Online Learning
&
Vision

Horst Bischof

Inst. for Computer Graphics and Vision
Graz University of Technology

Off-line learning

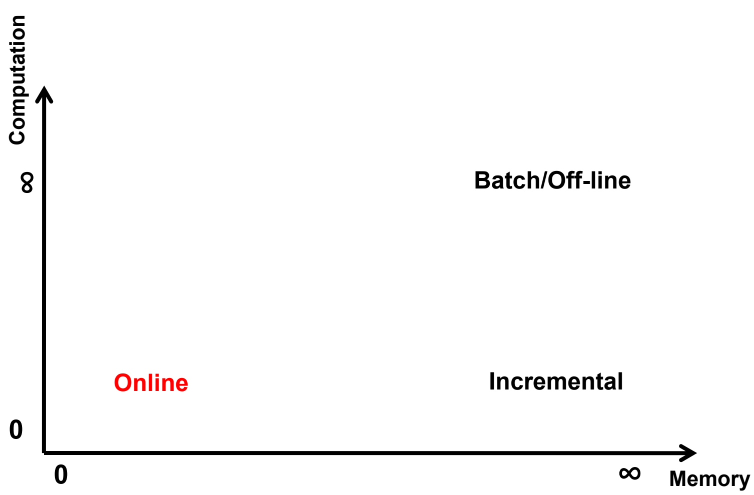
On-line learning

Labeled Information

Learning Algorithm

Teacher

Online Learning in Perspective



Why on-line learning?

Too much training data to fit in memory

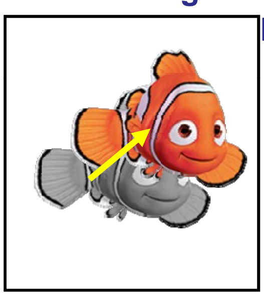
Sample generation process

Changing process

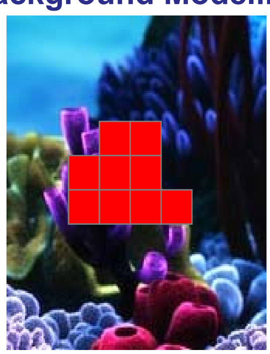
Specializing
- Forget irrelevant information

Applications

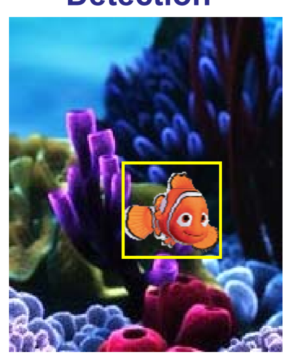
Tracking



Background Modeling



Detection



Agenda

PART I

On-line Learning

– On-line boosting

Applications

– Case Study

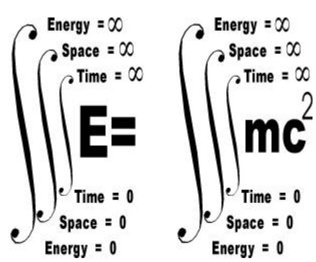
PART II

Problems

– Limitations

– The “Truth of PARTI”

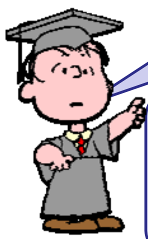
Possible Solutions



ON-LINE BOOSTING FOR FEATURE SELECTION

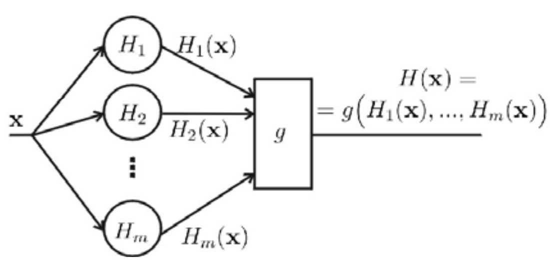
H. Grabner and H. Bischof. **On-line boosting and vision.** CVPR, 2006.

Sir, could you please tell me what boosting is?



Boosting need just some guy who is a little bit better than guessing.

Meta Learning

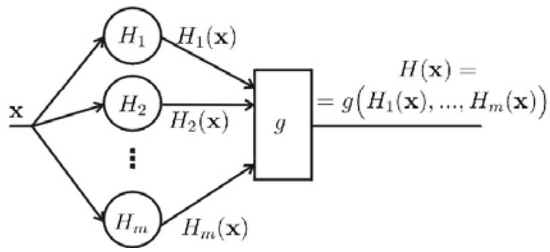


"How to combine rules of thumb into a single prediction rule?"

Bagging, Boosting
- different trainings samples

Stacking
- different lerning algorithm

Boosting



Boosting

- general method for improving the accuracy of any learning algorithm
- combine (weak) classifier (linear combination)

"Take (weighted) majority vote of rules of thumb"

AdaBoost

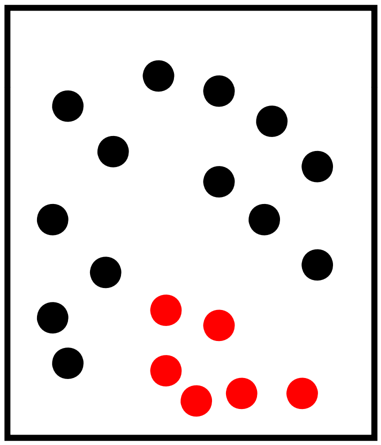
- **Adaptive boosting**
  - each sample has a associated weight

    Y. Freund and R. Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting.** Journal of Computer and System Sciences, 1997.
  - weak classifier
    - **require performance better than guessing (i.e. error < 50% for a binary problem)**
  - strong classifier
    - **linear combination of N weak classifiers**

$$h^{strong}(x) = \text{sign}\Big(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(x)\Big)$$

  - widely used
    - text recognition, routing, learning problems in natural language processing,...
    - image retrieval, generic object detection and recognition, active shape model,...

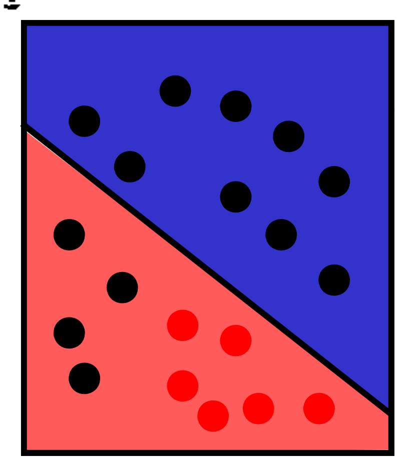
AdaBoost



Given:
- set of labeled training samples
- weight distribution over them

Algorithm:
```
for n = 1 to N
    - train a weak classifier using
      samples and weight dist.
    - calculate error
    - calculate weight
    - update weight dist.
next
```

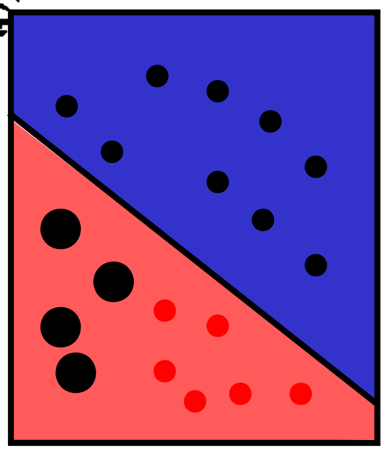

AdaBoost



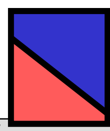
Given:
- set of labeled training samples
- weight distribution over them

Algorithm:
```
for n = 1 to N
    - train a weak classifier using
      samples and weight dist.
    - calculate error
    - calculate weight
    - update weight dist.
next
```

