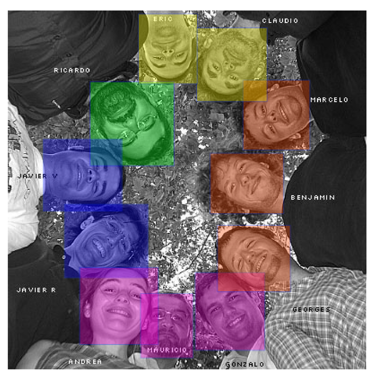
Multiclass object detection using multiclass coupled classifiers and nested classifiers

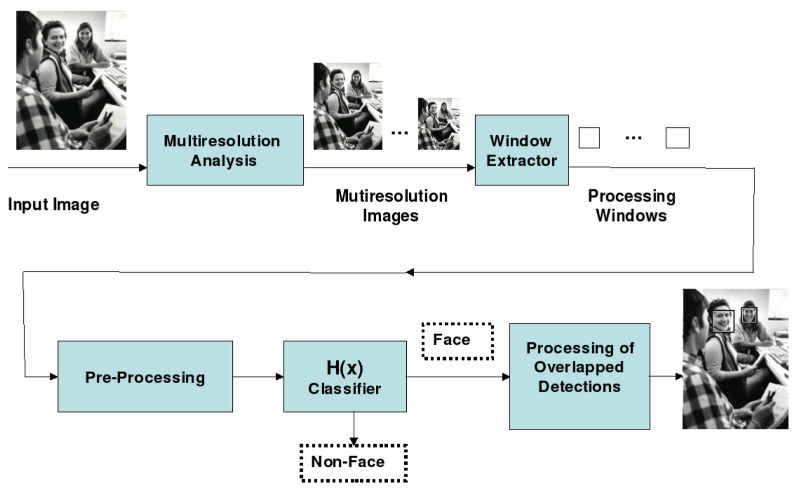
Rodrigo Verschae and Javier Ruiz-del-Solar
Department of Electrical Engineering, Universidad de Chile
CIARP, 2008

Introduction

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.
Detection Architecture: Exhaustive search over positions and scales



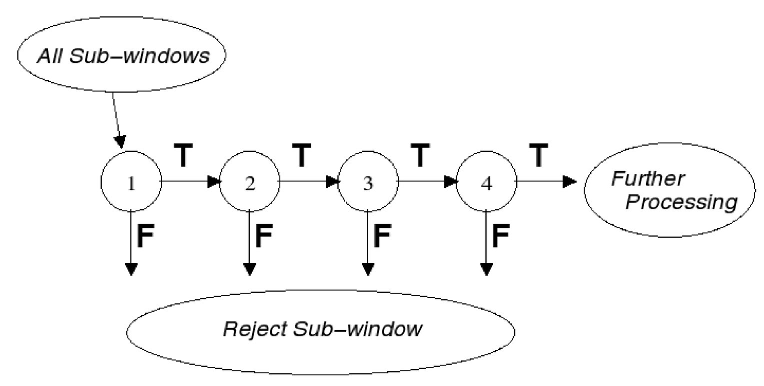


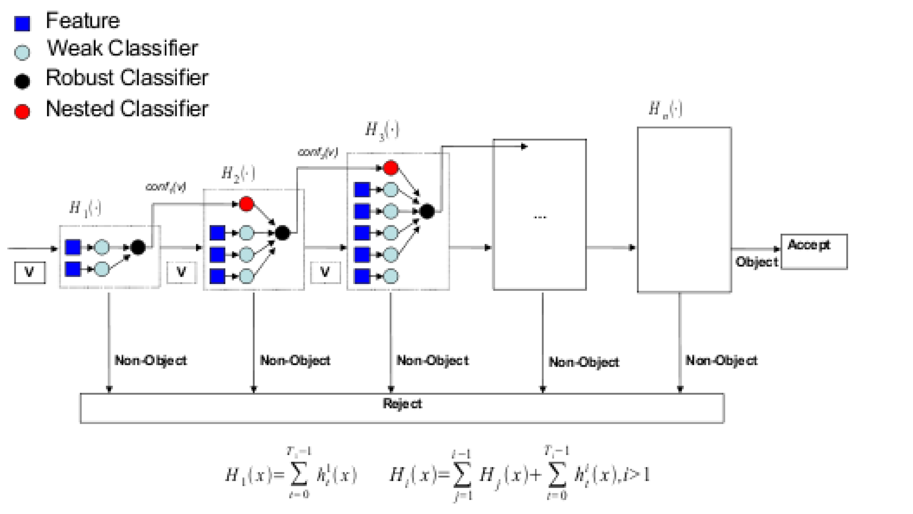
Adaboost [Schapire & Singer 1999]

- Build additive model: $H(x) = \sum_{t=0}^{T} h_t(x)$
- Add terms to the sum iteratively, greedy minimization of $E_{x\sim S}[\exp(-yH(x)]$.
- Drive focus to wrongly classified examples

