Multiclass object detection:

nested cascades and coupled components in multiclass classifiers

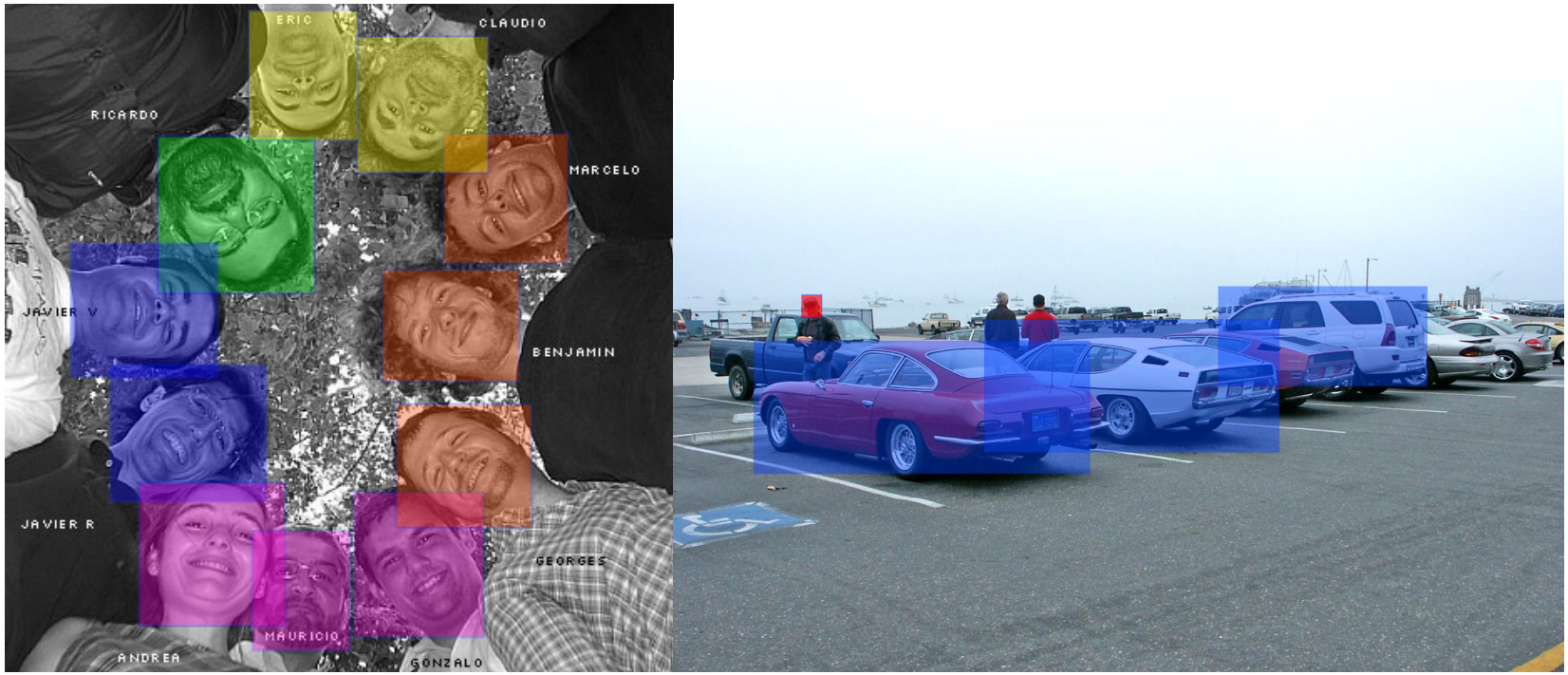
Rodrigo Verschae

Department of Electrical Engineering, Universidad de Chile

UPF
September 9, 2008

Multiclass Object detection

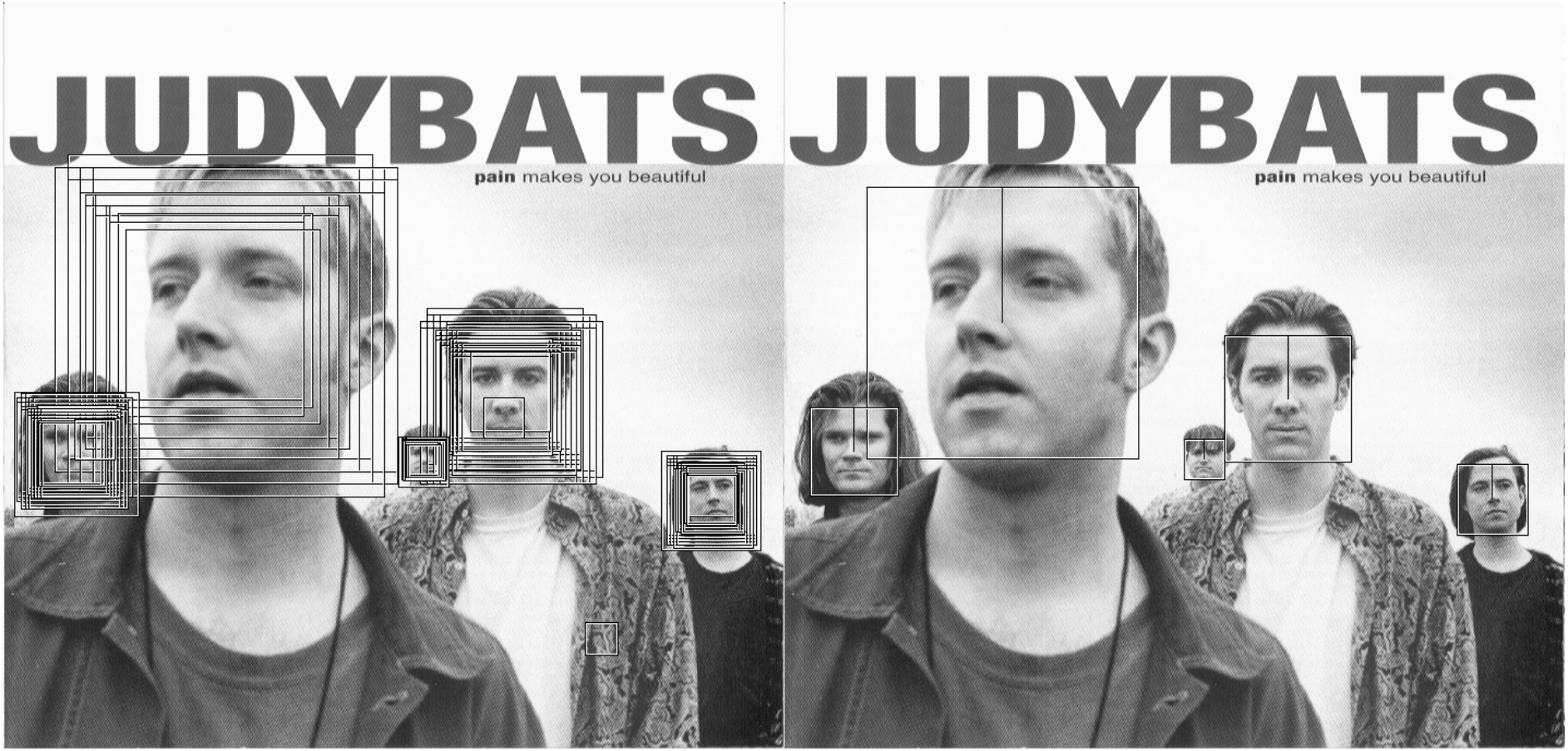
Object detection: For a given set of object classes, to find, if any, the instances of these classes appearing on the image.



