

Overview of section

- Object detection with classifiers
- **Boosting**
  - Gentle boosting
  - Weak detectors
  - Object model
  - Object detection
- Nearest-Neighbor methods
- Multiclass object detection
- Context

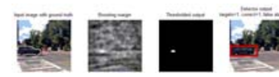
A simple object detector with Boosting

http://people.csail.mit.edu/torralba/iccv2005/

Download
- Toolbox for manipulating dataset
- Code and dataset

Matlab code
- Gentle boosting
- Object detector using a part based model

Dataset with cars and computer monitors



Why boosting?

- A simple algorithm for learning robust classifiers
  - Freund & Shapire, 1995
  - Friedman, Hastie, Tibshhirani, 1998
- Provides efficient algorithm for sparse visual feature selection
  - *Tieu & Viola, 2000*
  - *Viola & Jones, 2003*
- Easy to implement, not requires external optimization tools.

Boosting

- Defines a classifier using an additive model:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier

Features vector

Weight

Weak classifier

Boosting

- Defines a classifier using an additive model:

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Strong classifier

Features vector

Weight

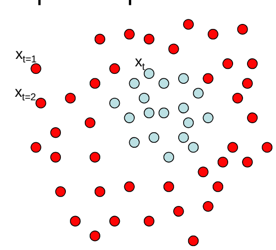
Weak classifier

- We need to define a family of weak classifiers

$f_k(x)$ from a family of weak classifiers

Boosting

- It is a sequential procedure:



Each data point has a class label:

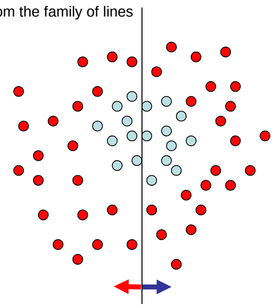
$y_t = \begin{cases} +1 \\ -1 \end{cases}$

and a weight:

$w_t = 1$

Toy example

Weak learners from the family of lines



Each data point has a class label:

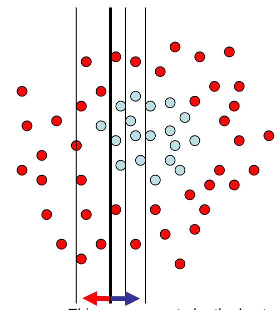
$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

and a weight:

$w_t = 1$

h => p(error) = 0.5 it is at chance

Toy example



Each data point has a class label:

$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

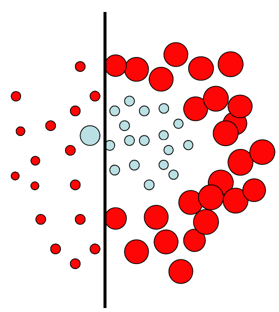
and a weight:

$w_t = 1$

This one seems to be the best

This is a '**weak classifier**': It performs slightly better than chance.

Toy example



Each data point has a class label:

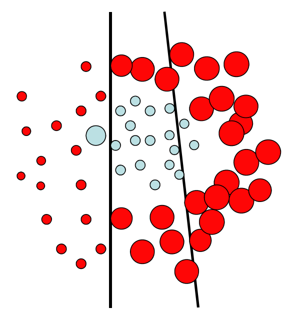
$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

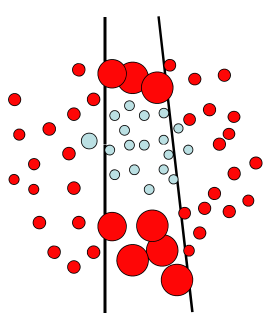
$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

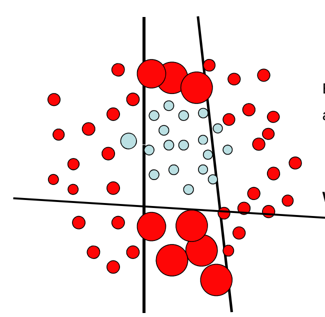
$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



Each data point has a class label:

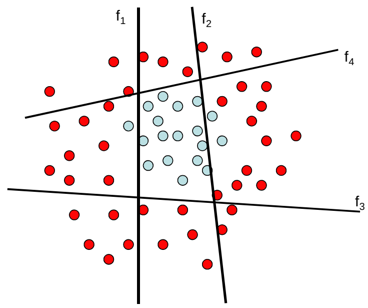
$y_t = \begin{cases} +1 & (\bullet) \\ -1 & (\circ) \end{cases}$

We update the weights:

$w_t \leftarrow w_t \exp\{-y_t H_t\}$

We set a new problem for which the previous weak classifier performs at chance again

Toy example



The strong (non- linear) classifier is built as the combination of all the weak (linear) classifiers.

Boosting

- Different cost functions and minimization algorithms result is various flavors of Boosting
- In this demo, I will use gentleBoosting: it is simple to implement and numerically stable.

