How to detect faces  ?

Sébastien MARCEL

Yann RODRIGUEZ

Introduction

- First step before:
  - face tracking
  - face recognition
  - facial expression recognition
  - gesture recognition
- Related projects at IDIAP
  - BANCA, M4, COST
  - IM2.SA, IM2.ACP

Where is the face ?

- Perfect conditions:
  - uniform background
  - uniform lightning

Where is the face ?

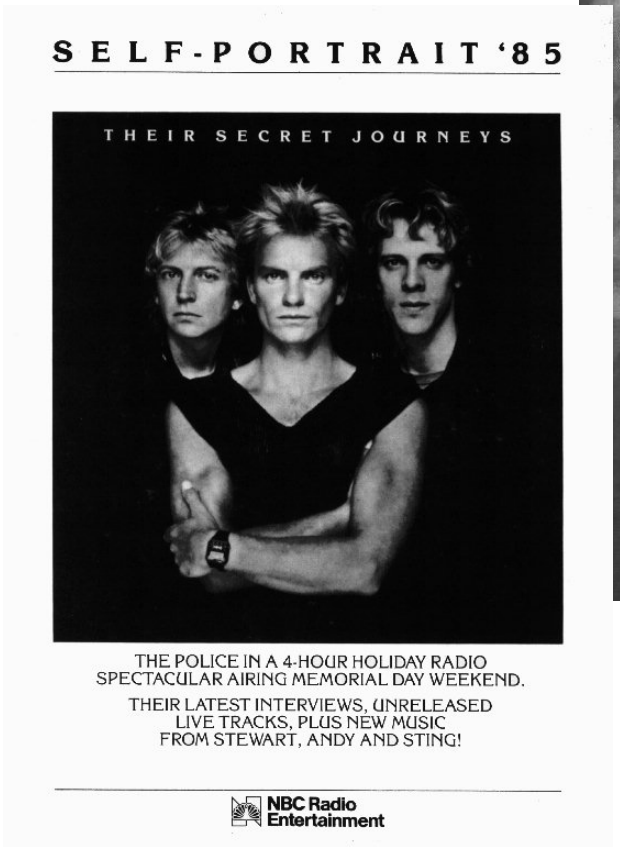
- Non-uniform lightning:

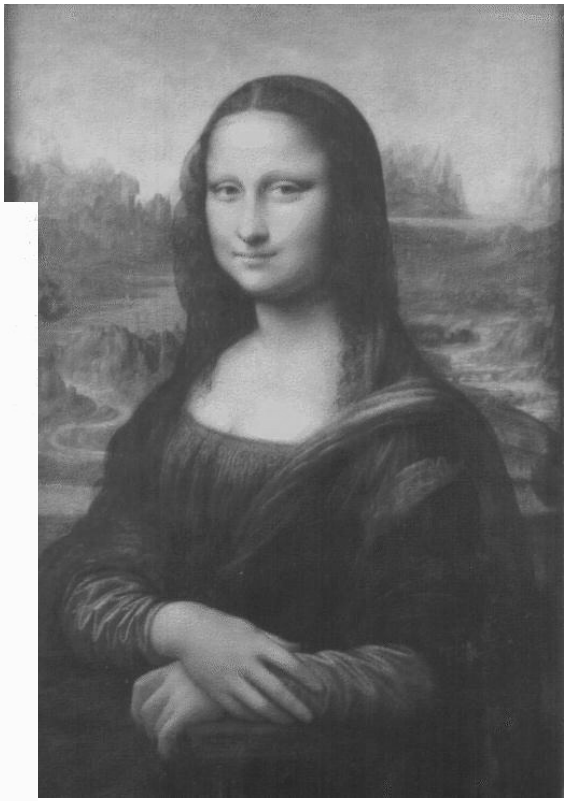
Where is the face ?

- Low quality images:

Where are the faces ?

- Scans:
  - gray level images
  - various quality





Where are the faces ?