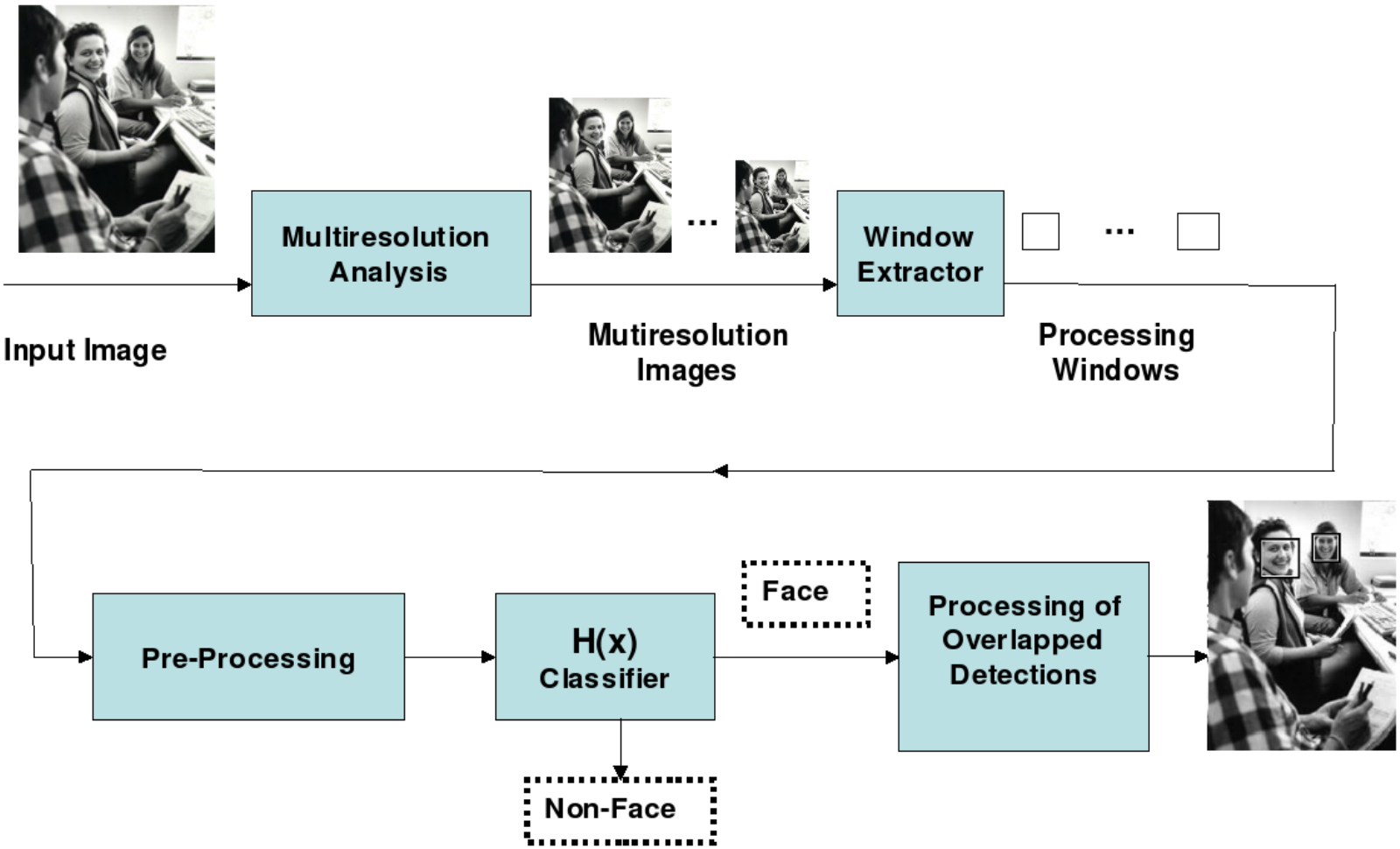
1. Introduction
   - Motivation
   - Cascade classifiers
2. Nested Cascade of Boosted Classifier
   - Adaboost
   - Nested Cascade Classifiers
3. Multiclass Object Detection
   - A multiclass extension of Adaboost
   - Weak classifier
   - Training procedures
4. Results, and future work
   - Experiment set up
   - Example results and Comments
   - Future work

Multiscale search

Multiscale search



Detection Architecture



What do we want from Multiclass object detector?

Use

- High accuracy
  - high detection rate ($> 99.9\%$)
  - low false postive rate ($< 10^{-6}$)
- Fast detection
- Robustness to noise, illumination, etc.

What do we want from Multiclass object detector?

Use

- High accuracy
  - high detection rate ($> 99.9\%$)
  - low false postive rate ($< 10^{-6}$)
- Fast detection
- Robustness to noise, illumination, etc.

Training procedure

- Fast training
- Good results when trained using small training sets

What do we want from Multiclass object detector?

Use

- High accuracy
  - high detection rate ($> 99.9\%$)
  - low false postive rate ($< 10^{-6}$)
- Fast detection
- Robustness to noise, illumination, etc.

Training procedure

- Fast training
- Good results when trained using small training sets

Scalablility with the number of classes

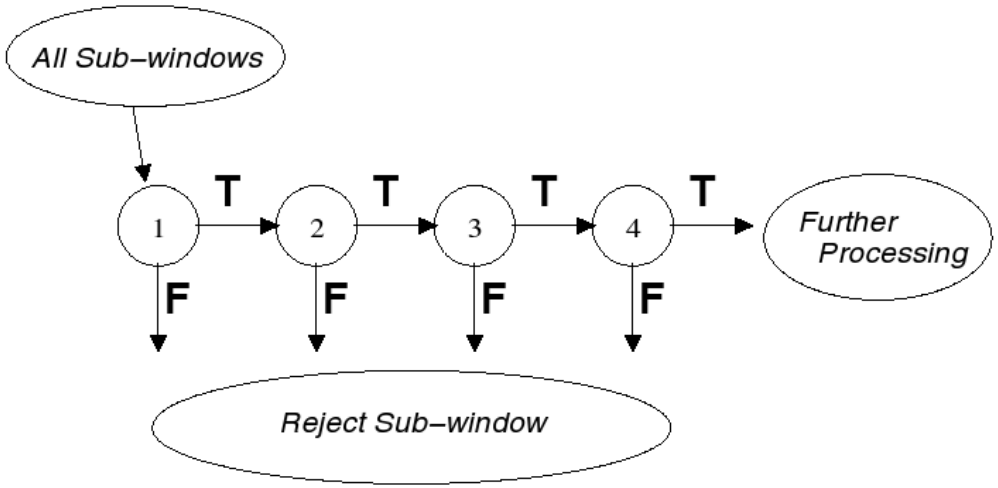
- Training time
- Detection time
- Accuracy

Cascade classifiers [Viola and Jones 2001]

- Very asymmetric problem
- In an image, most windows correspond to the background
- Most non-object windows are very different from object windows

$\longrightarrow$ Average processing time depends mainly on the processing time of non-object windows

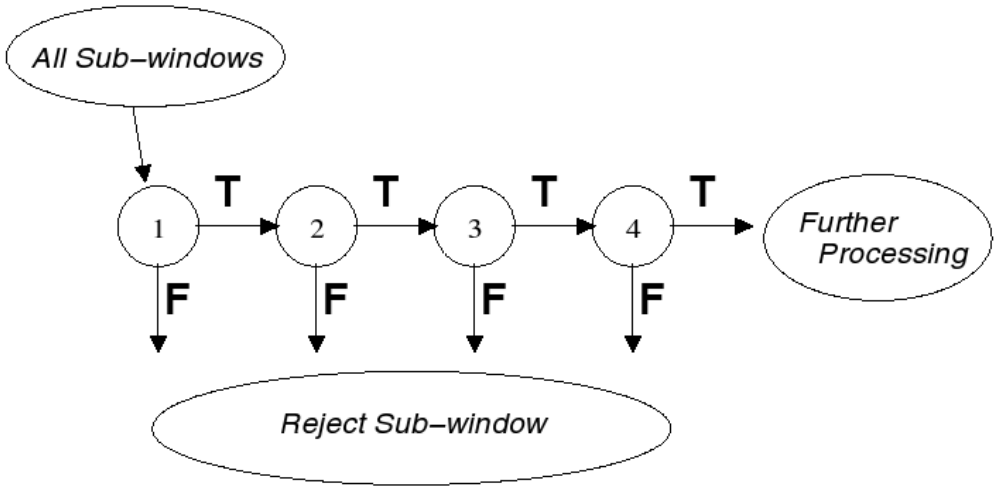
Cascade classifiers [Viola and Jones 2001]



Asymmetric problem

- Coarse-to-Fine approach:
  $\longrightarrow$ Reduce the (negative) hypothesis space incrementatlly.
- Efficient way of organizing object detection
  $\longrightarrow$ there are more non-object than object windows.
- First layers must be very fast

Cascade classifiers [Viola and Jones 2001]



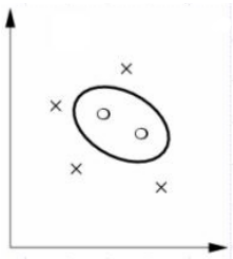
Asymmetric problem

- Layers must have high true positive rates (e.g.):
  $\rightarrow$ if $d_i = 0.999, n = 10, D = \prod_{i=1}^{n} \simeq 0.99$
  and low false positive rates (e.g.):
  $\rightarrow$ if $f_i = 0.2, n = 10, F = \prod_{i=1}^{n} \simeq 10^{-6}$
- First layers are easier to be trained, complexity of the problem increases as we move through the cascade

Adaboost (two-class case)

Classification: Basic concepts

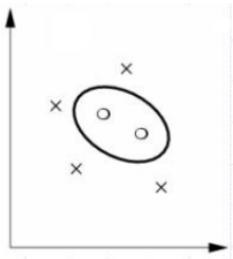
Using a training set $S = \{(x_i, y_i)\}_{i=1,\ldots,m}, y_i \in \{-1, 1\}$
build a classifier $H(x) : \mathbb{R}^n \rightarrow \mathbb{R}$
such that $Prob(sign(H(x)) \neq y)$ is small