Overview of section

- Boosting
  - **Gentle boosting**
  - Weak detectors
  - Object model
  - Object detection

Boosting

Boosting fits the additive model

$$F(x) = f_1(x) + f_2(x) + f_3(x) + \dots$$

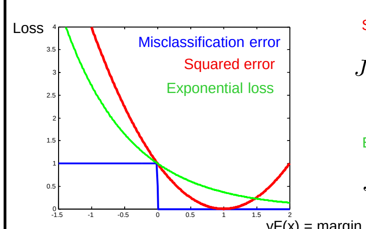
by minimizing the exponential loss

$$J(F) = \sum_{t=1}^{N} e^{-y_t F(x_t)}$$

Training samples

The exponential loss is a differentiable upper bound to the misclassification error.

Exponential loss



Squared error

$$J = \sum_{t=1}^{N} [y_t - F(x_t)]^2$$

Exponential loss

$$J = \sum_{t=1}^{N} e^{-y_t F(x_t)}$$

Boosting

Sequential procedure. At each step we add

$$F(x) \leftarrow F(x) + f_m(x)$$

to minimize the residual loss

$$(\phi_m) = \arg\min_{\phi} \sum_{t=1}^{N} J\left(y_i, F(x_t) + f(x_t; \phi)\right)$$

Parameters weak classifier **Desired output** **input**

For more details: Friedman, Hastie, Tibshirani. "Additive Logistic Regression: a Statistical View of Boosting" (1998)

gentleBoosting

- At each iteration:

We chose $f_m(x)$ that minimizes the cost:

$$J(F + f_m) = \sum_{t=1}^{N} e^{-y_t(F(x_t)+f_m(x_t))}$$

Instead of doing exact optimization, gentle Boosting minimizes a Taylor approximation of the error:

$$J(F) \propto \sum_{t=1}^{N} \boxed{e^{-y_t F(x_t)}} (y_t - f_m(x_t))^2$$

Weights at this iteration

At each iterations we just need to solve a weighted least squares problem

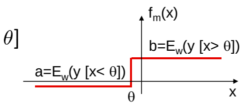
For more details: Friedman, Hastie, Tibshirani. "Additive Logistic Regression: a Statistical View of Boosting" (1998)

Weak classifiers

- The input is a set of weighted training samples (x,y,w)
- Regression stumps: simple but commonly used in object detection.

$$f_m(x) = a[x_k < \theta] + b[x_k \geq \theta]$$

Four parameters: $[a, b, \theta, k]$



fitRegressionStump.m

gentleBoosting.m

```
function classifier = gentleBoost(x, y, Nrounds)
…
for m = 1:Nrounds
   fm = selectBestWeakClassifier(x, y, w);
   w = w .* exp(- y .* fm);
   % store parameters of fm in classifier
   …
end
```

Initialize weights w = 1

Solve weighted least-squares

Re-weight training samples

Demo gentleBoosting

Demo using Gentle boost and stumps with hand selected 2D data:

> demoGentleBoost.m

Flavors of boosting

- AdaBoost (Freund and Shapire, 1995)
- Real AdaBoost (Friedman et al, 1998)
- LogitBoost (Friedman et al, 1998)
- Gentle AdaBoost (Friedman et al, 1998)
- BrownBoosting (Freund, 2000)
- FloatBoost (Li et al, 2002)
- …

Overview of section

- Boosting
  – Gentle boosting
  – **Weak detectors**
  – Object model
  – Object detection

From images to features: Weak detectors

We will now define a family of visual features that can be used as weak classifiers ("weak detectors")

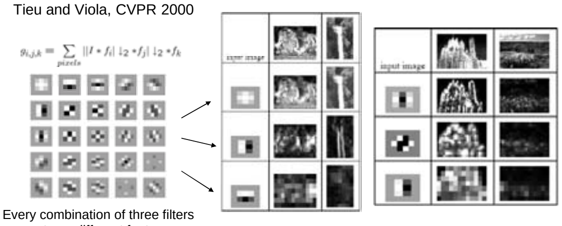

$\longrightarrow h_i(I, x, y) \longrightarrow$


Takes image as input and the output is binary response.
The output is a weak detector.

Weak detectors

Textures of textures

Tieu and Viola, CVPR 2000



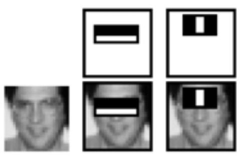
Every combination of three filters generates a different feature

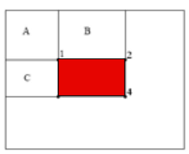
This gives thousands of features. Boosting selects a sparse subset, so computations on test time are very efficient. Boosting also avoids overfitting to some extend.

Weak detectors

Haar filters and integral image

Viola and Jones, ICCV 2001

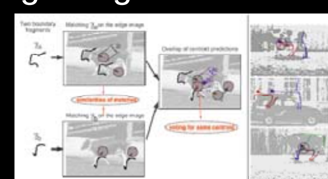




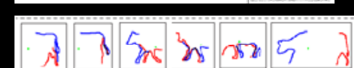
The average intensity in the block is computed with four sums independently of the block size.