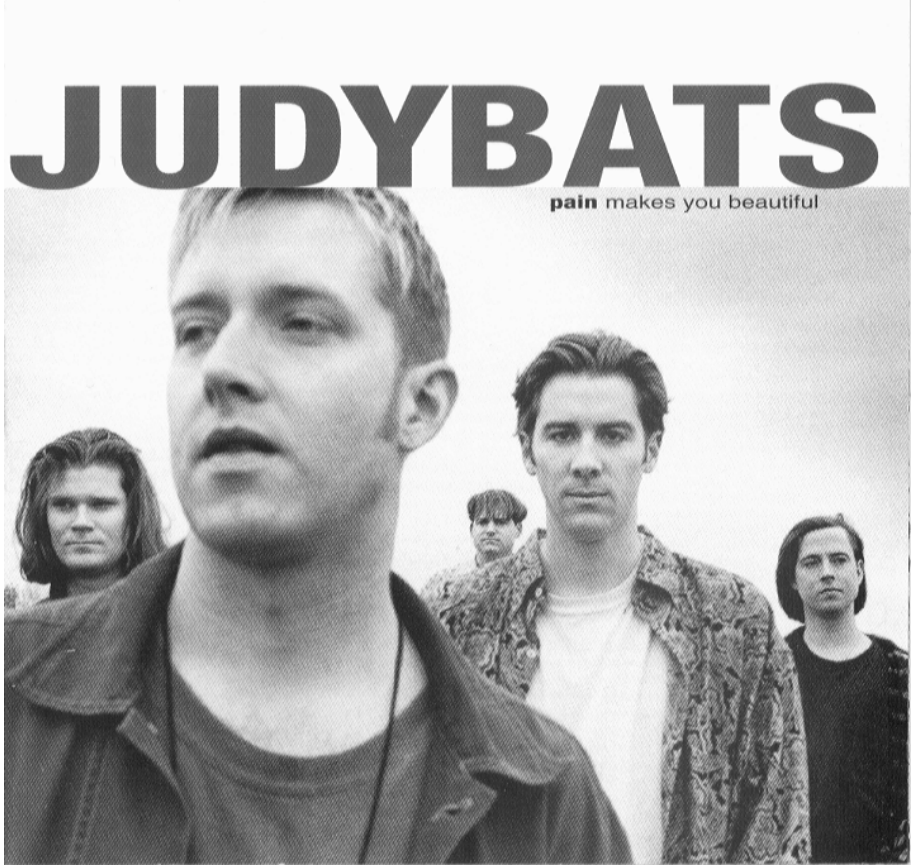
- Multiple faces (outdoor):

Where are the faces ?

- Multiple faces (indoor):

Where are the faces ?

- Multiple scales:



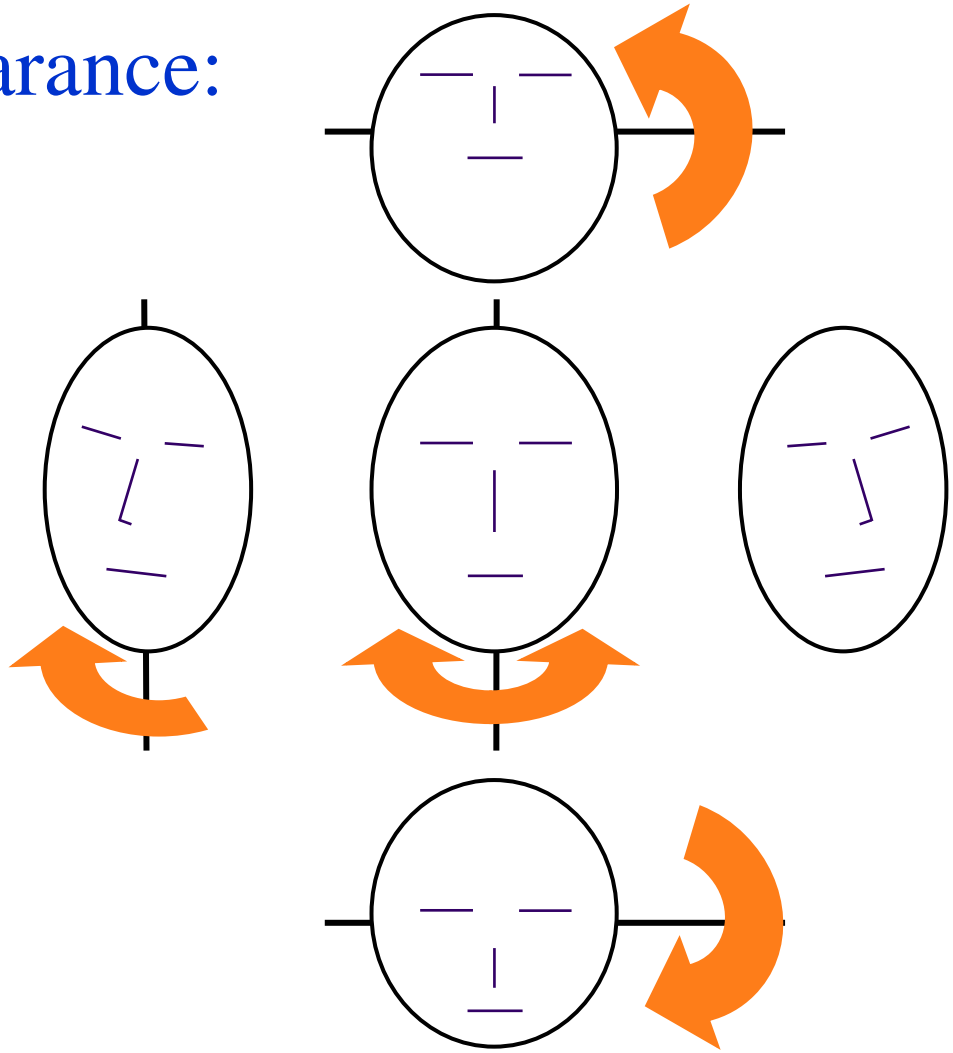
In-plane vs Out-of-plane rotations

- In-plane rotations:
  - rotation in the plane of the image
  - 1 degree of freedom
- Out-of-plane rotations:
  - rotation out-of the plane of the image
  - 2 degrees of freedom

Extreme case :-)