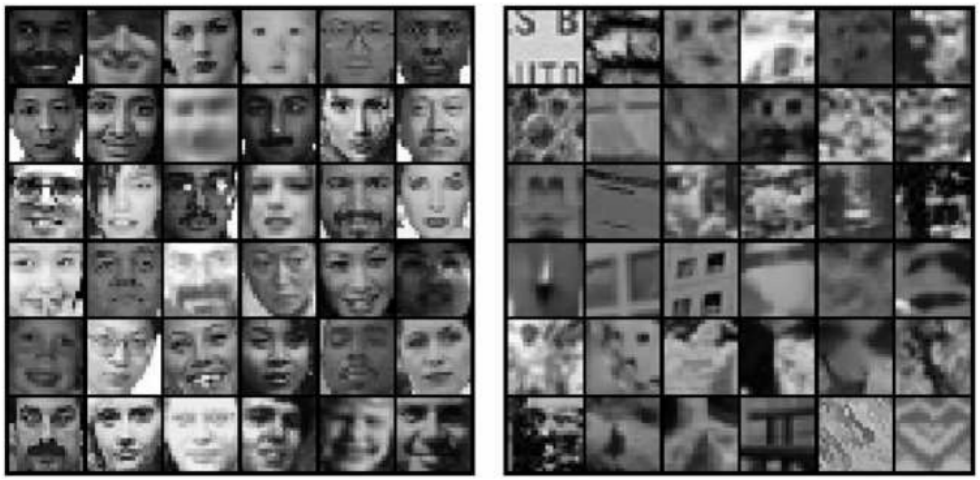


Adaboost (two-class case)

Classification: Basic concepts

Using a training set $S = \{(x_i, y_i)\}_{i=1,\ldots,m}, y_i \in \{-1, 1\}$
build a classifier $H(x) : \mathbb{R}^n \rightarrow \mathbb{R}$
such that $Prob(sign(H(x)) \neq y)$ is small





Faces Non-Faces

[Images from: Ce Liu & Hueng-Yeung Shum, 2003]

Adaboost (two-class problem)

Adaboost: Main ideas

- Build additive model: $H(x) = \sum_{t=0}^{T} h_t(x)$
- Add terms to sum iteratively
- Drive focus to wrongly classified examples
- Basicaly we will minimize $E_{x \sim S}[\exp(-yH(x)]$ iteratively

Note: The classification is correct $\iff$ the margin $yH(x) \geq 0$

Features

Associate a weak classifier to a feature

$$H(x) = \sum_{t=0}^{T} h_t(x) = \sum_{t=0}^{T} h_t(f_t(x))$$

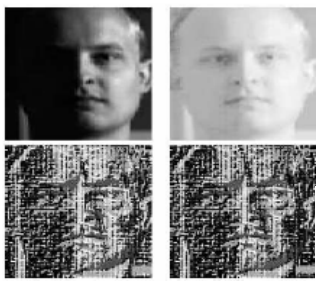
Features

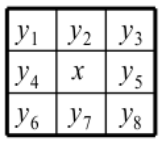
Associate a weak classifier to a feature

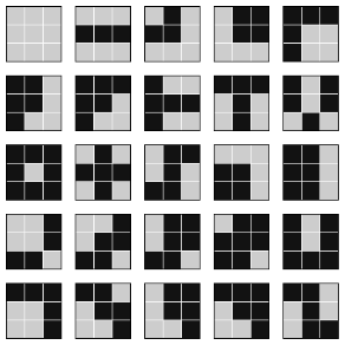
$$H(x) = \sum_{t=0}^{T} h_t(x) = \sum_{t=0}^{T} h_t(f_t(x))$$

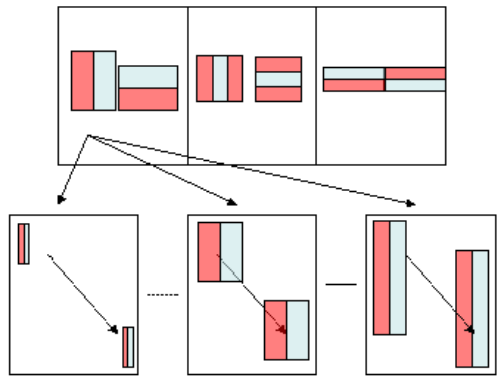
Examples of feature sets

- Modified Local Binary Patterns (mLBP) [Froba et al 2004]
  - Well suited for face detection
  - Invariant to linear constrast changes
- Haar-like wavelets [Viola and Jones 2001]
  - Well suited for frontal face detection,
  - Very fast evaluation using the Integral Image









Adaboost

- Input: Training set $S = \{(x_i, y_i)\}_{i=1,\ldots,m}$
- Initialize Weights: $w_i(0) = 1, i = 1, \ldots, m$
- For $t = 0, \ldots, T$
  - Normalized weigths $\{w_i(t)\}_{i=1,\ldots,m}$ such that they add to one
  - Select $h_t \in \mathbf{H}$ (and $f_t \in \mathbf{F}$), such that

$$Z_t = \sum_{i=1}^{m} w_i(t) \exp\left(-y_i h_t(f_t(x_i))\right)$$

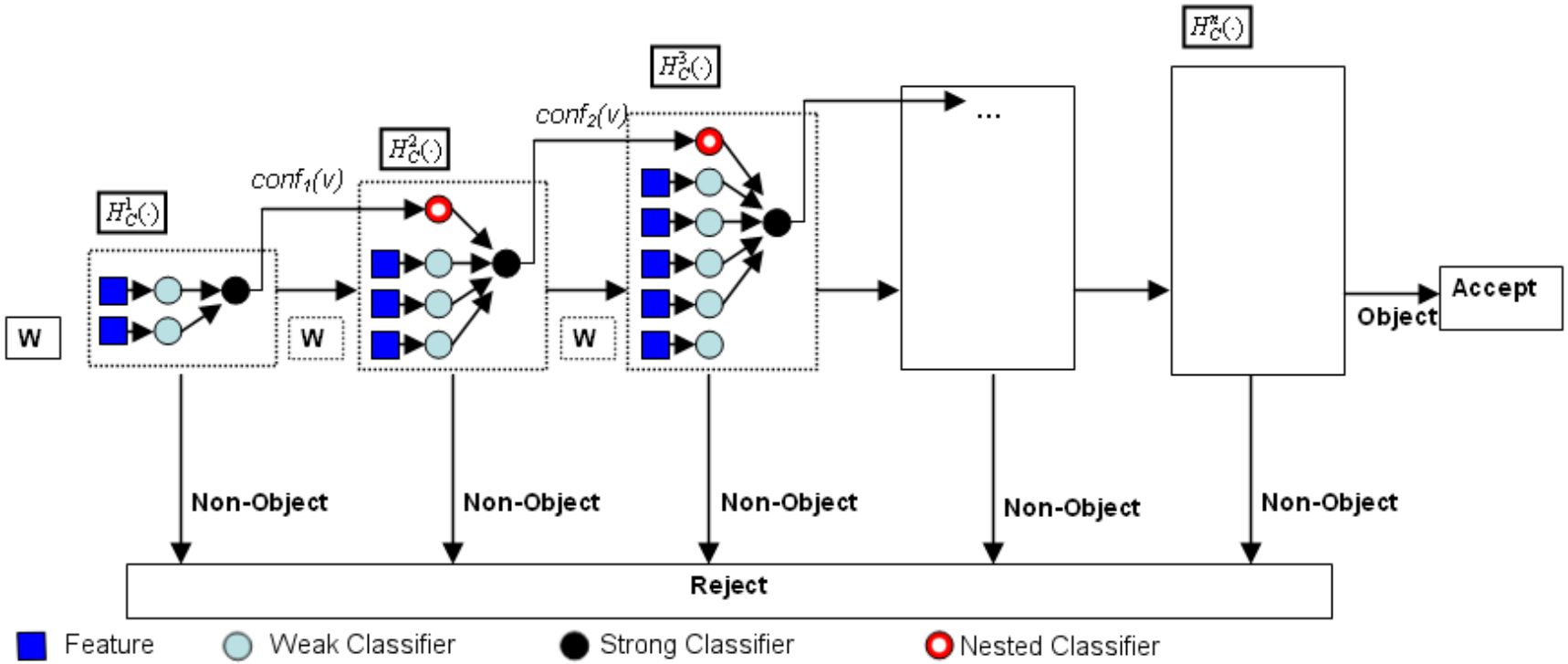
    is minimized
  - Update weigths: $w_i(t+1) = w_i(t) \exp\left(-y_i, h_t(f_t(x_i))\right)$
- Return:

$$H(x) = \sum_{t=0}^{T} h_t(f_t(x))$$

Nested Cascade

- Allows to reuse information
- Faster and more robust than the non-nested case

$$H^k(x) = \sum_{t=0}^{T_k} h_t^k(f_t(x)) + H^{k-1}(x), k > 1$$



1. Introduction
   - Motivation
   - Cascade classifiers
2. Nested Cascade of Boosted Classifier
   - Adaboost
   - Nested Cascade Classifiers
3. Multiclass Object Detection
   - A multiclass extension of Adaboost
   - Weak classifier
   - Training procedures
4. Results, and future work
   - Experiment set up
   - Example results and Comments
   - Future work

Multiclass Object Detection: Related work

- Multiple cascades [Wu & al 2004, Schneiderman 2004]
  $\longrightarrow$ Frontal faces, rotated faces, cars, traffic signs, etc.
- View estimation followed by multiple cascades [Jones & Viola 2003]
  $\longrightarrow$ Frontal and rotated faces.
- Tree of boosted cascades (Floatboost) [Li et al 2004 ; Ong & Bowden 2004]
  $\longrightarrow$ Faces, hands.
- Coarse-to-fine [Fleuret & Geman 2001; Amit et al. 2004; Gangaputra & Geman 2006]
  $\longrightarrow$ Faces, Numbers and Letters.
- Sharing of weak classifiers among classes (Jointboost) [Torralba et al. 2004]
  $\longrightarrow$ frontal faces, cars, persons, keyboards, telephones, etc.
- Vectorized Adaboost and tree structure (Vector Boosting) [Huang et al 2007]
  $\longrightarrow$ Faces at multiple rotations.