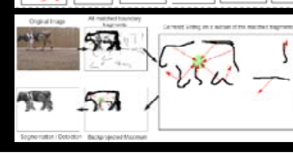
Edge fragments

Opelt, Pinz, Zisserman, ECCV 2006



Weak detector = k edge fragments and threshold. Chamfer distance uses 8 orientation planes





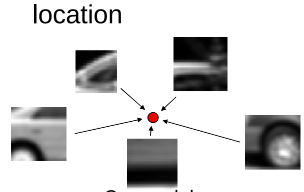
Weak detectors

Other weak detectors:

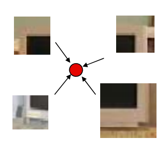
- Carmichael, Hebert 2004
- Yuille, Snow, Nitzbert, 1998
- Amit, Geman 1998
- Papageorgiou, Poggio, 2000
- Heisele, Serre, Poggio, 2001
- Agarwal, Awan, Roth, 2004
- Schneiderman, Kanade 2004
- …

Weak detectors

Part based: similar to part-based generative models. We create weak detectors by using parts and voting for the object center location



Car model



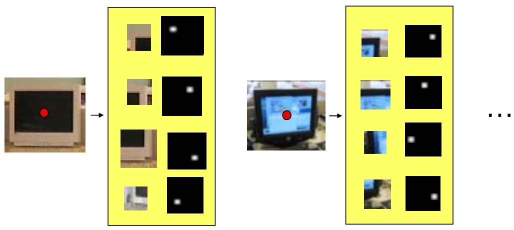
Screen model

These features are used for the detector on the course web site.

Weak detectors

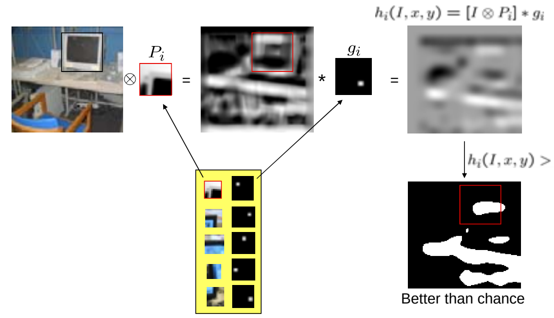
First we collect a set of part templates from a set of training objects.

Vidal-Naquet, Ullman (2003)



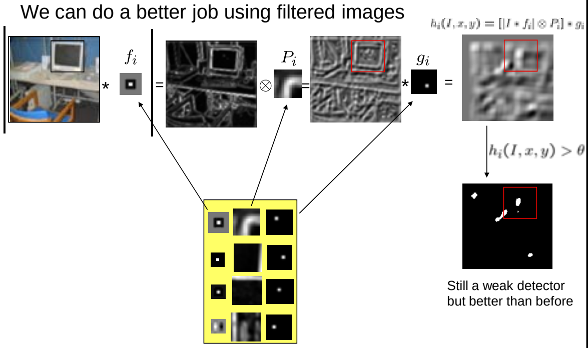
Weak detectors

We now define a family of "weak detectors" as:



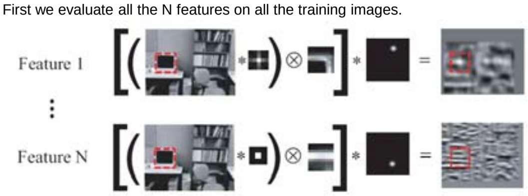
Weak detectors

We can do a better job using filtered images

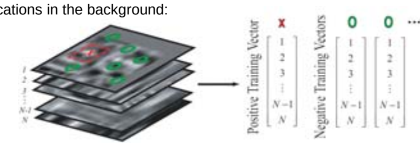


Training

First we evaluate all the N features on all the training images.

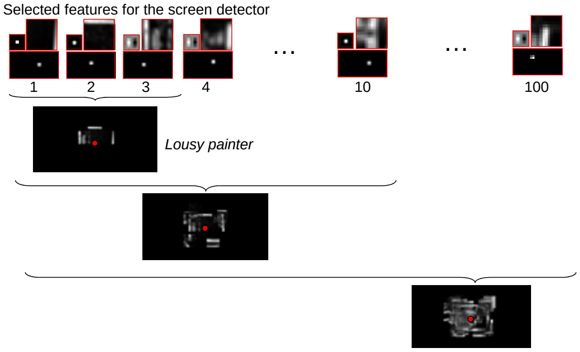


Then, we sample the feature outputs on the object center and at random locations in the background:



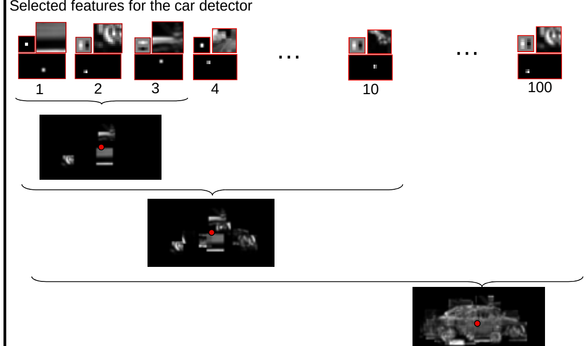
Representation and object model

Selected features for the screen detector



Representation and object model

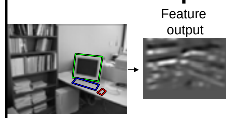
Selected features for the car detector



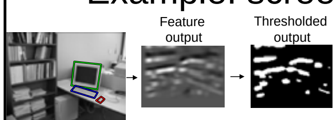
Overview of section

- Boosting
  - Gentle boosting
  - Weak detectors
  - Object model
  - **Object detection**

Example: screen detection

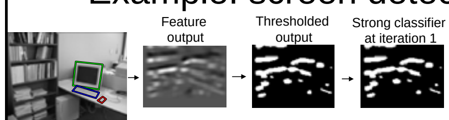


Example: screen detection

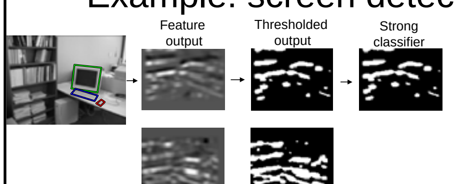


Weak 'detector'
Produces many false alarms.

Example: screen detection

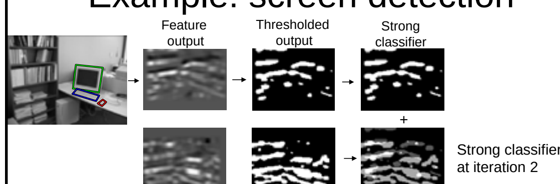


Example: screen detection

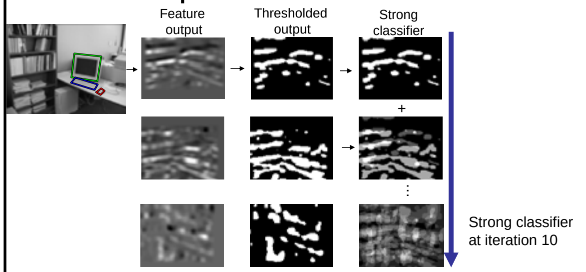


Second weak 'detector'
Produces a different set of false alarms.

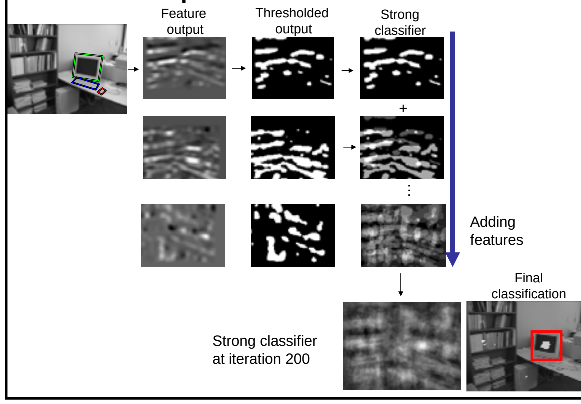
Example: screen detection



Example: screen detection



Example: screen detection



Demo

Demo of screen and car detectors using parts, Gentle boost, and stumps:

> runDetector.m

Probabilistic interpretation

- Generative model

$p(features,\ object\ class)$

- Discriminative (Boosting) model.

Boosting is fitting an additive logistic regression model:

$$p(object\ class \mid features) = \frac{1}{1 + e^{-\sum h_i(I,x,y)}}$$

It can be a set of arbitrary functions of the image

This provides a great flexibility, difficult to beat by current generative models. But also there is the danger of not understanding what are they really doing.

Weak detectors

- Generative model

$p(features,\ object\ class)$

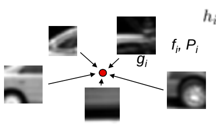
- Discriminative (Boosting) model.

Boosting is fitting an additive logistic regression model:

$$p(object\ class \mid features) = \frac{1}{1 + e^{-\sum h_i(I,x,y)}}$$

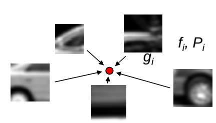
$$h_i(I,x,y) = [(I * f_i) \otimes P_i] * g_i$$

Image
Feature
Part template
Relative position wrt object center



Object models

- Invariance: search strategy
- Part based

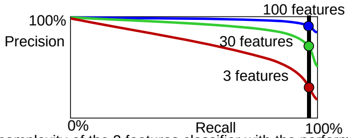


Here, invariance in translation and scale is achieved by the search strategy: the classifier is evaluated at all locations (by translating the image) and at all scales (by scaling the image in small steps).

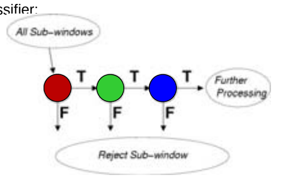
The search cost can be reduced using a cascade.