Nested Cascade classifiers [Viola & Jones 2001], [Wu et al. 2004]

- Efficient way of organizing object detection
  - Asymmetric problem: there are more non-object than object windows.
- Layers must have high true positive rates and low false positive rates, e.g.: if $n = 10$, $f_i = 0.2$, $d_i = 0.999$, then $F = \prod_{i=1}^{n} f_i \simeq 10^{-6}$, $D = \prod_{i=1}^{n} d_i \simeq 0.99$
- Nested cascade
  - Allows to reuse information
  - Faster and more robust than the non-nested case





Proposed method

Main Idea: Coupled components

- Fast training
- Few parameters need to be estimated
- Reuse of feature and classifier evaluations

Multiclass formulation

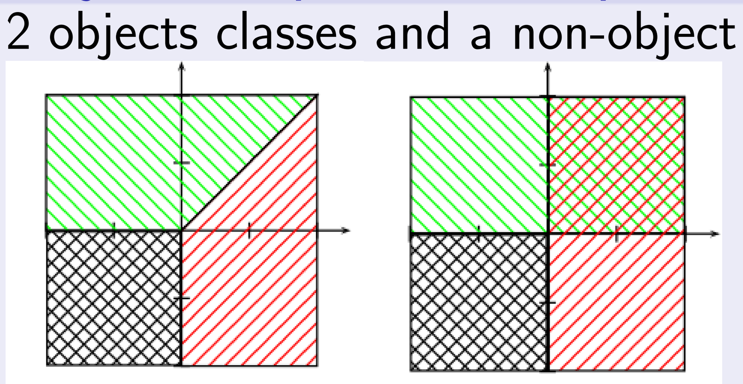
- One class case:
  - Training example: (x_i, y_i), $y_i \in \{-1, 1\}$
  - y_i represents the target halfspace:
    - $H(x)$ is in the correct half space $\iff yH(x) \geq 0$
- Multiclass case (Vector Boosting [Huang et al 2007, PAMI])
  - Training example: $(x_i, \mathbf{V}_i)$, $\mathbf{V}_i \in \Re$.
  - $\mathbf{V}_i$ represents the target sub space defined as the intesection of halfspaces:
    - $\vec{H}(x)$ is in the correct target region $\iff \forall \vec{V}_j \in \mathbf{V}_m, \vec{V}_j \cdot \vec{H}(x) \geq 0$

Generalized Adaboost

- Training set $S = \{(x_i, \mathbf{V}_i)\}_{i=1,\dots,m}$, $\mathbf{V}_i = \{\vec{V}_i^j\}_j$
- Initialize Weights: $w_{i,j}(0) = 1$
- For $t = 0, \dots, T$
  - Normalized weigths $\{w_{i,j}(t)\}_{i,j}$ so that they add to one
  - Select $\vec{h}_t$ and f_t, such that Z_t is minimized, with $Z_t = \sum_{i=1}^{m} \sum_{j:V_i^j \in \mathbf{V}_i} w_{i,j}(t) \exp\left(-Q\left(\vec{V}_i^j, \vec{h}_t(f_t(x_i))\right)\right)$
  - 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 $\vec{H}(x) = \sum_t \vec{h}_t(f_t(x))$

Objective space: Examples

2 objects classes and a non-object class



Example 1 (left): 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

Example 2 (right): Ovelapped target space

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

Multiclass Formulation

Vectorized classifier and feature sharing:

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

Notation:

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

Structure of the multiclass 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 [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)

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

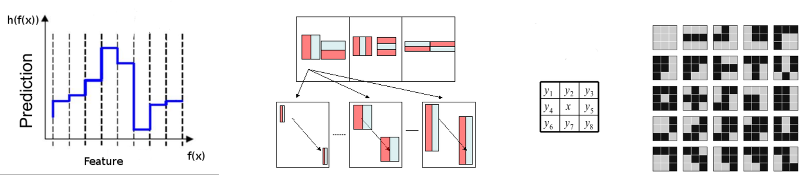
Features and Weak classifiers

Weak Classifiers:

- 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

Features:

- Haar-like and mLBP features



Optimization problem

Method	Problem	Solution	N. of Problems	Vars per Problem	Order $O()$
Adaboost	$\min_{c_j} \sum_j w_+^j e^{-c_j} + w_-^j e^{+c_j}$	Analitic	J	1	$N + J$
Joint	$\min_{c_j, b_m \in \{0,1\}} \sum_{j,m} w_+^{j,m} e^{-b_m c_j} + w_-^{j,m} e^{b_m c_j}$	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	M	$N + JM$
Coupled	$\min_{c_j, b_m \in \mathbb{R}} \sum_{j,m} w_+^{j,m} e^{-\gamma_m c_j} + w_-^{j,m} e^{\gamma_m c_j}$	Newton *	1	$J + M$	$N + (J + M)^2$