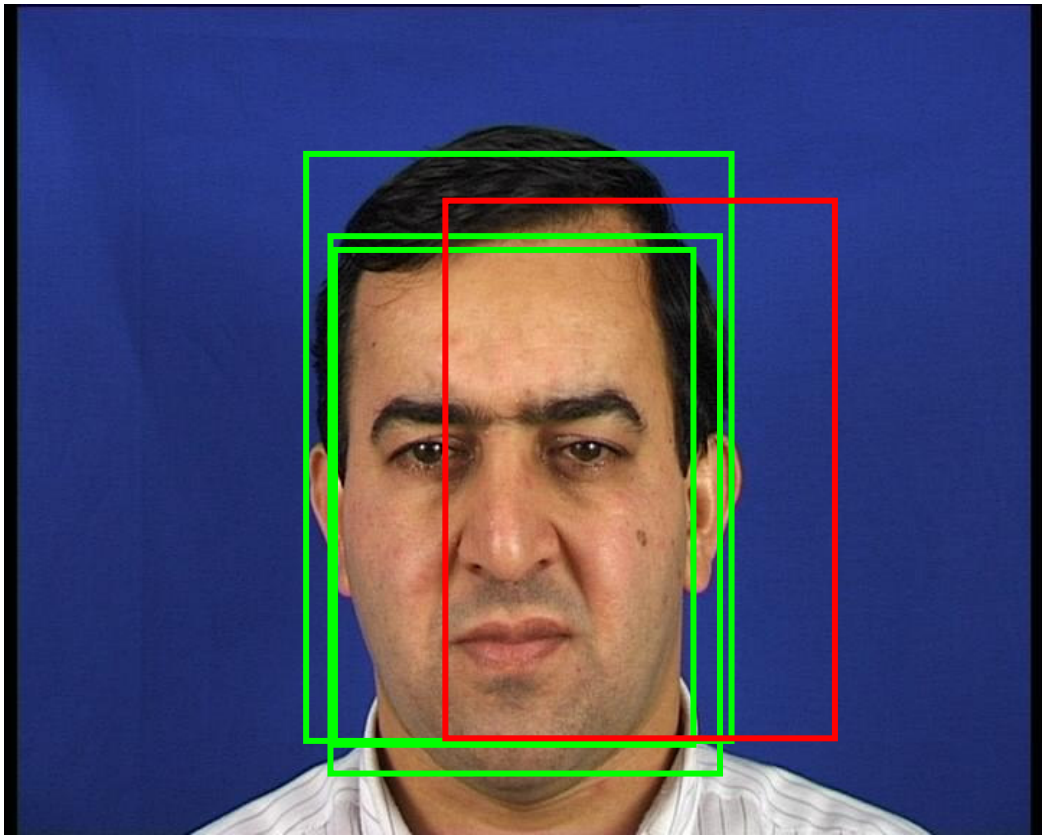
Appearance based models

- 1 model for each appearance:
  - frontal
  - profile/half profile
  - up/down



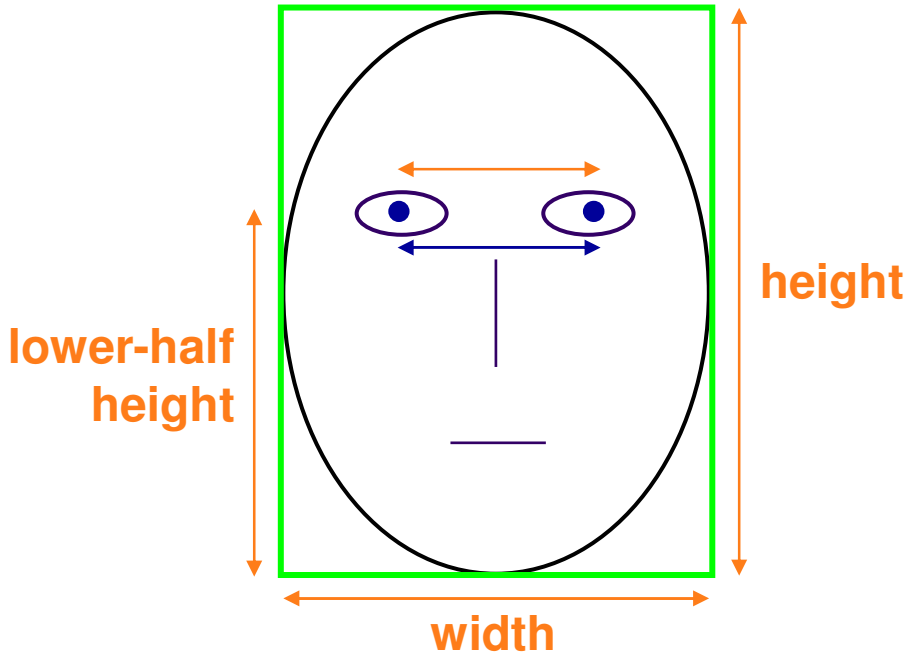
Face modeling

- Ideal detection:



Face modeling

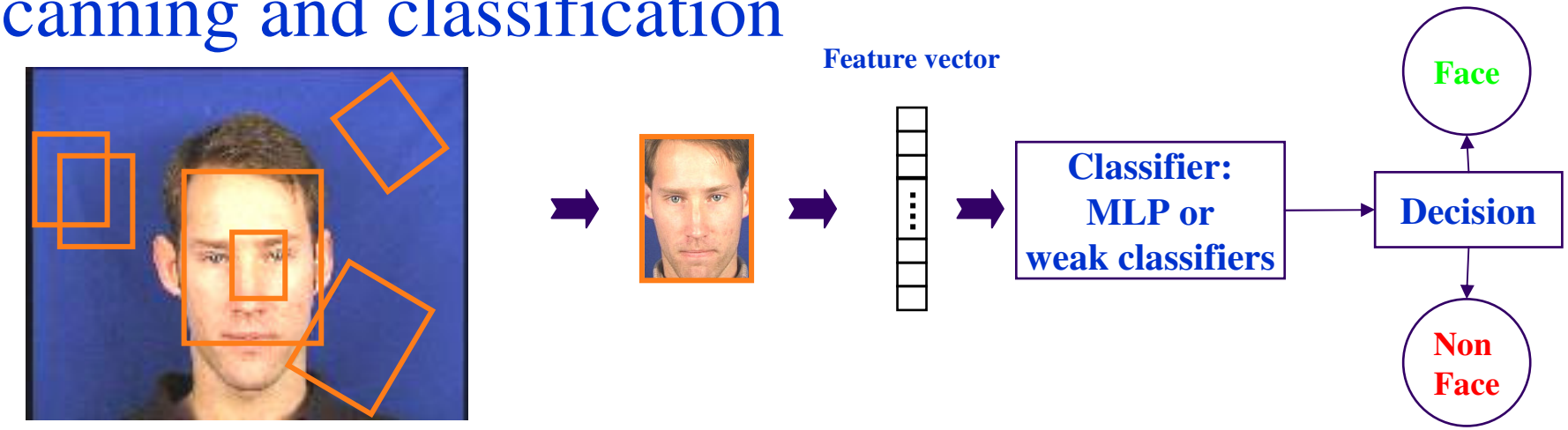
- Determining the face bounding box:
  - facial measures [1]
  - facial landmarks (eyes)
- Useful for:
  - database collection
  - evaluation



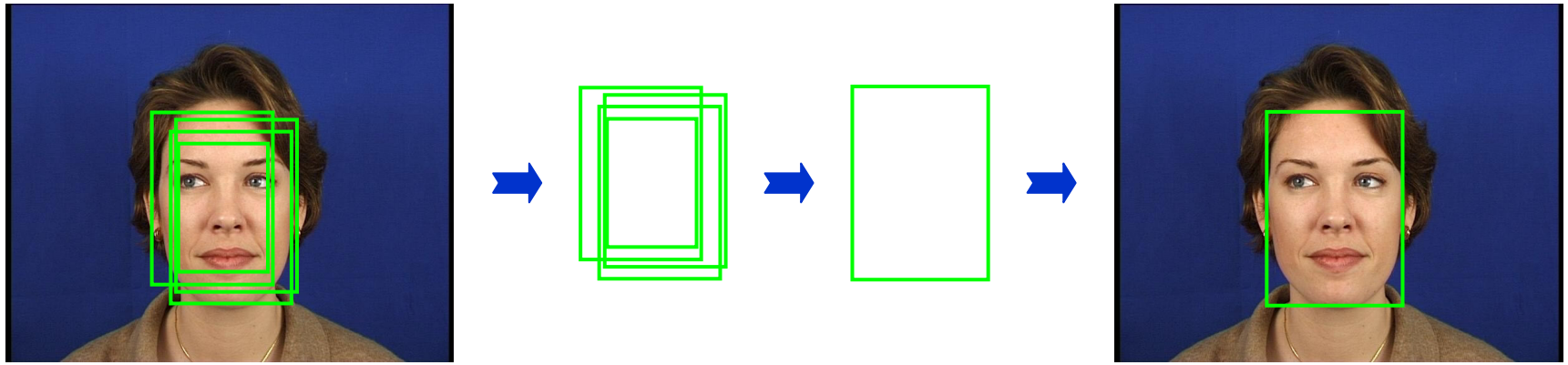
[1] "Anthropometry of the Head and Face" L.G. Farkas