Cascade of classifiers

Fleuret and Geman 2001, Viola and Jones 2001



We want the complexity of the 3 features classifier with the performance of the 100 features classifier:



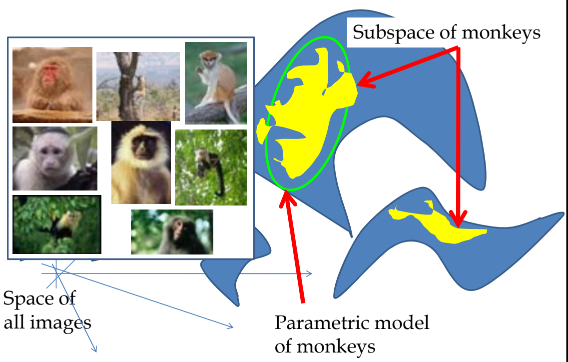
Select a threshold with high recall for each stage.

We increase precision using the cascade

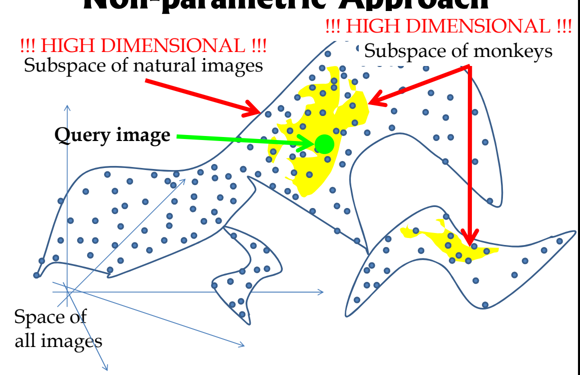
Overview of section

- Object detection with classifiers
- Boosting
  - Gentle boosting
  - Weak detectors
  - Object model
  - Object detection
- Nearest-Neighbor methods
- Multiclass object detection
- Context

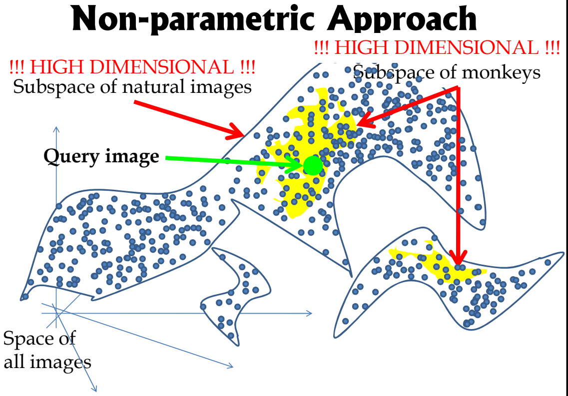
Parametric models



Non-parametric Approach



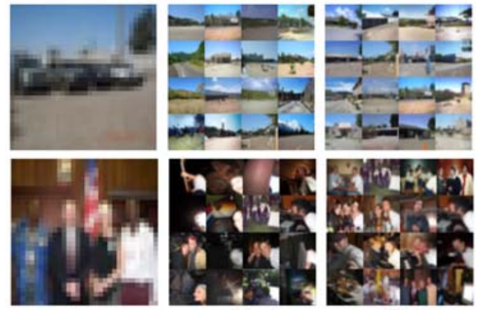
Non-parametric Approach



Nearest Neighbors in 80 million images

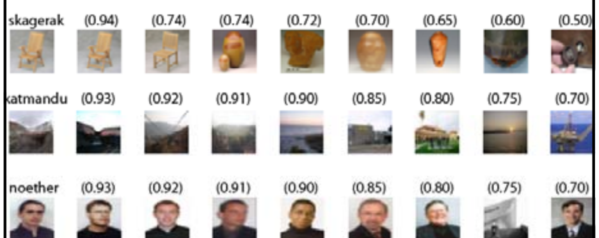
Differing density of images

Neighbors with Different Metrics

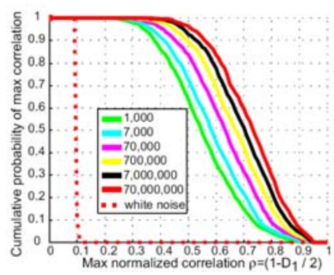


Examples

Normalized correlation scores



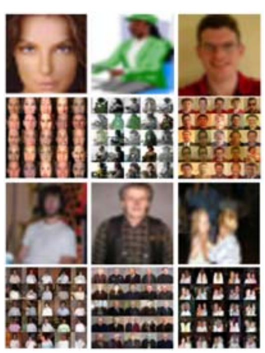
How Many Images Are There?