AdaBoost



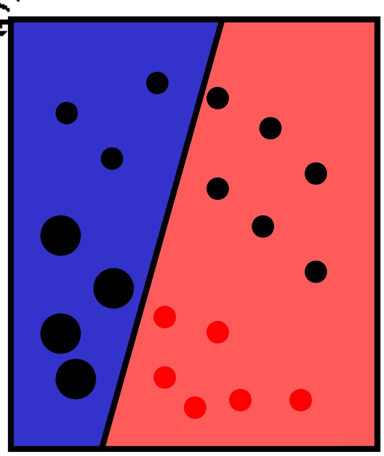
$\alpha_1 \cdot$



Given:
- set of labeled training samples
- weight distribution over them

Algorithm:
```
for n = 1 to N
    - train a weak classifier using
      samples and weight dist.
    - calculate error
    - calculate weight
    - update weight dist.
next
```

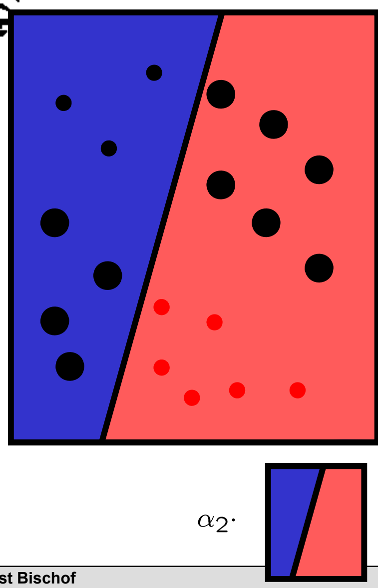

AdaBoost



Given:
- set of labeled training samples
- weight distribution over them

Algorithm:
```
for n = 1 to N
    - train a weak classifier using
      samples and weight dist.
    - calculate error
    - calculate weight
    - update weight dist.
next
```

AdaBoost



Given:

- set of labeled training samples
- weight distribution over them

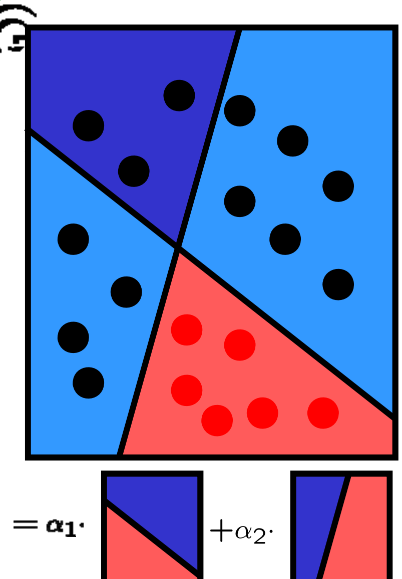
Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

AdaBoost



Given:

- set of labeled training samples
- weight distribution over them

Algorithm:

for n = 1 to N

- train a weak classifier using samples and weight dist.
- calculate error
- calculate weight
- update weight dist.

next

Result:

$$h^{strong}(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h_n^{weak}(x)\right)$$

AdaBoost Algorithm

Require: trainingsset $S = \langle\langle \mathbf{x_1}, y_1\rangle, ..., \langle \mathbf{x_L}, y_L\rangle\rangle, x_l \in X, y_l \in Y = \{-1, +1\}$
- initialize $w_{l,1} = \frac{1}{L}, l = 1, ..., L$

for $n = 1, 2, .., N$ **do**
- train weak classifier in respect to W_n

$$h_n^{weak} : X \to \{-1, +1\}$$

- measure "goodness"

$$e_n = \text{Prob}_{l \sim W_n}[h_n^{weak}(\mathbf{x_l}) \neq y_l] = \sum_{l: h_n^{weak}(\mathbf{x_l}) \neq y_l} w_{n,l}$$

- exit if $e_n = 0$ oder $e_n > \frac{1}{2}$
- calculate voting factor

$$\alpha_n = \frac{1}{2} \cdot \ln\left(\frac{1 - e_n}{e_n}\right)$$

- update prob. distribution

$$w_{n+1,l} = \frac{w_{n,l}}{Z_n} \cdot \begin{cases} \exp(-\alpha_n) & \text{if } h^{weak}(\mathbf{x_l}) = y_l \\ \exp(\alpha_n) & \text{if } h^{weak}(\mathbf{x_l}) \neq y_l \end{cases}$$

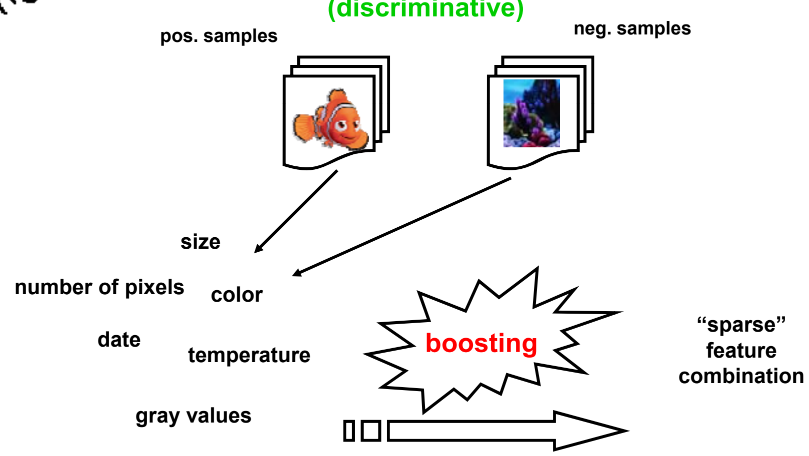
- where Z_n is a normalization factor (such that W_{n+1} is a distribution)

end for
- output the final (strong) classifier

$$h^{strong}(x) = \text{sign}\left(\sum_{n=1}^{N} \alpha_n \cdot h^{weak_n}(x)\right)$$

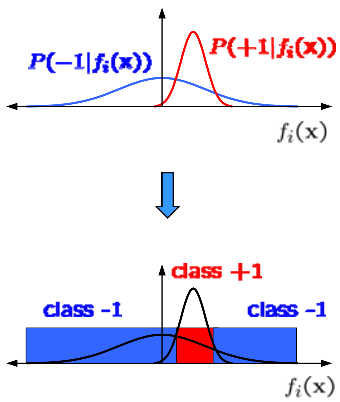
Boosting for Feature Selection

learning problem (discriminative)



Feature = Weak Classifier

Each feature corresponds to a weak classifier



1. feature f_i
2. evaluate feature f_i on image patches x
3. apply a learning algorithm $\mathcal{L}$

$$f_i(x) \xrightarrow{\mathcal{L}} h^{weak}(x)$$

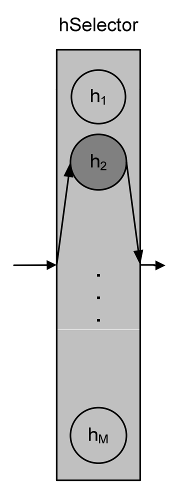
– (only) requires: error < 50 %

Off-line Boosting for Feature Selection

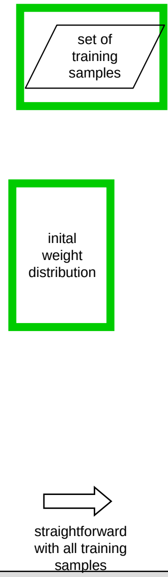
Introducing "Selector"

– selects one feature from its local feature pool

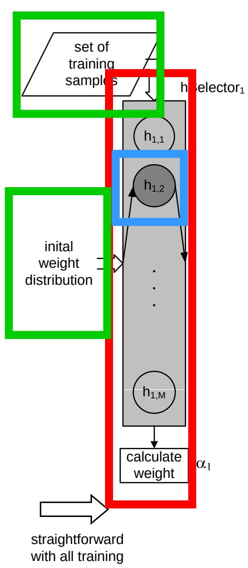
Boosting is performed on the Selectors and not on the weak classifiers directly.