Extending the classifier to the multiclass case

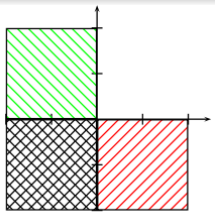
Desired requirements/properties

- The classifier should scale well with the number of classes
  Accuracy, robustness, processing time
- The training time should scale well with the number of classes.
- Good results should be obtained using small training sets.
- Fast evaluation
  - Reuse of feature and classifier evaluations

Multiclass formulation

Multiclass

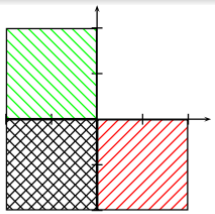
$$\vec{H}(x) = \sum_{t=0}^{T} \vec{h}_t(\vec{f}_t(x))$$



Multiclass formulation

Multiclass

$$\vec{H}(x) = \sum_{t=0}^{T} \vec{h}_t(\vec{f}_t(x))$$



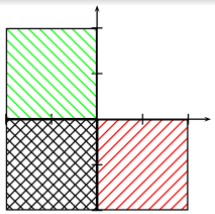
Feature sharing

$$\vec{H}(x) = \sum_{t=0}^{T} \vec{h}_t(f_t(x))$$

Multiclass formulation

Multiclass

$$\vec{H}(x) = \sum_{t=0}^{T} \vec{h}_t(\vec{f}_t(x))$$



Feature sharing

$$\vec{H}(x) = \sum_{t=0}^{T} \vec{h}_t(f_t(x))$$

Notation

$$H(x, c) = \sum_{t=0}^{T} h_t(f_t(x), c), c = 1, \ldots, M$$

Multiclass formulation

One case

- Training example: (x_i, y_i), $y_i \in \{-1, 1\}$ represents the target halfspace
- $yH(x) \geq 0$ if $H(x)$ is in the correct half space

Multiclass formulation

One case

- Training example: (x_i, y_i), $y_i \in \{-1, 1\}$ represents the target halfspace
- $yH(x) \geq 0$ if $H(x)$ is in the correct half space

Multiclass case

- Training example: $(x_i, \mathbf{V}_i)$, The vector set $\mathbf{V}_i$ represents the target sub space defined as the intesection of halfspaces
- An object instance x_i (belonging to class C_m) is classified correctly
  - iff $\forall \vec{V} \in \mathbf{V}_m, \vec{V} \cdot \vec{H}(x) \geq 0$ (Vector Boosting, [Huang et al 2007])

Multiclass formulation

One case

- Training example: (x_i, y_i), $y_i \in \{-1, 1\}$ represents the target halfspace
- $yH(x) \geq 0$ if $H(x)$ is in the correct half space

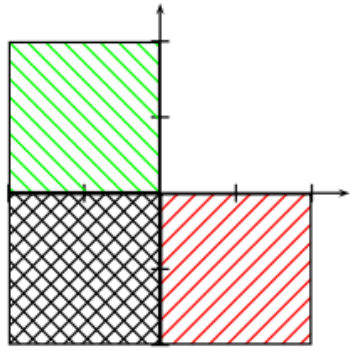
Multiclass case

- Training example: $(x_i, \mathbf{V}_i)$, The vector set $\mathbf{V}_i$ represents the target sub space defined as the intesection of halfspaces
- An object instance x_i (belonging to class C_m) is classified correctly
  - iff $\forall \vec{V} \in \mathbf{V}_m, \vec{V} \cdot \vec{H}(x) \geq 0$ (Vector Boosting, [Huang et al 2007])

(iff $\forall \vec{V} \in \mathbf{V}_m, Q(\vec{V}, \vec{H}(x)) \geq 0$) (Generalized Multiclass Adaboost)

Objective space

Example 1: 2 objects classes and a non-object class

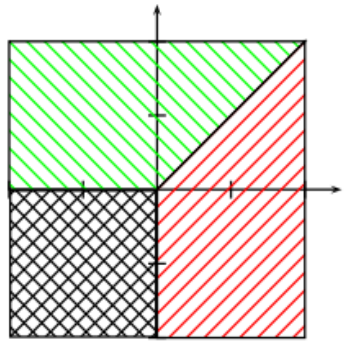


Disjoint and non overlapping

Class	Vectors	Objective region	Region on figure
Object 1	$(+1, 0); (0, -1)$	$x > 0, y < 0$	Red
Object 2	$(-1, 0); (0, +1)$	$x < 0, y > 0$	Green
Non Object	$(-1, 0); (0, -1)$	$x < 0, y < 0$	Dashed Grey

Objective space

Example 2: 2 objects classes and a non-object class

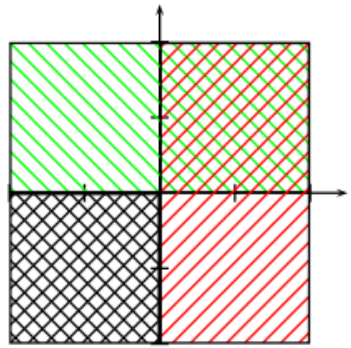


Joint boundary

Class	Vectors	Objective region	Region on figure
Object 1	$(+1,0);(+1,-1)$	$x>0, x>y$	Red
Object 2	$(-1,0);(-1,+1)$	$y>0, y>x$	Green
Non Object	$(-1,0);(0,-1)$	$x<0, y<0$	Dashed Grey

Objective space

Example 3: 2 objects classes and a non-object class



Ovelapping

Class	Vectors	Objective region	Region on figure
Object 1	$(1,0)$;	$x > 0$	Red
Object 2	$(0,1)$;	$y > 0$	Green
Non Object	$(-1,0);(0,-1)$	$x < 0, y < 0$	Dashed Grey

Generalized Adaboost

- Training set $S = \{(x_i, \mathbf{V}_i)\}_{i=1,\ldots,m}$
- $\mathbf{V}_i = \{\vec{V}_i^j\}_j$ is a vector/parameter set representing the objective region associated to the example x_i
- Initialize Weights: $w_{i,j}(0) = 1$
- For $t = 0, \ldots, T$
  1. Normalized weigths $\{w_{i,j}(t)\}_{i,j}$ so that they add to one
  2. Select $\vec{h}_t$ and f_t, such that Z_t is minimized, with

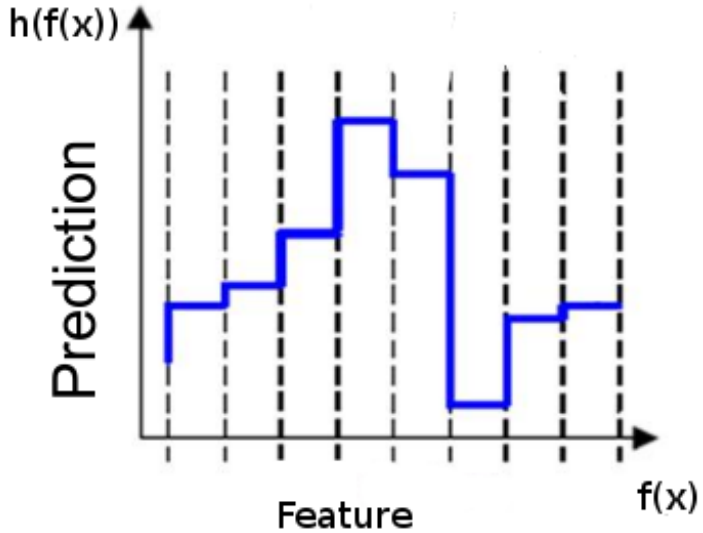
$$Z_t = \sum_{i=1}^{m} \sum_{j:V_i^j \in \mathbf{V}_i} w_{i,j}(t) \exp\left(-Q\left(\vec{V}_i^j, \vec{h}_t(f_t(x_i))\right)\right)$$

  3. Update weigths: $w_{i,j}(t+1) = w_{i,j}(t) \exp\left(-Q\left(\vec{V}_i^j, \vec{h}_t(f_t(x_i))\right)\right)$
- return $\vec{H}(x) = \sum_t \vec{h}_t(f_t(x))$

Domain Partitioning (one-class case)

Domain Partitioning weak classifiers [Shapire and Singer 1999]

- Fast evaluation thanks to the use of look up tables: $O(1)$.
- Well fitted for different types of features (e.g, rectangular features and mLBP).