Note: D_1=D_{SSD}

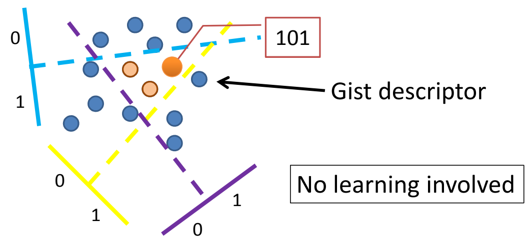
Person Recognition

- 23% of all images in dataset contain people
- Wide range of poses: not just frontal faces



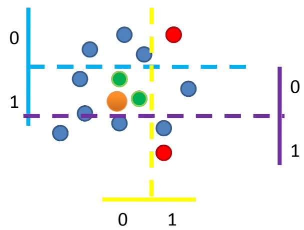
Locality Sensitive Hashing

- Gionis, A. & Indyk, P. & Motwani, R. (1999)
- Take random projections of data
- Quantize each projection with few bits



Learn Hash Codes using Boosting

- Modified form of BoostSSC [Shaknarovich, Viola & Darrell, 2003]
- Positive examples are pairs of similar images
- Negative examples are pairs of unrelated images



Learn threshold & dimension for each bit (weak classifier)

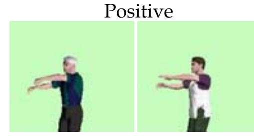
Fast Example-Based Pose Estimation

Shaknarovich, Viola & Darrell, 2003

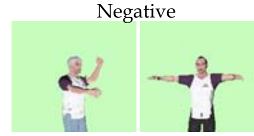
Input:

Desired output:

Positive



Negative



Overview of section

- Object detection with classifiers
- Boosting
  - Gentle boosting
  - Weak detectors
  - Object model
  - Object detection
- Nearest-Neighbor methods
- Multiclass object detection
- Context

Single category object detection
and the
"Head in the coffee beans problem"

"Head in the coffee beans problem"

Can you find the head in this image?

Multiclass object detection

Studying the multiclass problem, we can build detectors that are:

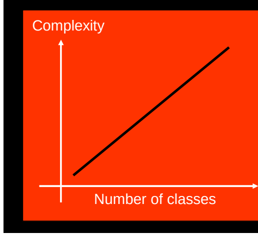
- more efficient,
- that generalize better, and
- more robust

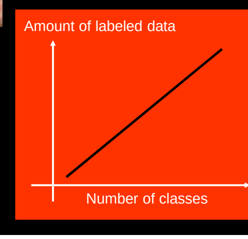
Multiclass object detection benefits from:

- Contextual relationships between objects
- Transfer between classes by sharing features

Multiclass object detection

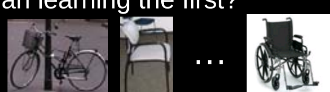
Multiclass object detection





Shared features

- Is learning the object class 1000 easier than learning the first?



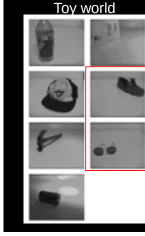
- Can we transfer knowledge from one object to another?
- Are the shared properties interesting by themselves?

Sharing invariances

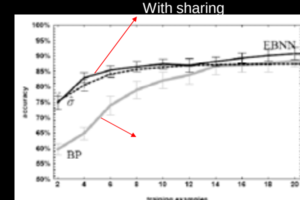
S. Thrun. Is Learning the n-th Thing Any Easier Than Learning The First? NIPS 1996

"Knowledge is transferred between tasks via a learned model of the invariances of the domain: object recognition is invariant to rotation, translation, scaling, lighting, … These invariances are common to all object recognition tasks".

Toy world



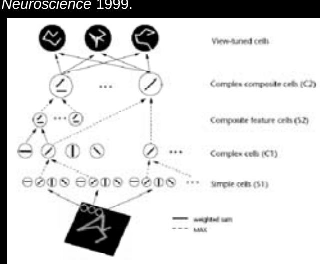
With sharing



Models of object recognition

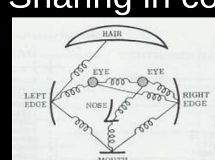
I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, 1987.

M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nature Neuroscience* 1999.

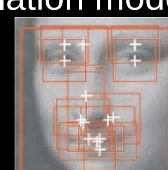


T. Serre, L. Wolf and T. Poggio. "Object recognition with features inspired by visual cortex". CVPR 2005

Sharing in constellation models



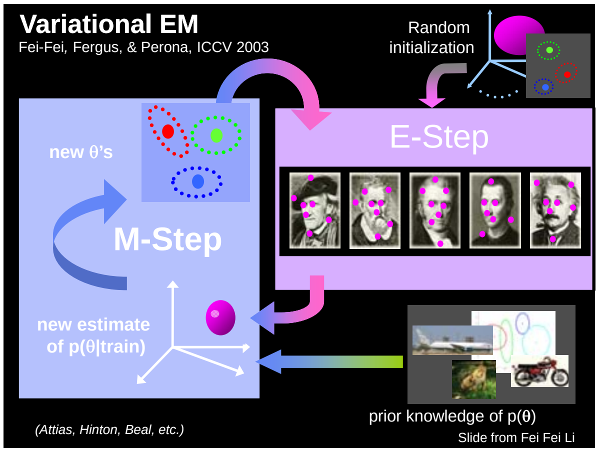
Pictorial Structures
Fischler & Elschlager, IEEE Trans. Comp. 1973