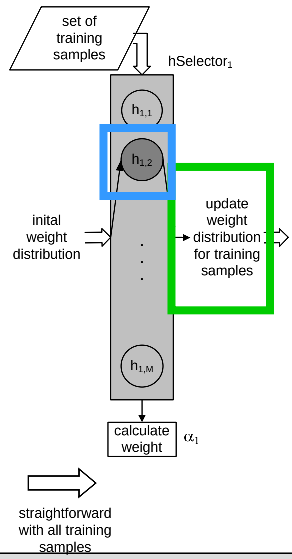
Off-line Boosting for Feature Selection



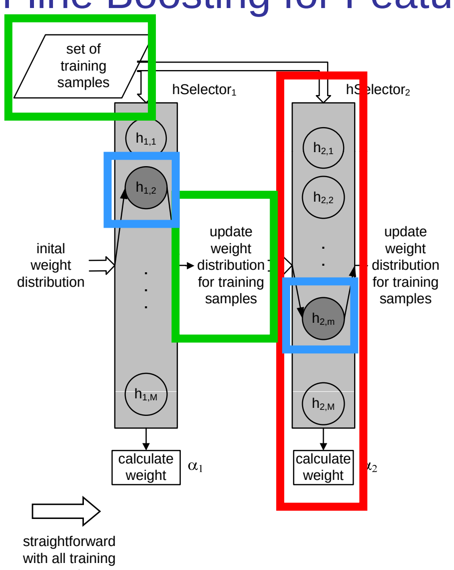
Off-line Boosting for Feature Selection



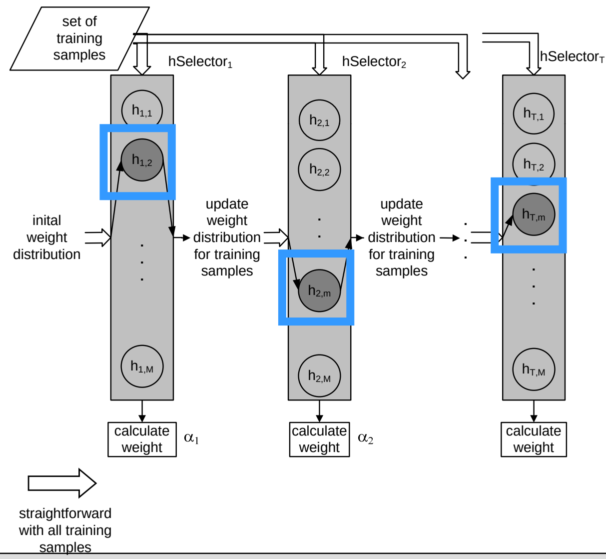
Off-line Boosting for Feature Selection



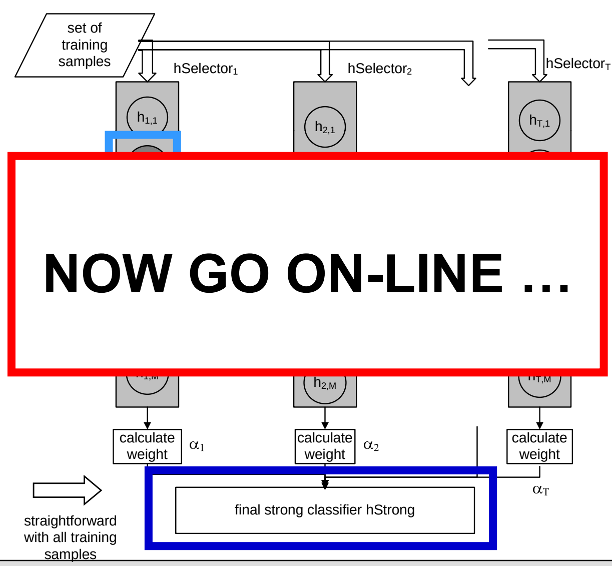
Off-line Boosting for Feature Selection



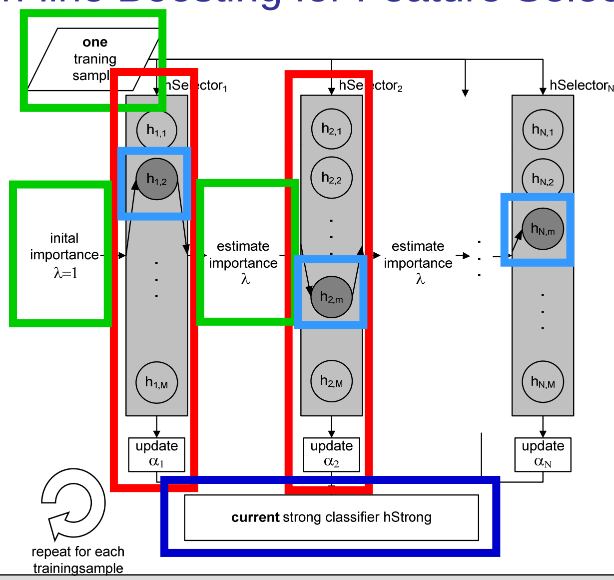
Off-line Boosting for Feature Selection



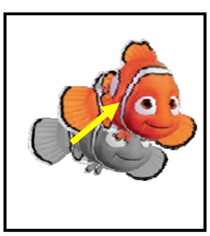
Off-line Boosting for Feature Selection



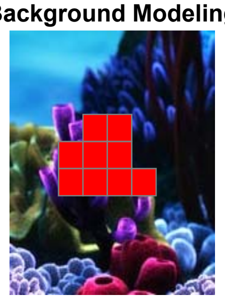
On-line Boosting for Feature Selection



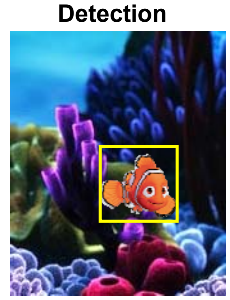
Tracking



Background Modeling



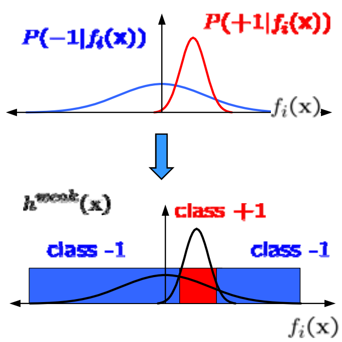
Detection



APPLICATIONS

Boosting for Feature Selection

Each feature corresponds to a weak classifier



K. Tieu and P. Viola. **Boosting Image Retrival,** CVPR 2000

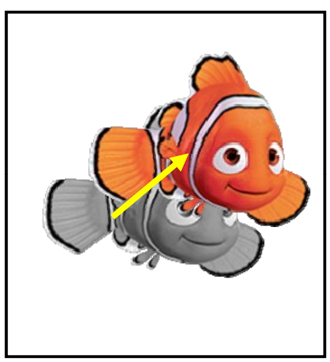
Features

- Haar-like wavelets
- Orientation histograms
- Locally binary patterns (LBP)

Fast computation using efficient data structures

- integral images
- integral histograms

F. Porikli. **Integral histogram: A fast way to extract histograms in cartesian spaces.** CVPR 2005.



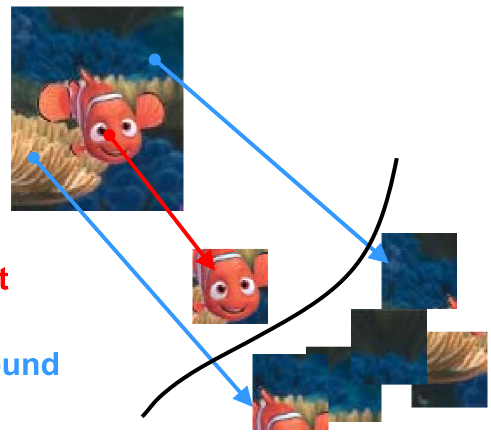
TRACKING

H. Grabner, M. Grabner and H. Bischof. **Real-Time Tracking via On-line Boosting.** BMVC, 2006.

Tracking as Classification

S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

Tracking as binary classification



object

vs.