$$H(x) = \sum_{t=0}^{T} h_t(f_t(x))$$

Posible outputs of the weak classifier $\vec{h}_k$

- Independent classifiers [Huang et al 2007]

$$H(x, c) = \sum_{t=1}^{T} h_t(f_t(x), c)$$

- Joint classifiers (Similar to [Torralba et al 2007])

$$H(x, c) = \sum_{t=1}^{T} \beta_t^c h_t(f_t(x)), \beta_t^c \in \{0, 1\}$$

- Coupled classifiers (Proposed)

$$H(x, c) = \sum_{t=1}^{T} \gamma_t^c h_t(f_t(x)), \gamma_t^c \in \mathbb{R}$$

Posible outputs of the weak classifier $\vec{h}_k$

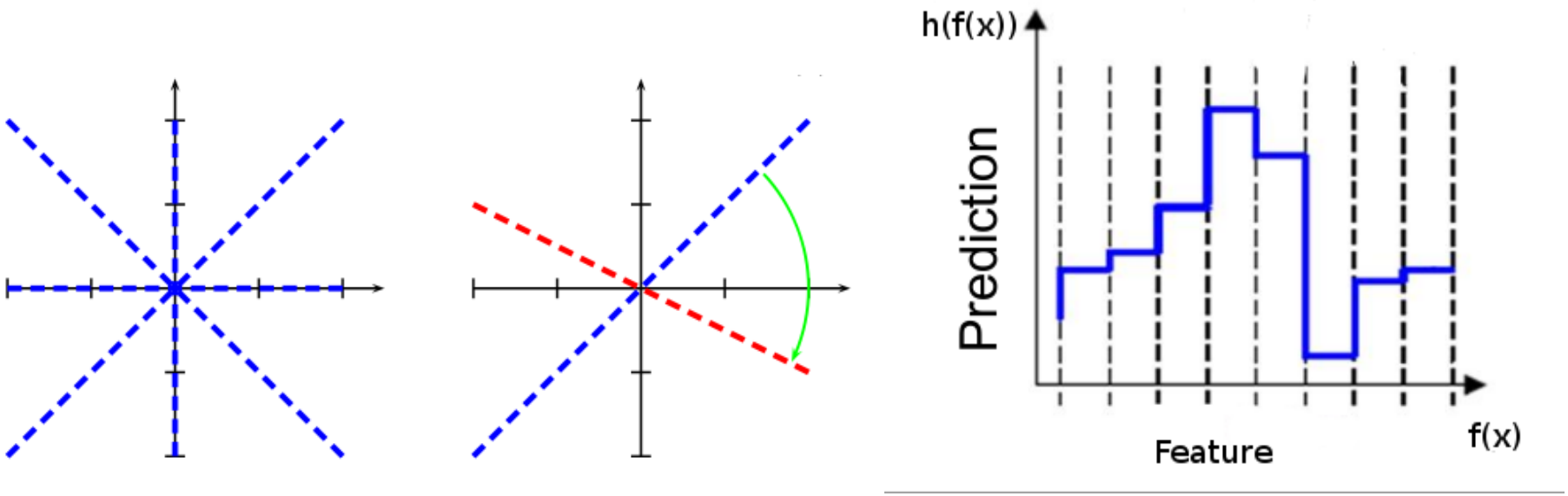


Figure: Left: Joint, Center: Coupled (Proposed), Right: domain partitioning

Optimization problem

Method	Problem	Solution	N. of Prob.	N. of Variables	Order $O()$
Adaboost	$\min_{c_j} \sum_j w_+^j e^{-c_j} + w_-^j e^{+c_j}$	Analitic	J	J	$N + J$
Joint	$\min_{c_j, b_m \in \{0,1\}} \sum_{j,m} w_+^{j,m} e^{-\beta_m c_j} + w_-^{j,m} e^{\beta_m c_j}$	Analytic	2^M	$J(+M)$	$N + J2^M$
Independent	$\min_{c_{j,m}} \sum_m \sum_j w_+^{j,m} e^{-c_{j,m}} + w_-^{j,m} e^{c_{j,m}}$	Analytic	J	JM	$N + JM$
Coupled	$\min_{c_j, \gamma_m \in \mathbb{R}} \sum_{j,m} w_+^{j,m} e^{-\gamma_m c_j} + w_-^{j,m} e^{\gamma_m c_j}$	Newton *	1	$J + M$	$N + (J + M)^2$

Weak classifier: domain partitioning (J bins); M classes; N training examples

Train Nested Cascade

Input: Positive Training Set $S = \{(x_i, \vec{V}_i)\}_{i=1,\ldots,m}$
Input: Bootstrapp Set B
Input: Min true positive rate per node D
Input: Max false positive rate per node F and max false positive of the cascade F_C

1: $k = 1$, $F_k = 1$, $H \leftarrow 0$, $S_{B_0} = B$
2: **while** $F_k > F_C$ **do**
3: $S_{B_k} = Bootstrapp(S_{B_{k-1}}, \vec{H}, M)$
4: $[\vec{H}_k] = TrainLayer(S \otimes S_{B_k}, \vec{H}, F, D)$
5: $\vec{H} \leftarrow \vec{H} \cup \vec{H}_k$
6: $F_k \leftarrow$ Evaluate current cascade $\vec{H}$
7: $k \leftarrow k + 1$
8: **end while**

Output: Trained cascade $\vec{H}$

$Bootstrapp(B, H, M)$

Input: Boostrapp Set B
Input: Classifier $\vec{H}$
1: $S = \emptyset$, $i = 0$
2: **while** $i < M$ **do**
3: $x \leftarrow$ sample S_B
4: **if** $H(x) > 0$ **then**
5: $S \leftarrow S \cup x$
6: $i \leftarrow i + 1$
7: **end if**
8: **end while**
Output: S

Train Layer

Input: Training Set $S = \{(x_i, \vec{e}_i)\}_{i=1,\ldots,m}$

Input: Already trained cascade $\vec{H}$

1: Init: $t \leftarrow 0, w_0(i) \leftarrow \exp(-\vec{e}_i \cdot \vec{H}(x_i)), F_0 \leftarrow 1, \vec{H}_0 \leftarrow \vec{H}_0, m_0 \leftarrow m$

2: **while** $F_t > F$ **do**

3: Normalize the weigths $\{w_t(i)\}_{i=1,\ldots,m_t}$ s.t they add to one

4: **Select** $\vec{h}_t(\cdot)$ s.t $Z_t = \sum_i w_t(i) \exp\left(-\vec{e}_i \cdot \vec{h}_t(x_i)\right)$ is minimized

5: Set $\vec{H}_t \leftarrow \vec{H}_{t-1} + \vec{h}_t$

6: Update weigths: $w_{t+1}(i) = w_t(i) \exp(-\vec{e}_i \cdot \vec{h}_t(f_t(x_i)))$

7: Set threshold for $\vec{H}_t$ such that $D_t > D$ and evaluate current false positive rate F_t

8: $t \leftarrow t + 1$

9: **end while**

Output: $\vec{H}(\cdot) = \sum_{t=1}^{T} \vec{h}_t(\cdot), \vec{h}_t(\cdot) = \vec{\beta}_t h_t(f_t(\cdot))$

1. Introduction
   - Motivation
   - Cascade classifiers
2. Nested Cascade of Boosted Classifier
   - Adaboost
   - Nested Cascade Classifiers
3. Multiclass Object Detection
   - A multiclass extension of Adaboost
   - Weak classifier
   - Training procedures
4. Results, and future work
   - Experiment set up
   - Example results and Comments
   - Future work

Experiments

In plane Rotated Faces

- Nested Cascade
- Rectangular and mLBP features
- Vectorized weak classifiers trained with :
  - Independent classifiers: $H(x, c) = \sum_{t=1}^{T} h_t(f_t(x), c)$
  - Joint classifiers: $H(x, c) = \sum_{t=1}^{T} \beta_t^c h_t(f_t(x)), \beta_{c,t} \in \{0, 1\}$
  - Coupled classifiers: $H(x, c) = \sum_{t=1}^{T} \gamma_t^c h_t(f_t(x)), \gamma_{c,t} \in \mathbb{R}$

4-classes case

- Number of classes (n)
  - 1 class: Frontal faces without in plane rotation
  - 2 classes: 0 and 180 degrees of in plane rotation
  - 4 classes: 0, 90, 180 and 270 degrees of in plane rotation
  - 8 classes: 0, 45, 90, 135, 180, 225, 270 and 315 degrees
- Number of training examples. Positives: $n * 5000$, Negative: $n * 5000$