SVM Detectors
Heisele, Poggio, et. al., NIPS 2001

Constellation Model
Burl, Liung,Perona, 1996; Weber, Welling, Perona, 2000
Fergus, Perona, & Zisserman, CVPR 2003

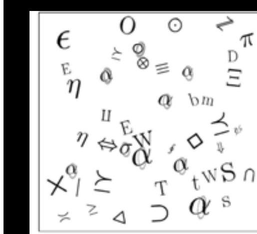
Model-Guided Segmentation
Mori, Ren, Efros, & Malik, CVPR 2004

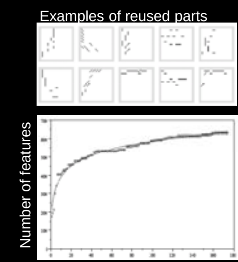


Reusable Parts

Krempp, Geman, & Amit "Sequential Learning of Reusable Parts for Object Detection". TR 2002

Goal: Look for a vocabulary of edges that reduces the number of features.





Sharing patches

- Bart and Ullman, 2004

For a new class, use only features similar to features that where good for other classes:

Proposed Dog features

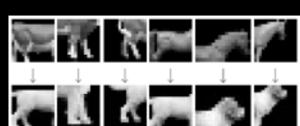


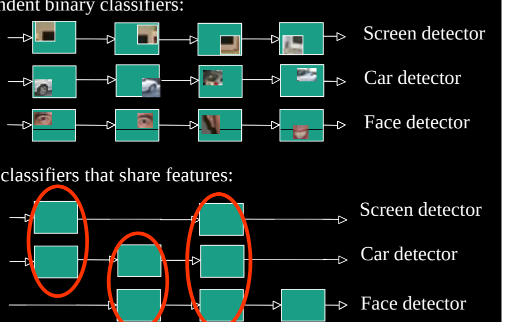
Figure 1. Feature adaptation. (a) Top row: features extracted from multiple images of cows (first three) and horses (last three), as described in section 3.1. Bottom row: features adapted to the dogs class by the proposed cross-generalization algorithm (section 3.2), using a single dog image.

Multiclass boosting

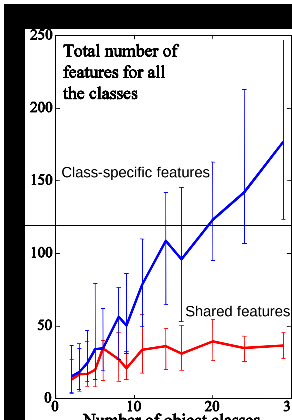
- Adaboost.MH (Shapire & Singer, 2000)
- Error correcting output codes (Dietterich & Bakiri, 1995; …)
- Lk-TreeBoost (Friedman, 2001)
- …

Shared features

- Independent binary classifiers:



Torralba, Murphy, Freeman. CVPR 2004. PAMI 2007

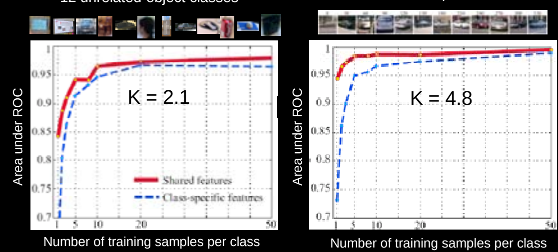


50 training samples/class
29 object classes
2000 entries in the dictionary

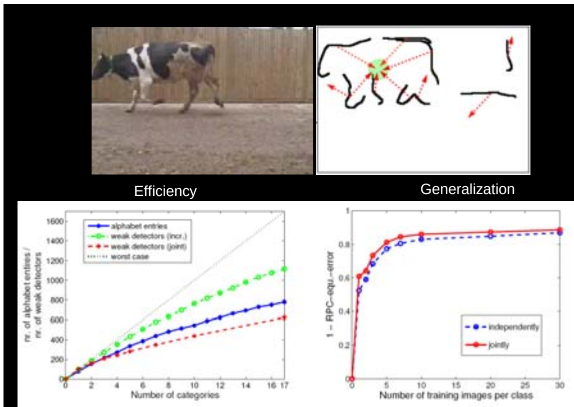
Results averaged on 20 runs
Error bars = 80% interval

Krempp, Geman, & Amit, 2002
Torralba, Murphy, Freeman. CVPR 2004

Generalization as a function of object similarities



Torralba, Murphy, Freeman. CVPR 2004. PAMI 2007



Opelt, Pinz, Zisserman, CVPR 2006

Some references on multiclass

- Caruana 1997
- Schapire, Singer, 2000
- Thrun, Pratt 1997
- Krempp, Geman, Amit, 2002
- E.L.Miller, Matsakis, Viola, 2000
- Mahamud, Hebert, Lafferty, 2001
- Fink 2004
- LeCun, Huang, Bottou, 2004
- Holub, Welling, Perona, 2005
- …

Overview of section

- Object detection with classifiers
- Boosting
  - Gentle boosting
  - Weak detectors
  - Object model
  - Object detection
- Nearest-Neighbor methods
- Multiclass object detection
- Context

Context

What do you think are the hidden objects?