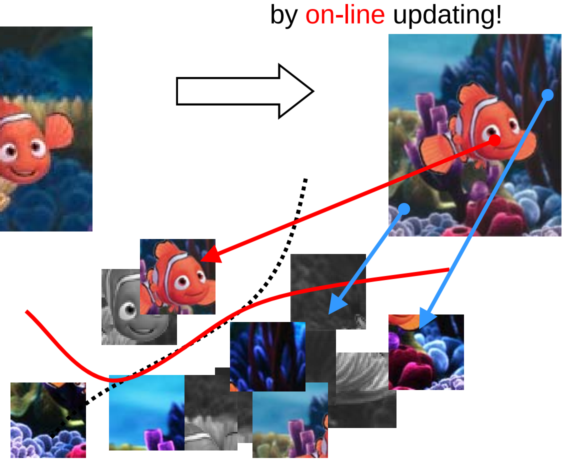
background

Tracking as Classification

S. Avidan. **Ensemble tracking**. CVPR 2005.
J.Wang, et al. **Online selecting discriminative tracking features using particle filter**. CVPR 2005.

Tracking as binary classification

Object <u>and</u> background changes are robustly handled by on-line updating!

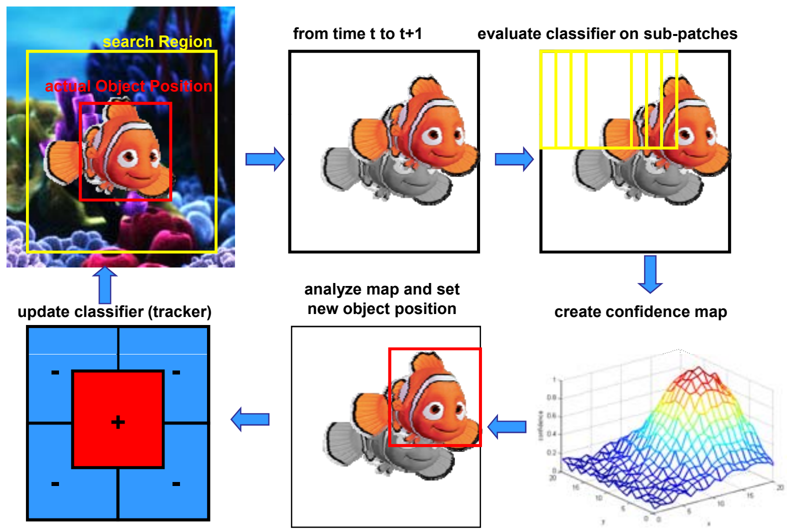


object

vs.

background

Template based Object Tracking



Tracking Examples

Feature Exchange

Tracking Examples

Object Detector

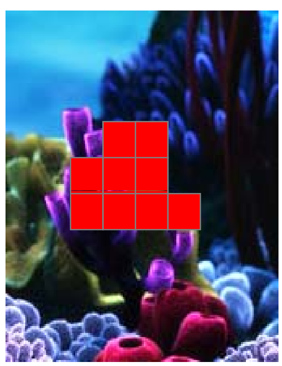
Fixed Training set
General object detector

Object Tracker

On-line update
Object vs. Background

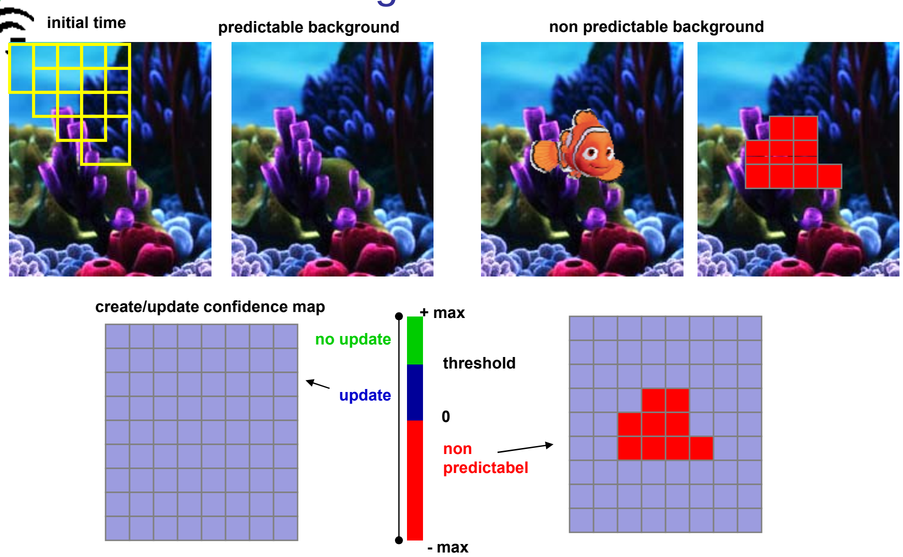
Off-line Boosting for Feature Selection

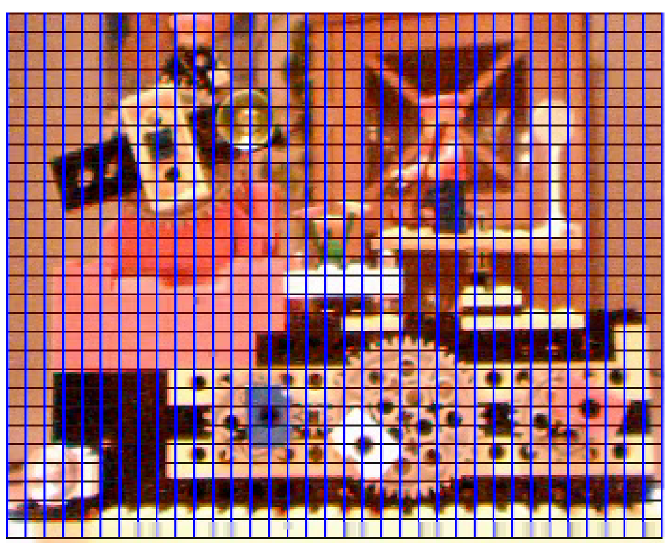
On-Line Boosting for Feature Selection

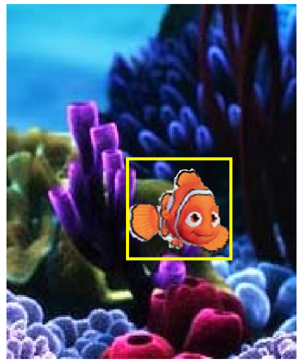


BACKGROUND MODELING

Background Model







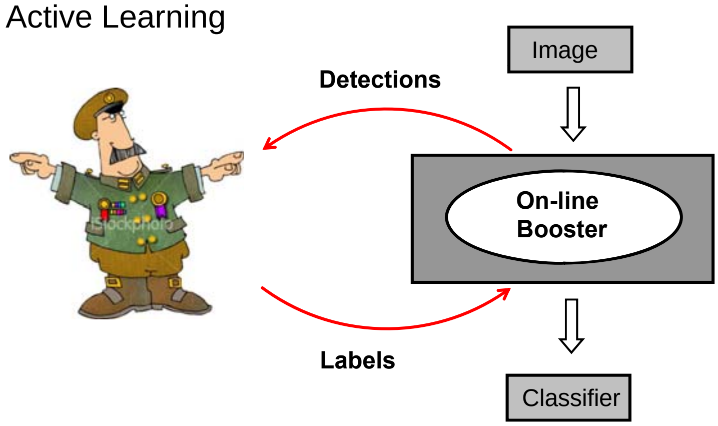
OBJECT DETECTION

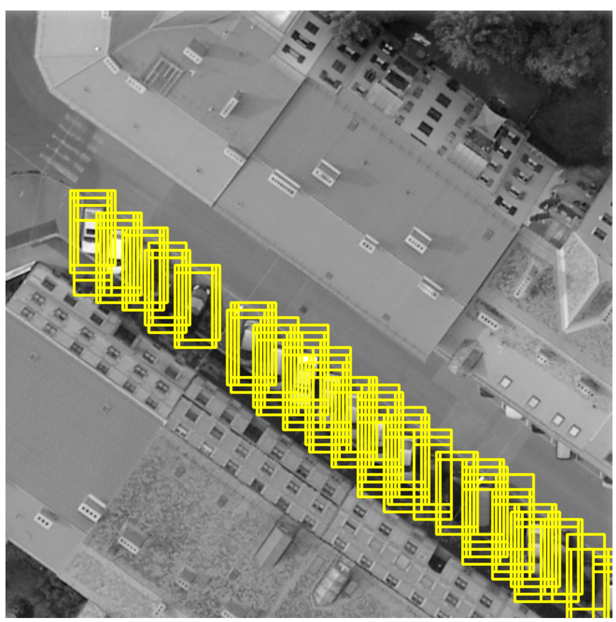
H. Grabner, T. Nguyen, B. Gruber, H. Bischof: **On-line Boosting-based Car Detection from Arial Images**
ISPRS Journal of Photogrammetry & Remote Sensing , 2007