1-class

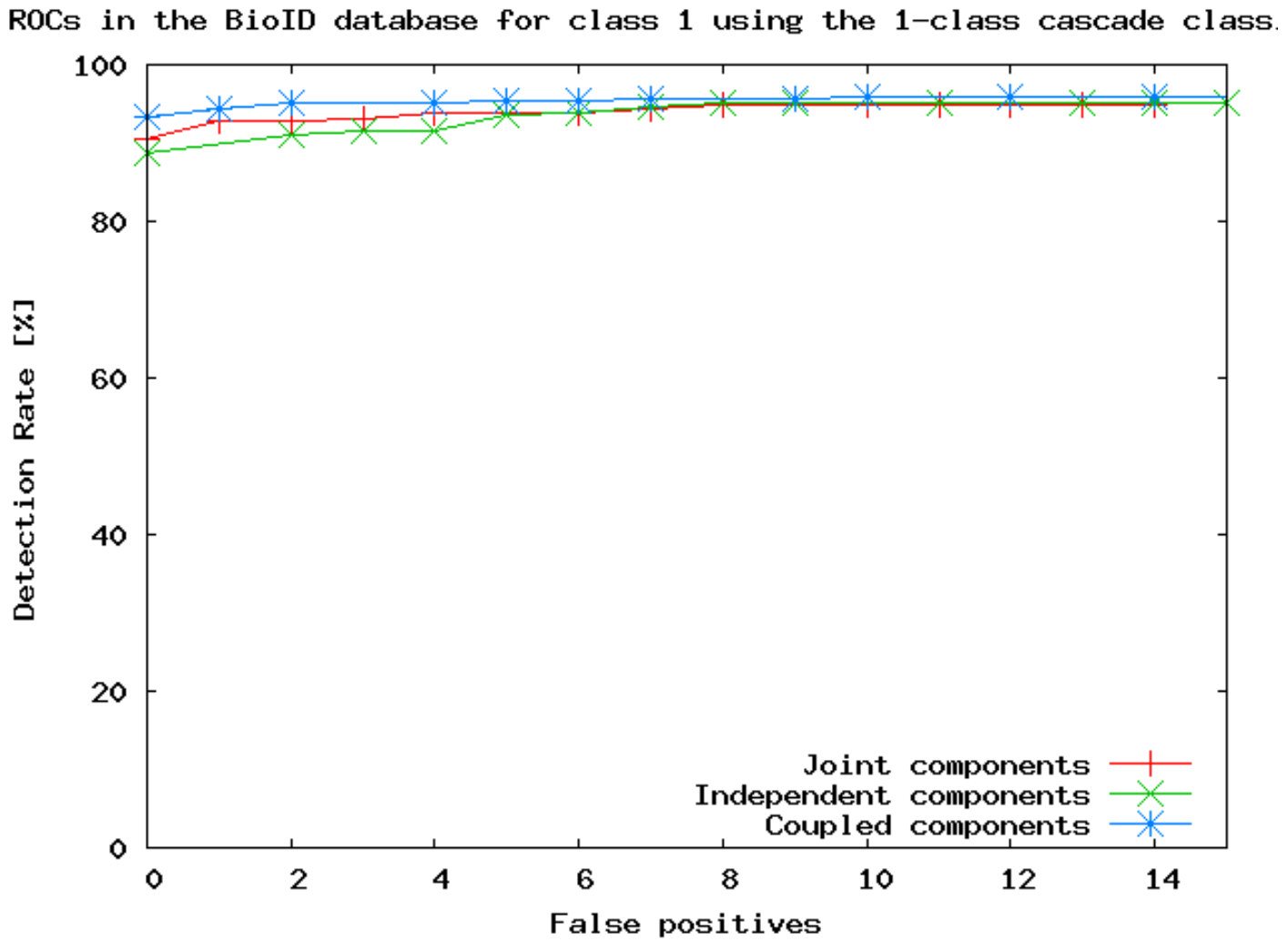


Figure: ROCs of the 1-class cascade used for detecting faces of class 1

2-classes

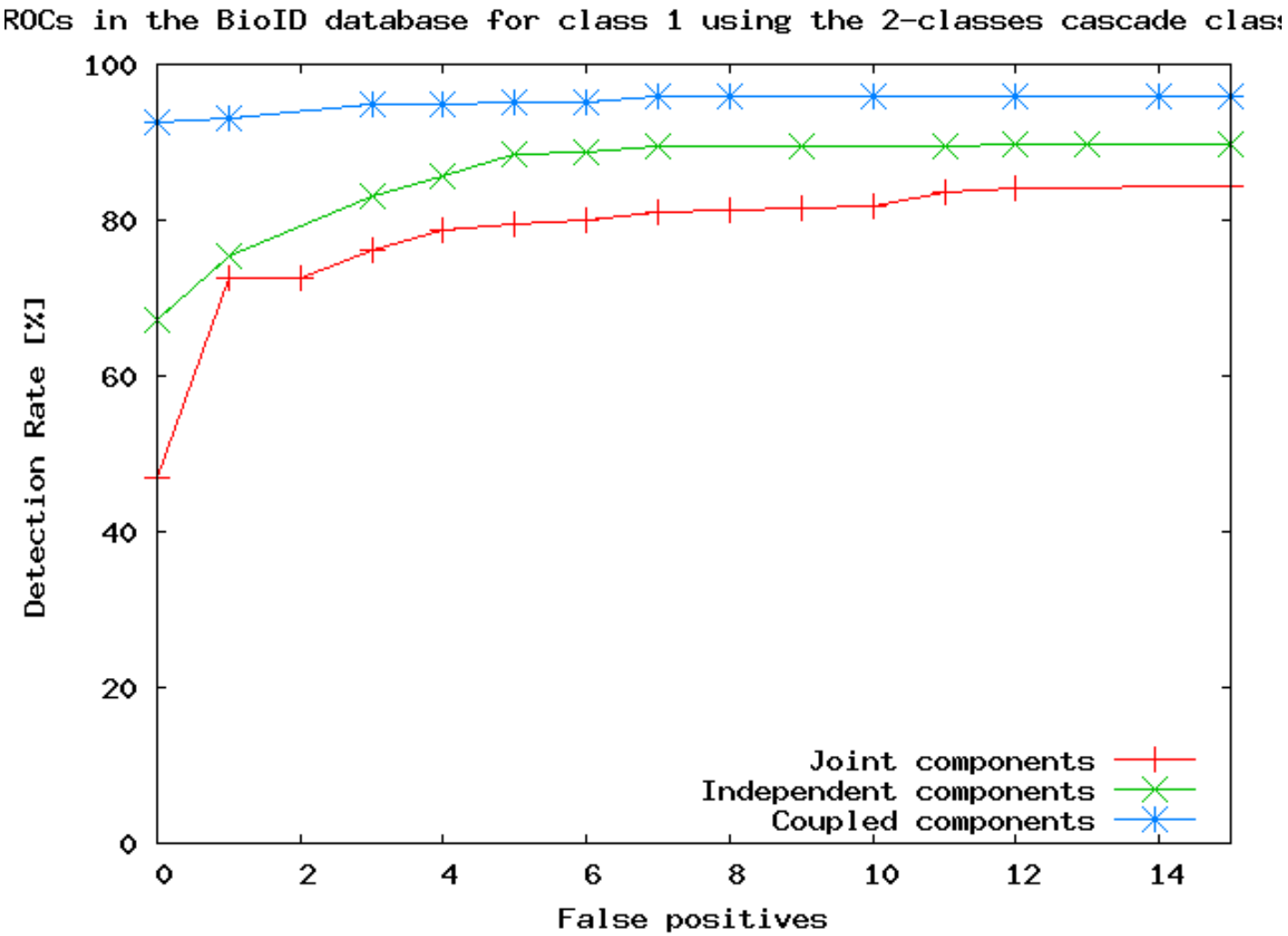


Figure: ROCs of the 2-classes cascade used for detecting faces of class 1

4-classes

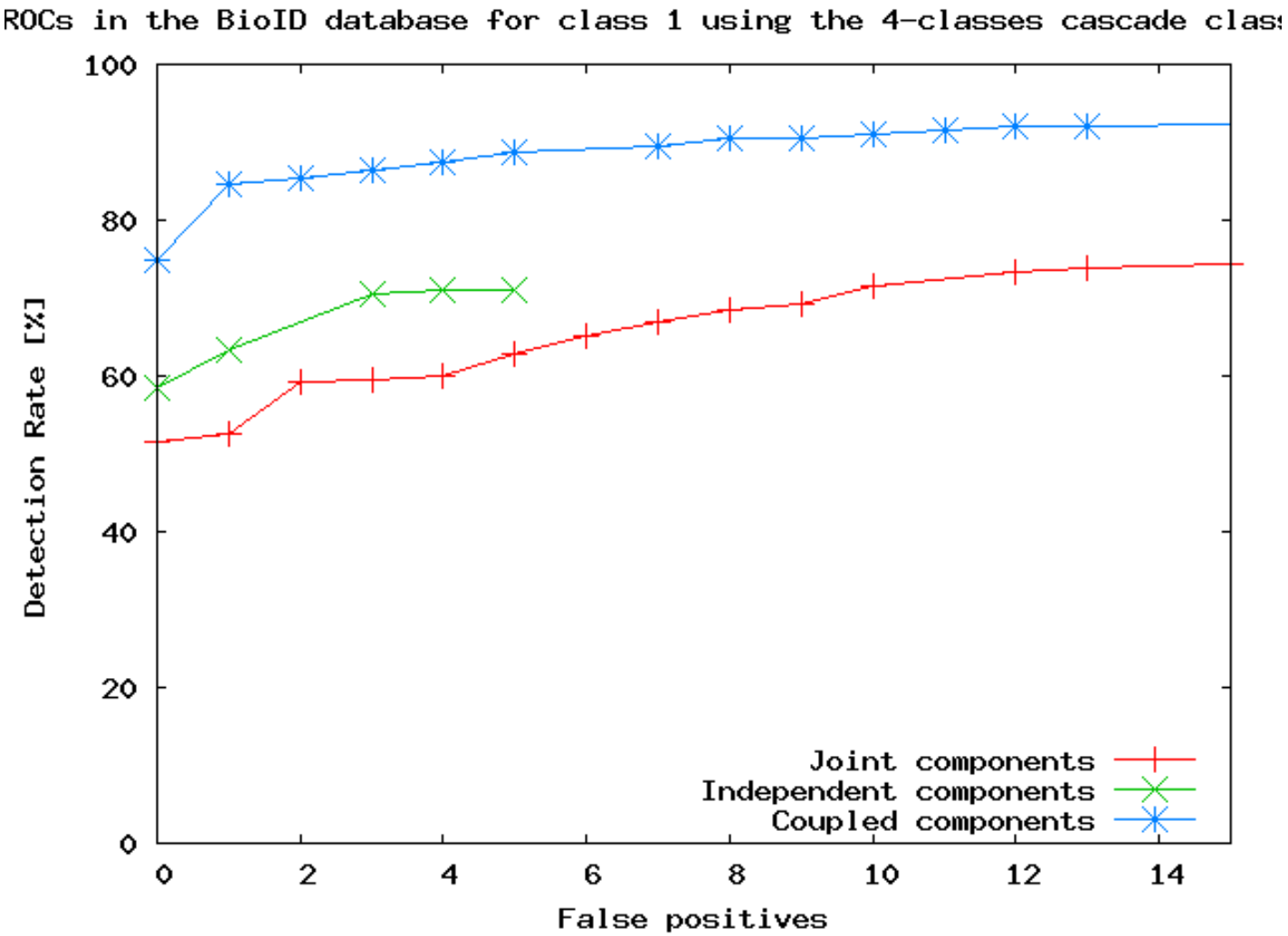


Figure: ROCs of the 4-classes cascade used for detecting faces of class 1

8-classes

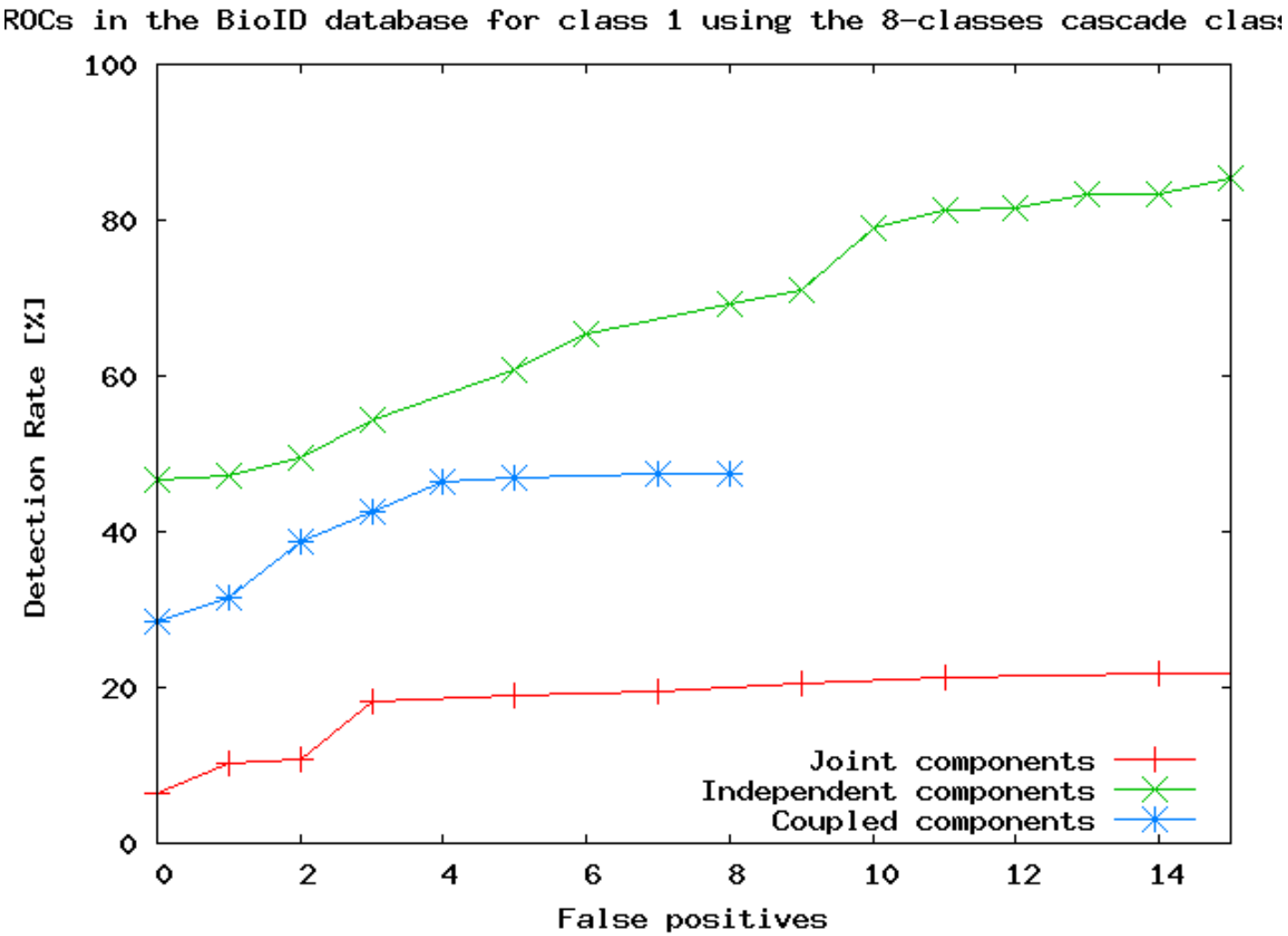


Figure: ROCs of the 8-classes cascade used for detecting faces of class 1

Comparision