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

Results and conclusions

Experiment set up

- In plane Rotated Faces
- Nested Cascade
- Rectangular and mLBP features
- Vectorized weak classifiers trained with :
  - Independent, joint and coupled classifiers
- Number of classes (n)
  - 1 class: Frontal faces without in-plane rotation
  - 2 classes: 0 and 180 in-plane rotation
  - 4 classes: 0, 90, 180 and 270 in-plane rotation
  - 8 classes: 0, 45, 90, 135, 180, 225, 270 and 315
- Number of training examples:
  - Positives: $n * 5000$; Negative: $n * 5000$

Method Comparision

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

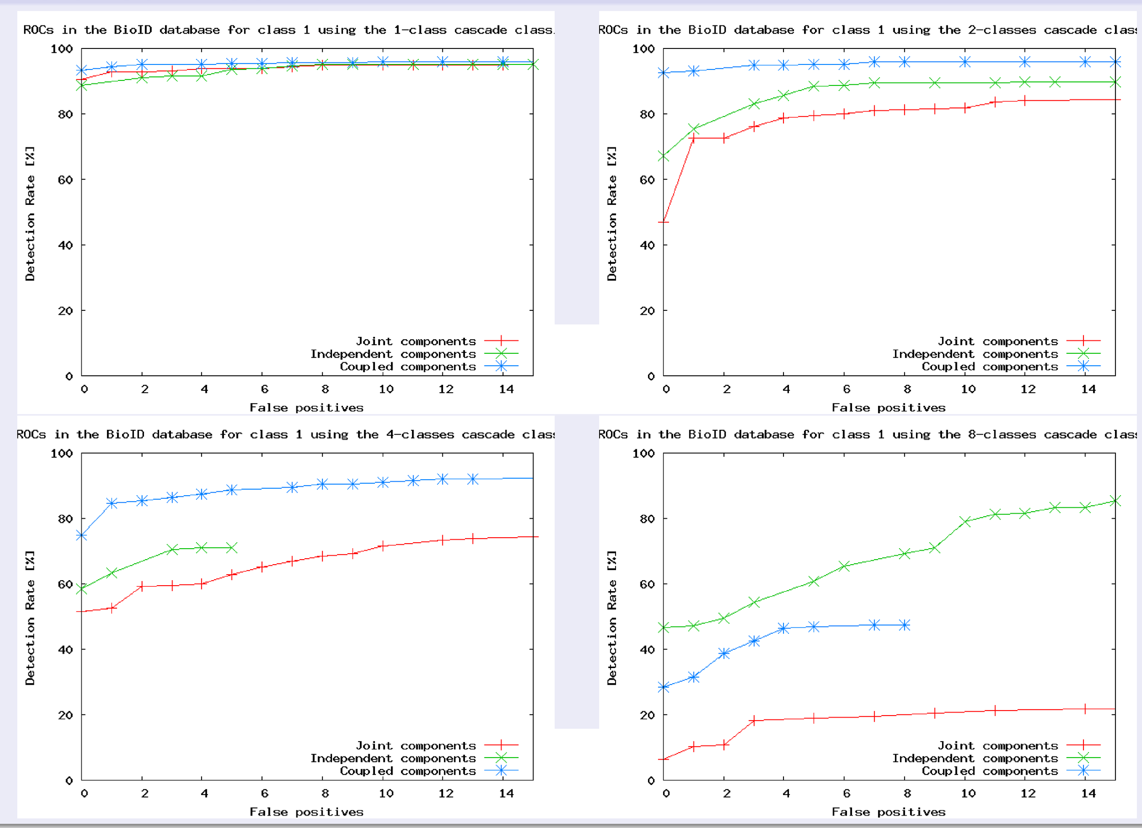
Weak classifier's training method	Number of classes 1	2	4	8
Independent	93.56	88.36	71.07	**60.88**
Jointly trained	93.82	79.55	62.72	18.94
Coupled	**95.27**	**95.07**	**88.63**	46.94

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	0.99	1.16	1.59	19.17
Jointly trained	0.99	1.57	3.99	2.85
Coupled	0.93	2.86	10.48	15.28

Roc curves



Training time

Table: Training time [Hours].

Weak classifier's training method	1 class	8 classes
Independent	8	111
Jointly trained	8	218
Coupled	8	103

Conclusions

- Summary:
  - A multiclass object detection system was proposed
- Main ideas:
  - An extention of Vector Boosting is introduced.
  - A multiclass boostrapping procedure is proposed.
  - The concept of coupled components on multiclass classifiers is proposed.
- Three training methods of multiclass weak classifiers are evaluated:
  - Independent weak classifiers:
    - It works well even when the classes are very different.
  - Joint weak classifiers
    - The training time does not scales well with the number of classes.
  - Coupled weak classifiers (proposed):
    - Fast training. Good performace when the features can represent all classes