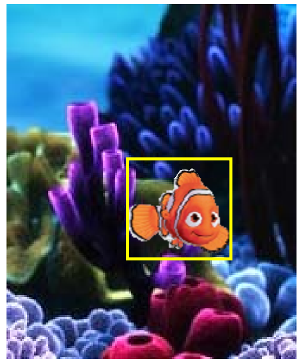
Car Detection

"Fewer Clicks - Less Frustration"

Active Learning



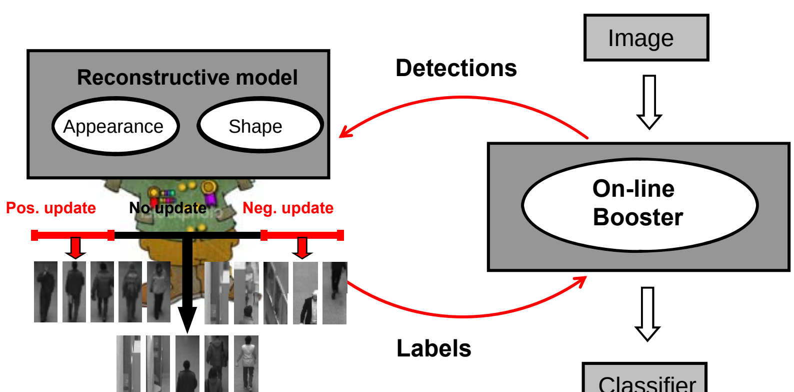




OBJECT DETECTION

P. Roth, H. Grabner, D. Skoncaj, H. Bischof, A. Leonardis: **On-line conservative learning for person detection**, VS- PETS, 2005

Conservative Learning

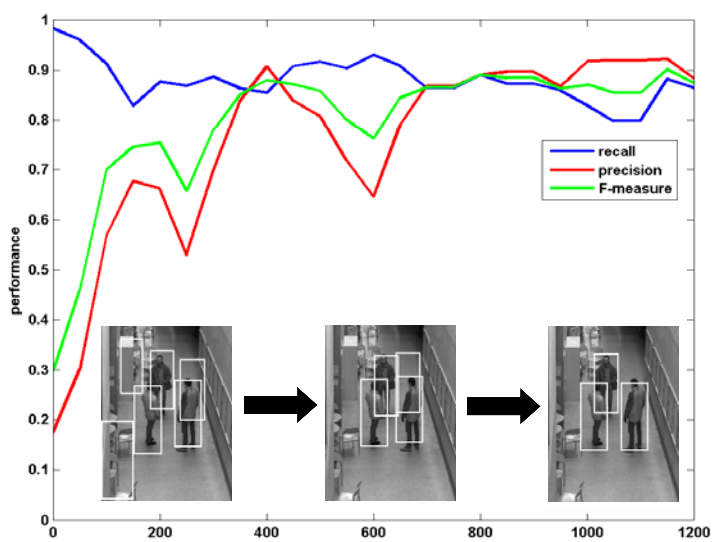


Improving Performance

initial classifier

on-line updated classifier

Improving Performance



ICG

Any Problems?

unsupervised

↓

DRIFTING!

Horst Bischof | Online Learning

ICG

$\langle \mathbf{x}_i, ? \rangle$

$\langle \mathbf{x}_i, y_i \rangle$

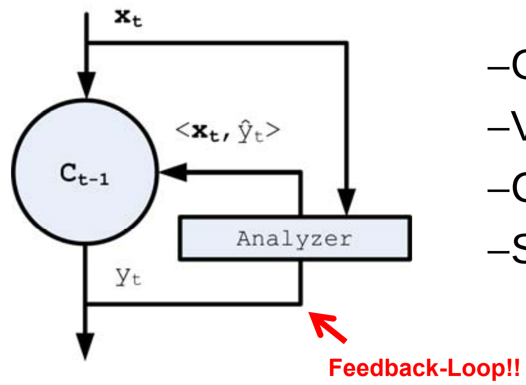
THE ROLE OF SUPERVISION

Supervised vs. Unsupervised learning

Horst Bischof | Online Learning

Update Strategies

Unlabeled sample $\mathbf{x}_t$, who to choose $\hat{y}_t$?



–Oracle
–Verification
–Co-training
–Self-training

Update Strategies

- We are interested in building systems that learn 24/7/52

- Can we guarantee that nothing explodes?

Update Strategies

- Human interaction
  - If human is O.K
- Verification by:
  - Geometry (if possible)
  - Redundancy (Data)
  - Multiple Classifiers (Cons. Learn)
  - Self-Learning

ICG

A few examples und trials to overcome the problems.

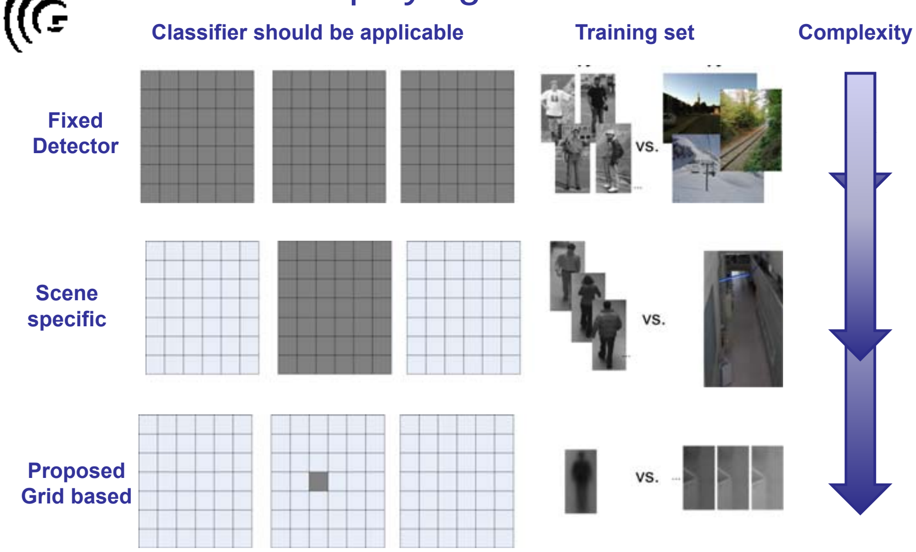
HOW TO LIMIT DRIFTING?

Grid-based Person Detector

We want to build a system which runs 24 hours a day, 7 days a week!

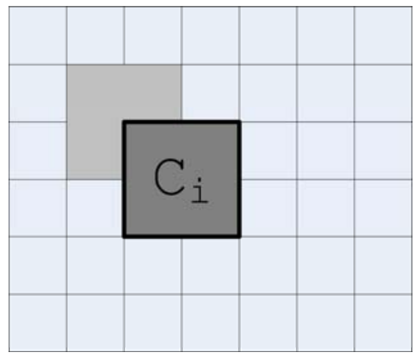
Simplify the problem
such that we can use a
fixed update strategy,
no drifting problem.