Table: Detection rate [%] for 5 false positives.

Weak classifier's training method	Number of classes			
	1	2	4	8
Independent	93.56	88.36	71.07	**60.88**
Jointly trained	93.82	79.55	62.72	18.94
Coupled	**95.27**	**95.07**	**88.63**	46.94

Processing time

Table: Average processing time [sec] for a 384x286 pixels image

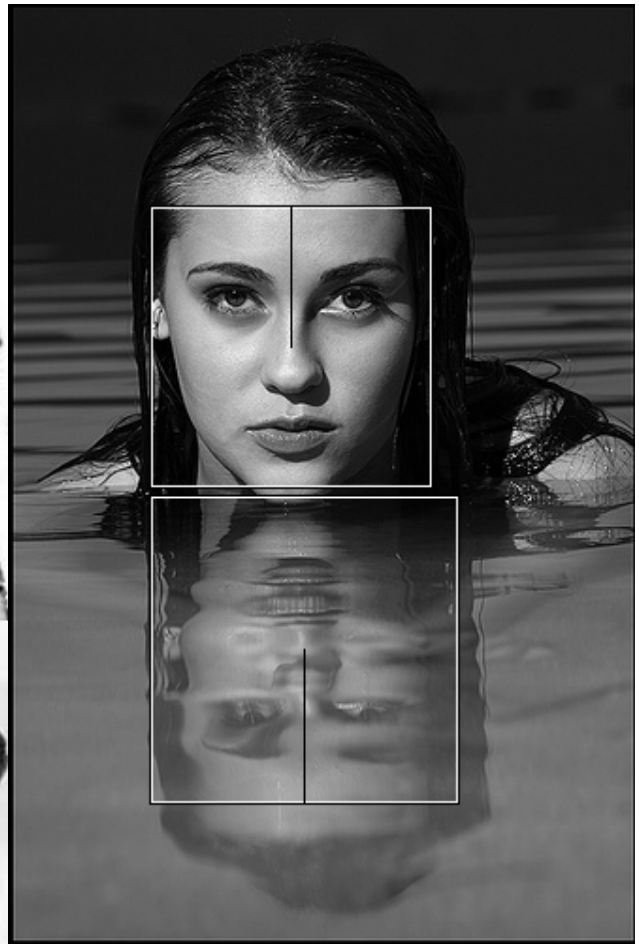
Weak classifier's training method	Number of classes			
	1	2	4	8
Independent	0.33	0.38	0.53	6.39
Jointly trained	0.33	0.52	1.33	0.95
Coupled	0.31	0.95	3.49	5.09