Context

What do you think are the hidden objects?

Even without local object models, we can make reasonable detections!

Global scene representations

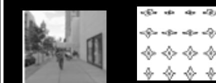
Bag of words

Sivic, Russell, Freeman, Zisserman, ICCV 2005
Fei-Fei and Perona, CVPR 2005
Bosch, Zisserman, Munoz, ECCV 2006

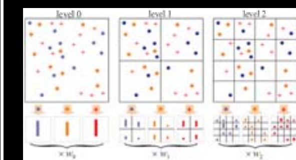
Non localized textons

Walker, Malik. Vision Research 2004 . . .

Spatially organized textures



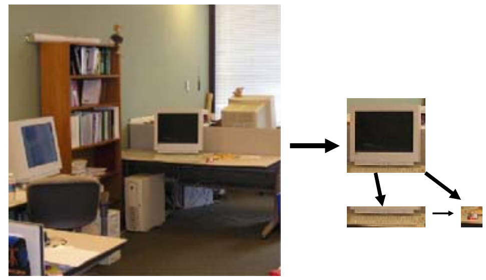
M. Gorkani, R. Picard, ICPR 1994
A. Oliva, A. Torralba, IJCV 2001



S. Lazebnik, et al, CVPR 2006 . . .

Spatial structure is important in order to provide context for object localization

Context: relationships between objects

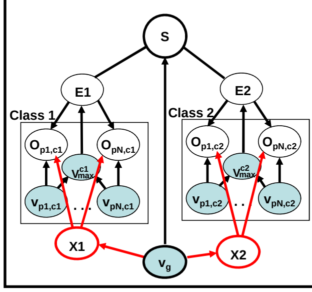


Detect first simple objects (reliable detectors) that provide strong contextual constraints to the target (screen -> keyboard -> mouse)

Context

- Murphy, Torralba & Freeman (NIPS 03)

Use global context to predict presence and location of objects



Keyboards

Context

- Fink & Perona (NIPS 03)

Use output of boosting from other objects at previous iterations as input into boosting for this iteration

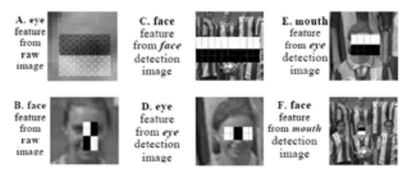
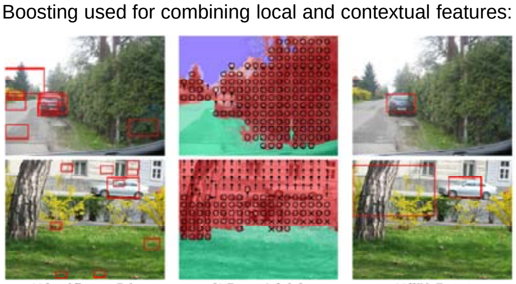


Figure 5: A-E. Emerging features of eyes, mouths and faces (presented on windows of raw images for legibility). The windows' scale is defined by the detected object size and by the map mode (local or contextual). C. faces are detected using face detection maps H^{Face}, exploiting the fact that faces tend to be horizontally aligned.

Context

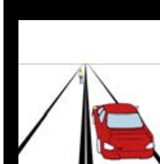
- Hoiem, Efros, Hebert (ICCV 05)

Boosting used for combining local and contextual features:

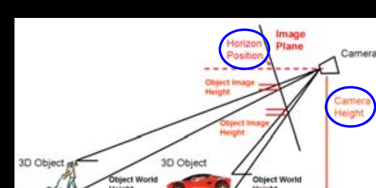


(a) Local Features Only (b) Geometric Labels (c) With Context

3d Scene Context



Image

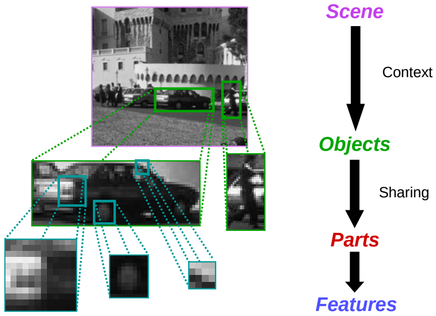


World

[Hoiem, Efros, Hebert ICCV 2005]

Context (generative model)

- Sudderth, Torralba, Freeman, Willsky (ICCV 2005).



Some references on context

With a mixture of generative and discriminative approaches

- Strat & Fischler (PAMI 91)
- Torralba & Sinha (ICCV 01),
- Torralba (IJCV 03)
- Fink & Perona (NIPS 03)
- Murphy, Torralba & Freeman (NIPS 03)
- Kumar and M. Hebert (NIPS 04)
- Carbonetto, Freitas & Barnard (ECCV 04)
- He, Zemel & Carreira-Perpinan (CVPR 04)
- Sudderth, Torralba, Freeman, Wilsky (ICCV 05)
- Hoiem, Efros, Hebert (ICCV 05)
- …

A car out of context …