Simplifying the Problem



Classifier Grid

Trainingset



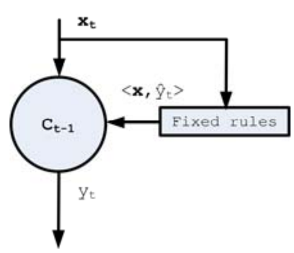
H. Grabner, P. Roth, M. Grabner and H. Bischof. **Autonomous Learning a Robust Background Model for Change Detection**. PETS WS 2006.

One On-line classifier for each grid element

Relations to background modeling

Simple Problem in time and space.

Fixed Update Rules

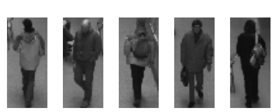


No dependencies between the update and the current model ⇨ stable by design

Positive updates

– From a fix set

$$\langle \mathbf{x}, +1 \rangle, \quad \mathbf{x} \in \mathcal{X}^+$$



– Correct by definition

Negative updates

– Current patch

$$\langle \mathbf{x}_{i,t}, -1 \rangle$$

– Correct most of the time, wrong with

$$P(\mathbf{x}_i = \text{person}) = \frac{\#p_i}{\Delta t}$$

Experiments and Results

Toy Example

Public Sequences

– PETS 2006
– and Caviar Sequences

Parameter (constant)

– Ground plane estimation (scale estimation)
– Positive dataset contains only **1** sample (mean image)
– 10 selectors each with 20 weak classifier features

Result and Comparison

Dalal/Triggs (generic detector)

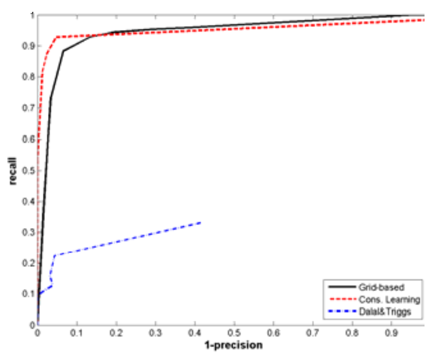
This approach

Conservative learning (scene adaption)

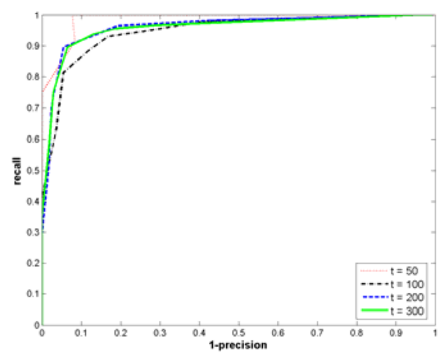
ICG

CoffeeCam Results

ROC

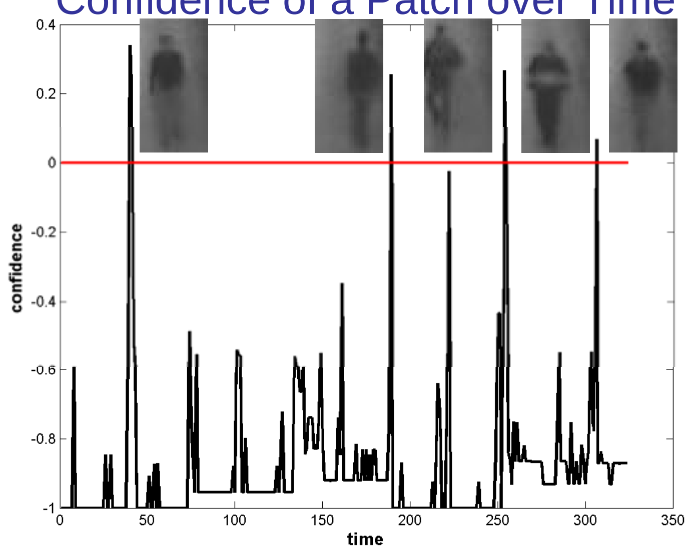


Convergence Speed



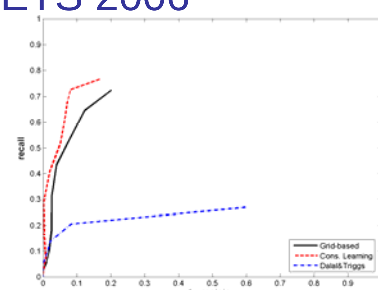
ICG

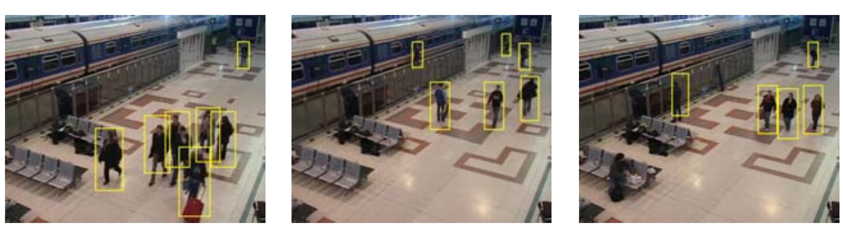
Confidence of a Patch over Time



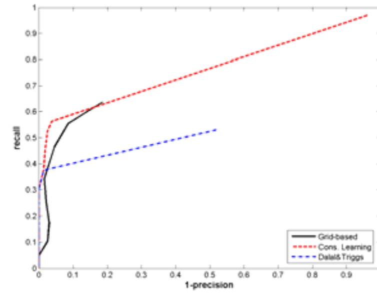
I'm NOT a simple Background Model

Results: PETS 2006



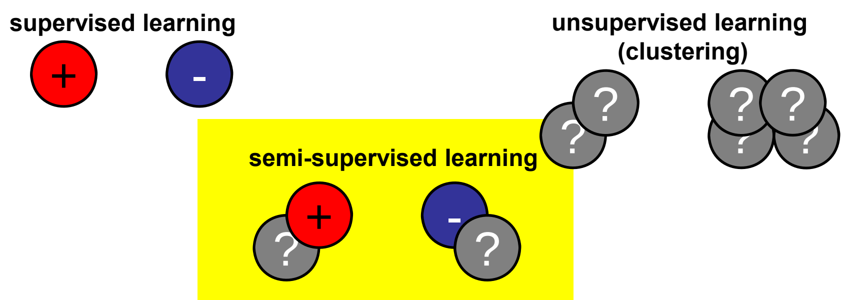


Results: Caviar



Are these fix updates sufficient enough?

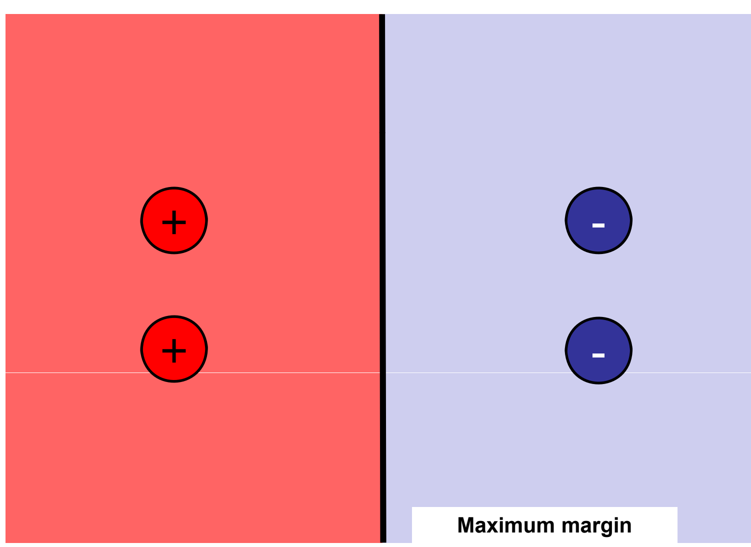
Can we do it even better?