Table: Number of weak classifiers at the first layer of the cascade

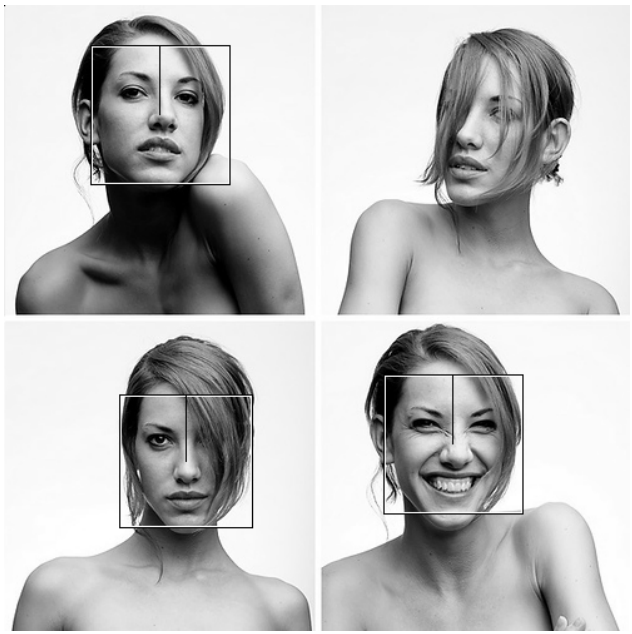
Weak classifier training's method	Number of classes			
	1	2	4	8
Independent	12	10	8	28
Jointly trained	12	9	7	9
Coupled	9	11	17	24

1-class case

4-classes case

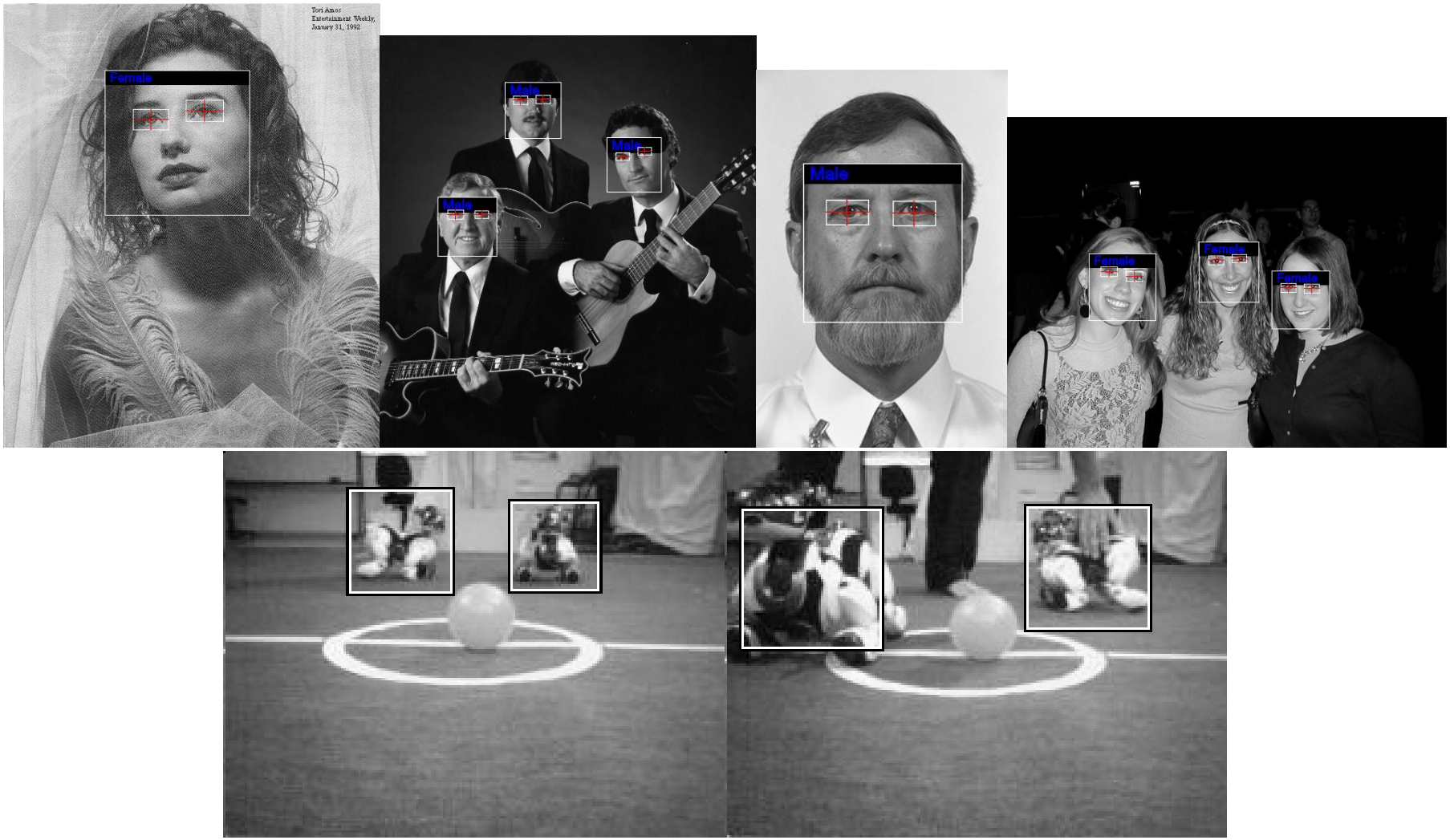


4-classes case



4-classes case

1-class case


Tori Amos
Entertainment Weekly,
January 31, 1992
Female
Male
Male
Male
Male
Female
Female
Female

Results

Training time

1-class

- Independent: 2.6 hours
- Joint: 2.6 hours
- Coupled: 2.6 hours

8-classes

- Independent: 37 hours
- Joint: 72.6 hours
- Coupled: 34.3 hours

Summary

Main proposed ideas

- Multiclass nested cascade
- Generalized version of Vector Boosting
- Use of coupled components
- Multiclass extention of bootstrapping

Conclusions

Features

- Using the right features is essential, in particular if the system will share features and classifier parameters

Nested Cascade

- Number of weak classifiers in the first layer is almost the same for the three variants
- Number of weak classifiers in the first layer grows logaritmically with the number of classes
- Number of layers in the cascade does not grows with the number of classes

Conclusions

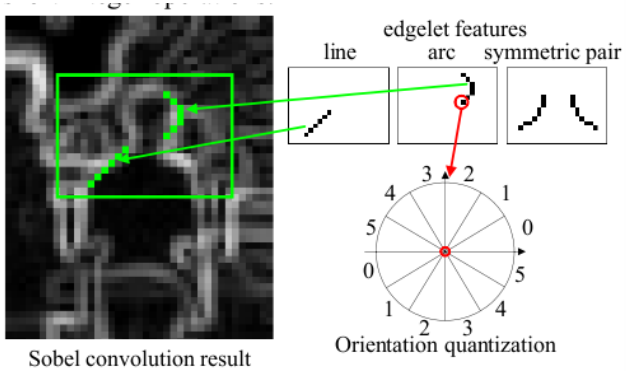
Vectorized weak classifiers

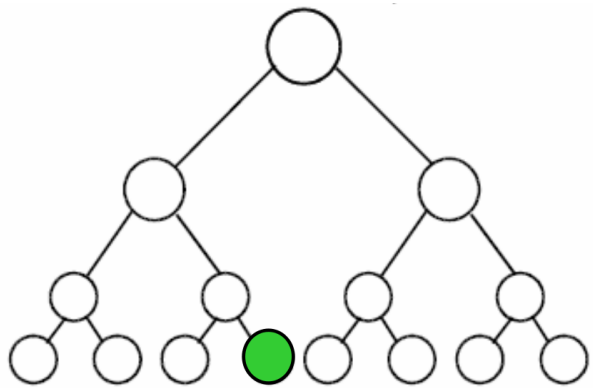
- Independent components:
  - Reasonable results overall, but only the best in the case of 8-classes.
  - It works well even when the classes are very "different".
  - It requires large size training set to aviod overfitting (preliminary results were not presented here)
- Joint components:
  - Many times objects are detected but classes are confused.
  - Does not scales well (Training is too slow)
  - More weak classifiers are selected in the later layers of the cascade (slow).
- Coupled components:
  - Best results for 1,2 and 4 classes.
  - They work quite well when the classes are "similar".
  - Scalable training (fast training).

Future Work

Some options of new features to implement

- Edged based
- Rotated Rectangular features:
- Gaussian features and transformations of it: rotation, bending, etc
- "Correlation" with parts of images
- SIFT
- ERCF





Future work

- Use different training algorithms for training the multiclass weak classifier at different layers of the cascade

Future work

Build a tree by split the nodes of the cascade as needed

Questions:

- When to split a node?
- Split at every node of the cascade?
- How to split the node? (how to group the classes)?

- Which feature to use at each layer?
- Which criteria to use in the selection of the features?
  - Trade off between processing speed/ accuracy and training time

[Geman et al.]

Future work

Speeding up the detection

- Sequential testing tecniques to determine the optimal thresholds.
- Adjacent windows can be skipped depending on the output of the current window.

[Werman et al.]

Thanks for your attention

Any questions?