Face detection in 2 stages

- Scanning and classification

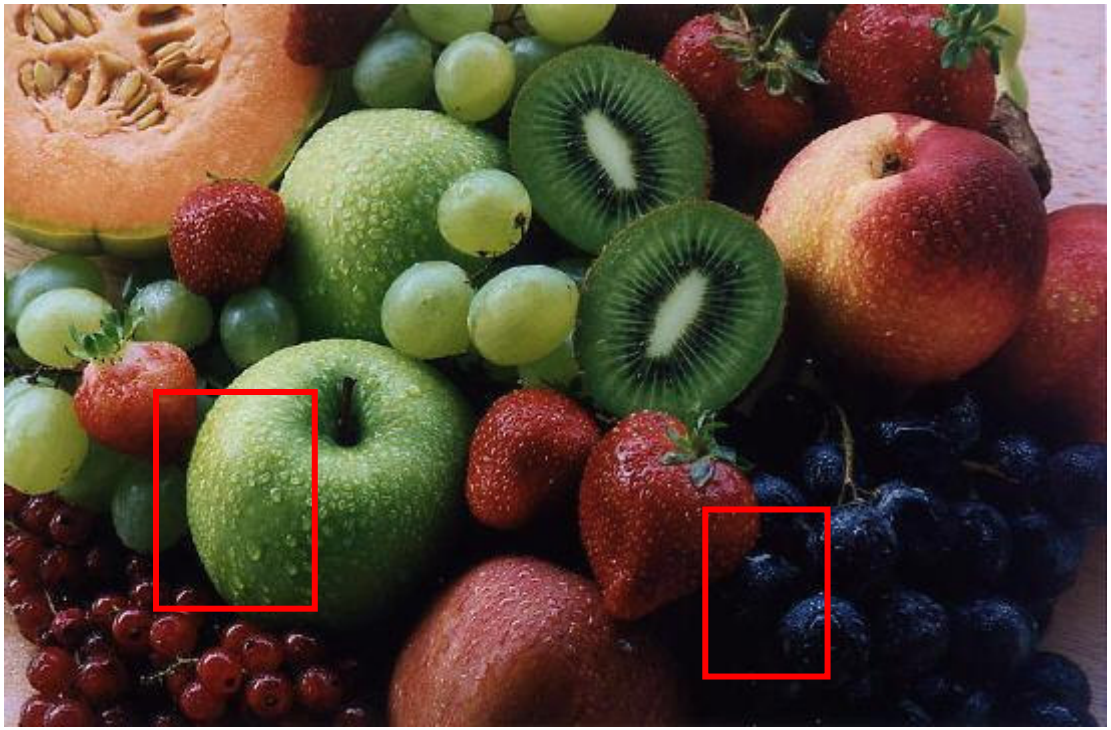


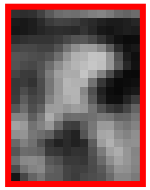
- Merging overlapped detections

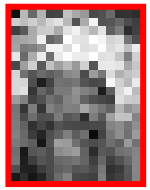


False detections

- ↗ number of tests: up to 1 million
- Targeted performance: < 10^{-7} false detection rate