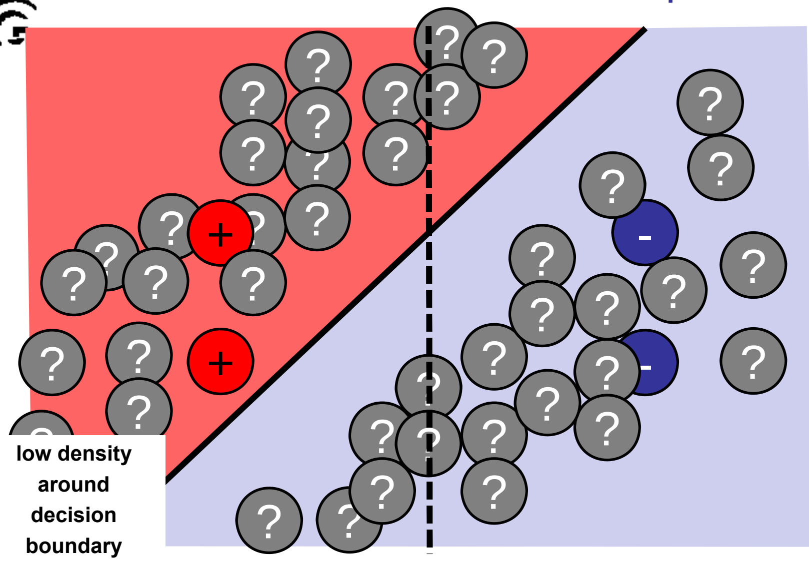
SEMI-BOOST AND VISUAL SIMILARITY LEARNING

C. Leistner, H. Grabner, H. Bischof. **Semi-Supervised Boosting using Visual Similarity Learning.** CVPR, 2008.

Supervised learning

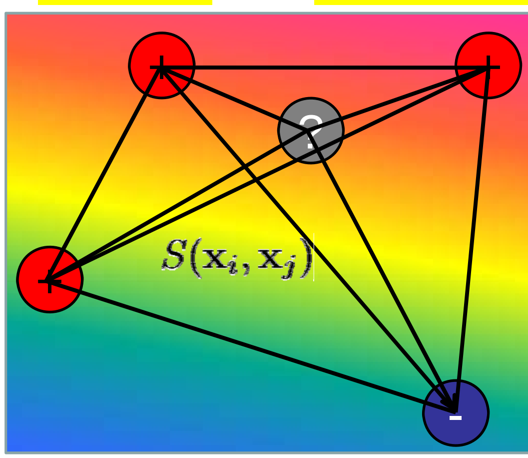


Can Unlabeled Data Help?



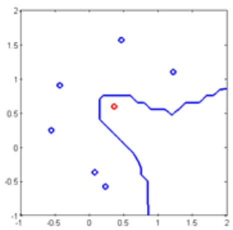
Semi Boosting

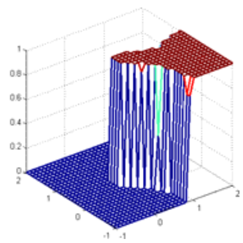
"Boosting with a graph inspired regularization"



$$H(\mathbf{x}) = \sum_{n=1}^{N} \alpha_n \cdot h_n(\mathbf{x})$$

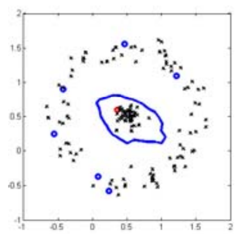
Mallapragada, Jin, Jain, Liu, **SemiBoost: Boosting for Semi-supervised Learning**, *Technical Report,* Department of Computer Science and Engineering, Michigan State University

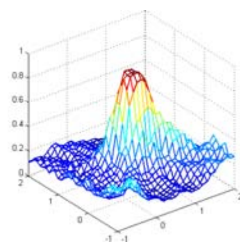




Boosting

SemiBoost



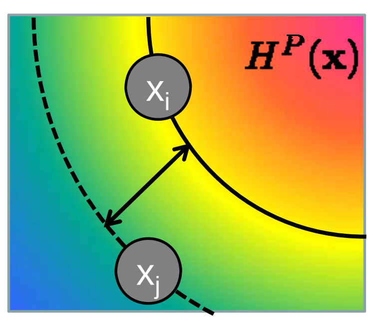


Similarities as Prior Classifier

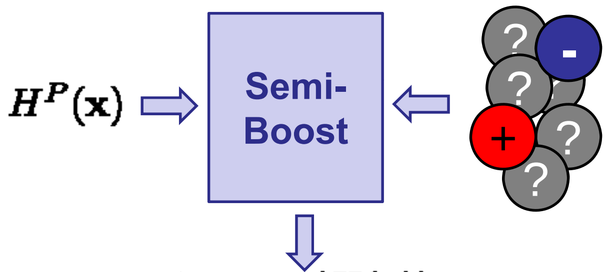
$$S(\mathbf{x}_i, \mathbf{x}_j)$$

$$\Downarrow$$

$$|H^P(\mathbf{x}_i) - H^P(\mathbf{x}_j)|$$

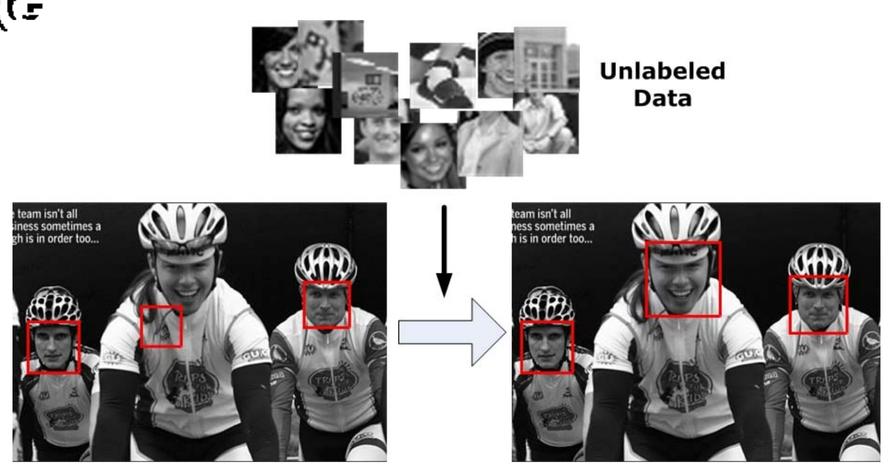


Classifier Improvement

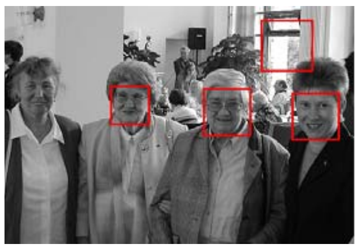


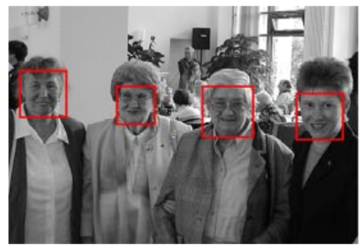
$$\hat{y} = \mathrm{sign}(H(\mathbf{x}))$$
$$\hat{y} = \mathrm{sign}(H^P(\mathbf{x}) + H(\mathbf{x}))$$

Note, this is NOT a simple sum-rule, since training of H(x) depends on H^P(x)!



Face Detector | Improved Results





Going On-line

$$\operatorname{sign}(H^P(\mathbf{x}) + H(\mathbf{x}))$$

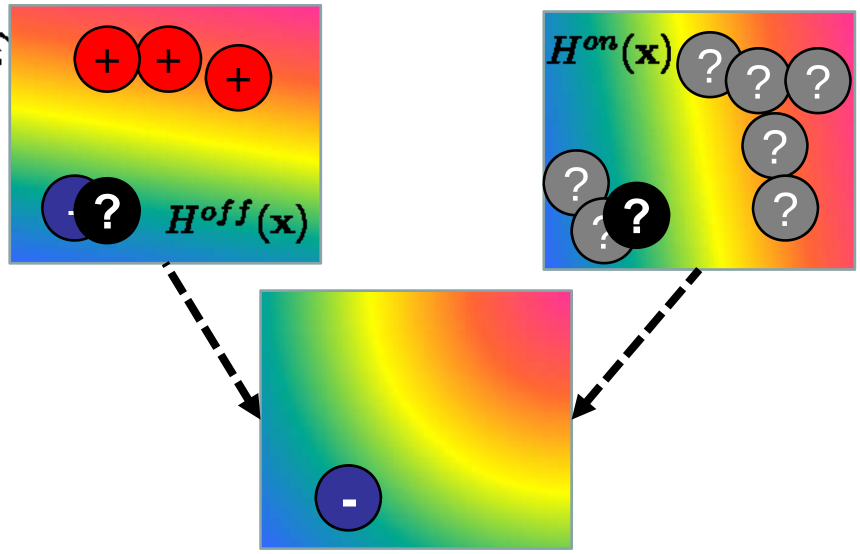
Classifier Fusion

On-line SemiBoost

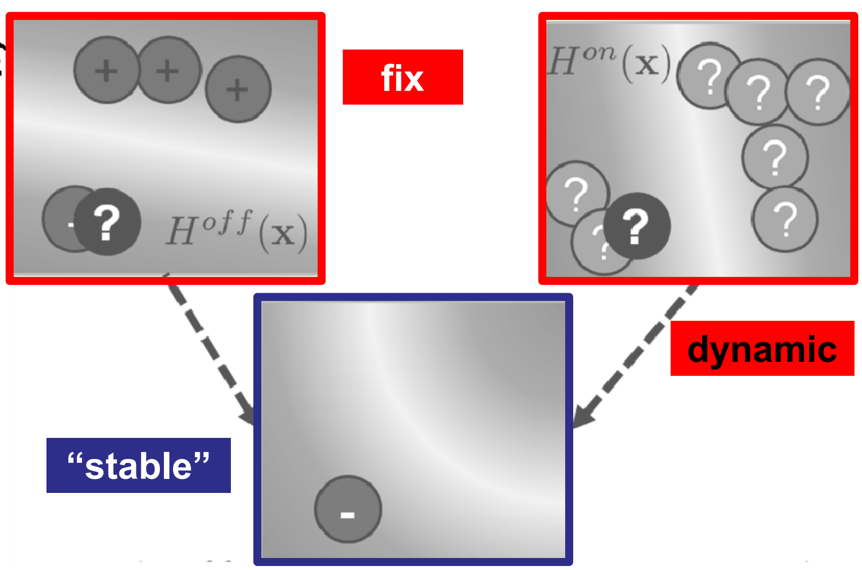
Accepted at ECCV08

$$\operatorname{sign}\left(H^{off}(\mathbf{x}) + H^{on}(\mathbf{x})\right)$$

Dynamic classifier Fusion



$$\mathrm{sign}\left(H^{off}(\mathbf{x}) + H^{on}(\mathbf{x})\right)$$



Object Detector | Our approach | Object Tracker

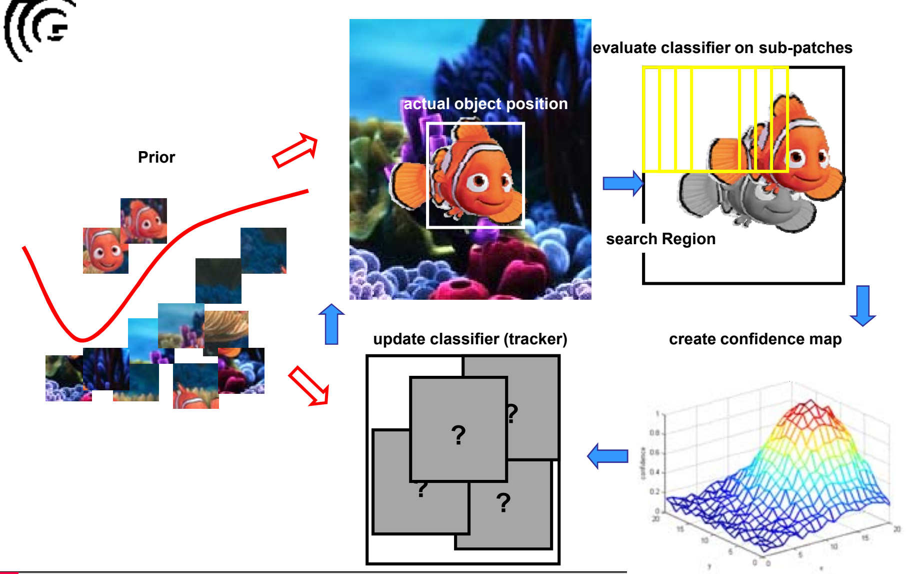
Fixed Training set
General object detector

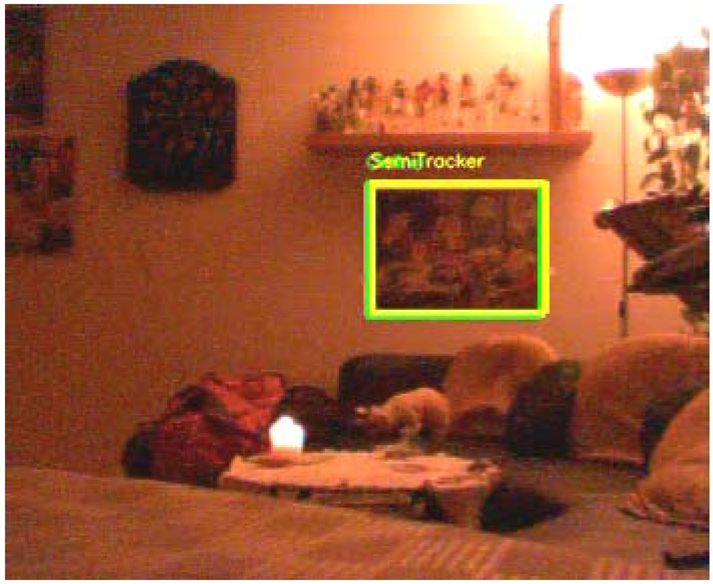
Fixed Prior for updating an
Adaptive on-line classifier

On-line update
Object vs. Background

ROBUST TRACKING

Modified Tracking Loop



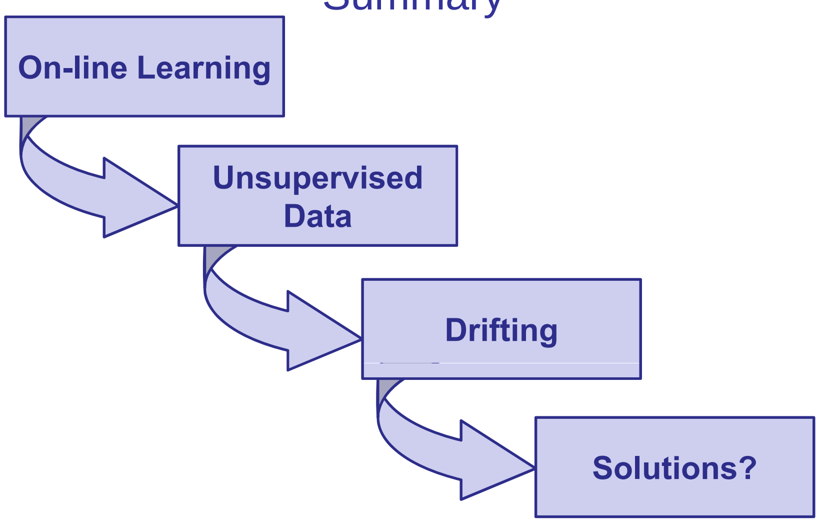


longterm.avi

Examples

Examples

Summary



Acknowledgments

Material presented is based on:

Joint work with my students: H. Grabner, M. Grabner, C. Leistner, P. Roth

Funding provided by:

Austrian Joint Research Project Cognitive Vision under sub-projects S9103-N03 and S9104-N04

FIT-IT Program funded by BMVIT under Project AUTOVISTA, EVis

Doctorial College: Confluence of Vision & Graphics