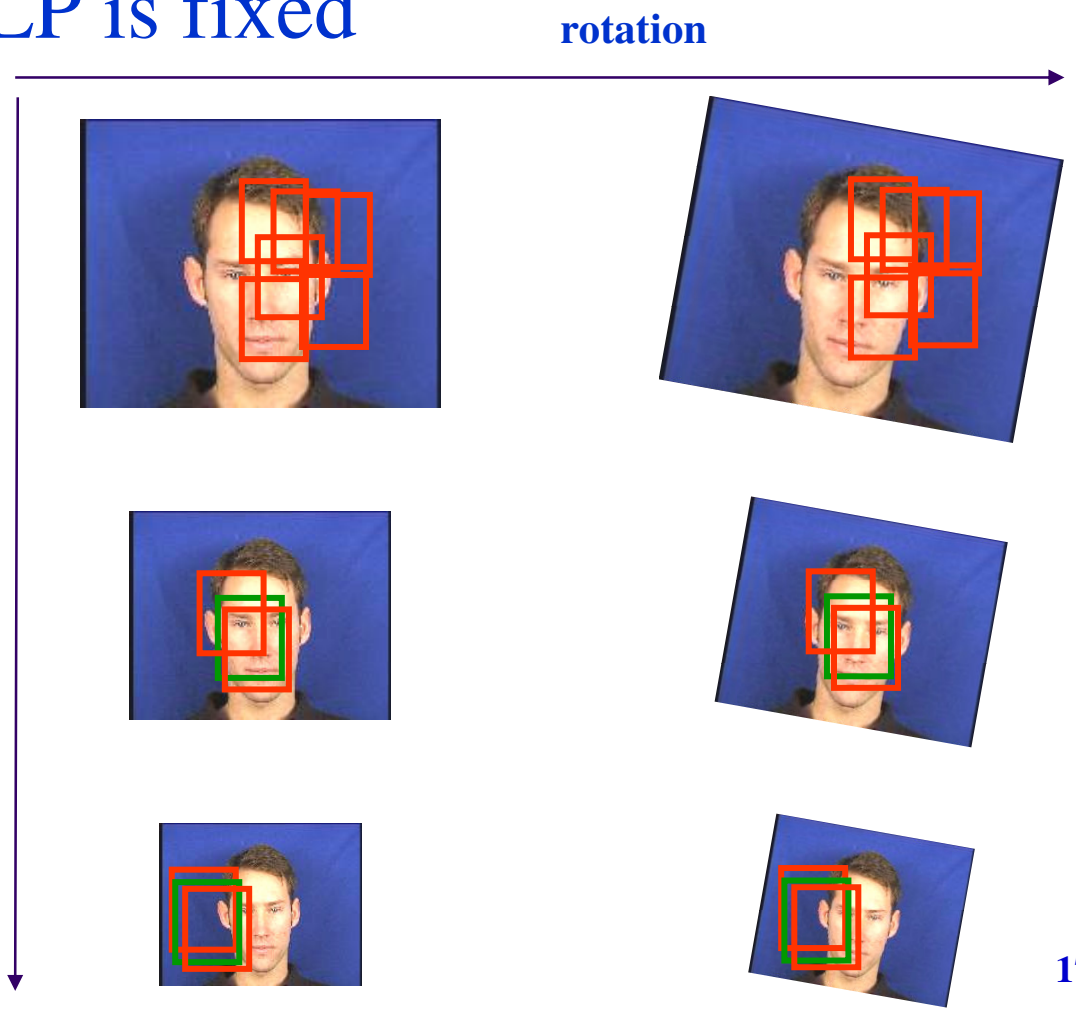






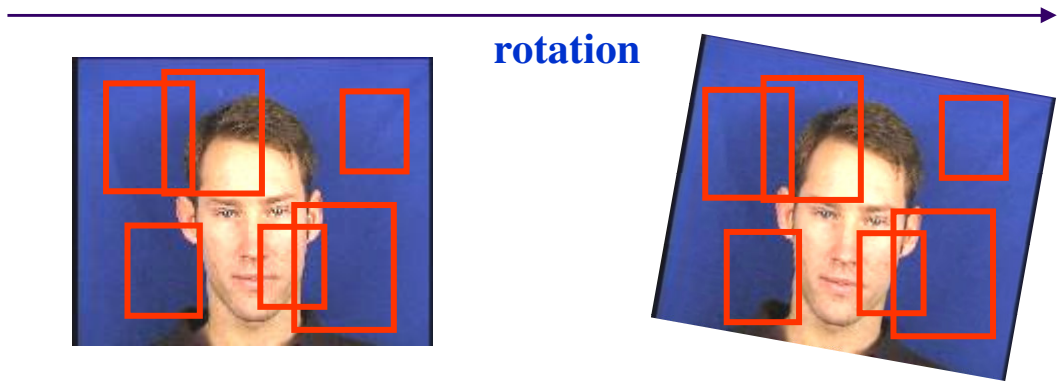
Using a MLP as classifier

- The input size of a MLP is fixed
- Scan a fixed window:
  - at any scale
  - at any rotation
  - at any location
- Low false detections
- Slow: 10s/image



A weighted sum of weak classifiers

- Using input features which can be computed at any scale:



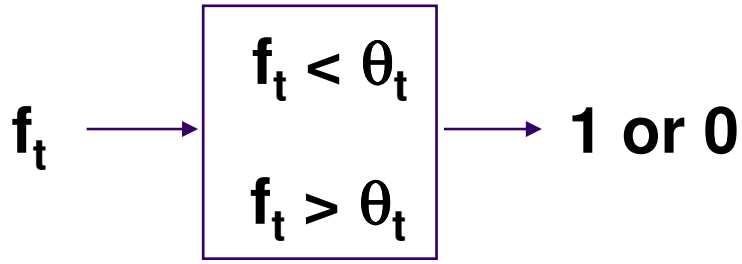
  - pixel based weak classifiers
  - Haar-like based weak classifiers
- Fast: 100ms/image
- Higher false detections than MLP

Weak classifiers

- Weak classifier: $\mathbf{h_t(x)}$

  $\mathbf{f_t}$: input feature

  θ_t : threshold



- Weighted sum of weak classifiers:

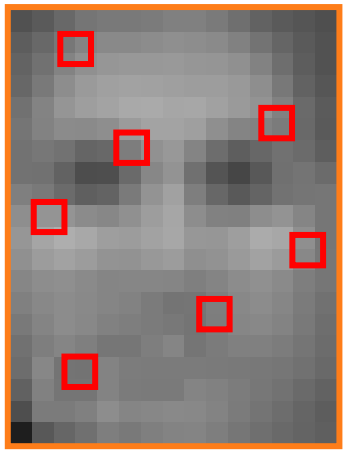
$$\mathbf{g(x)} = \Sigma_t \, \alpha_t \, \mathbf{h_t(x)}$$

- Boosting the performance of weak classifiers by combining them iteratively
- For more details see S. Bengio lectures

Pixel based weak classifiers

- The features are just the pixels of the image:

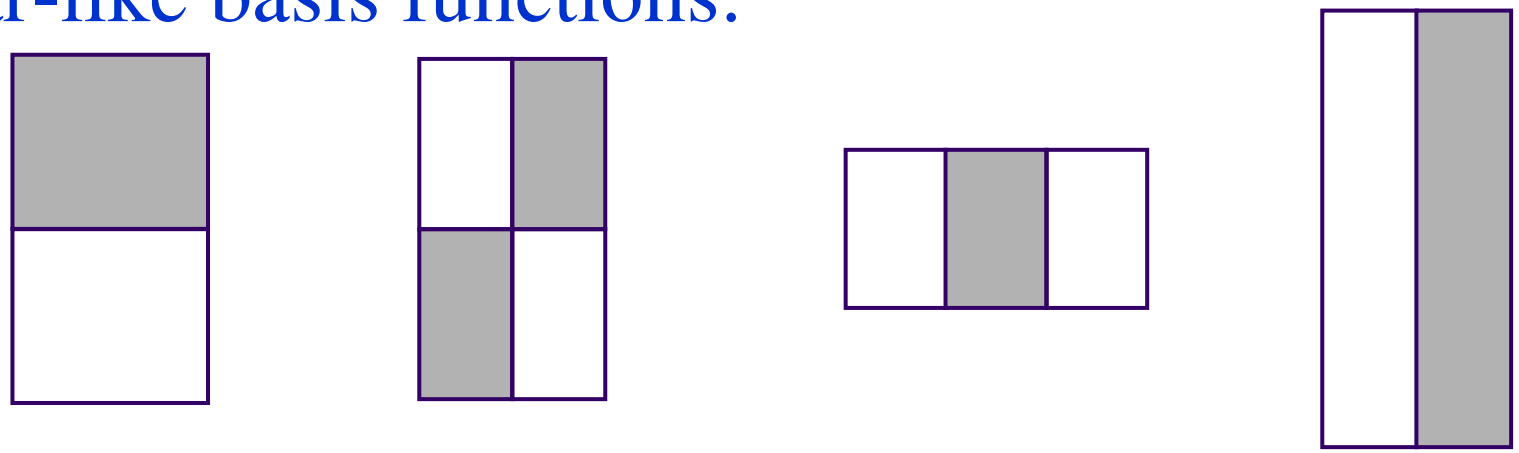
- Each weak classifier considers 1 pixel of the image **x**:



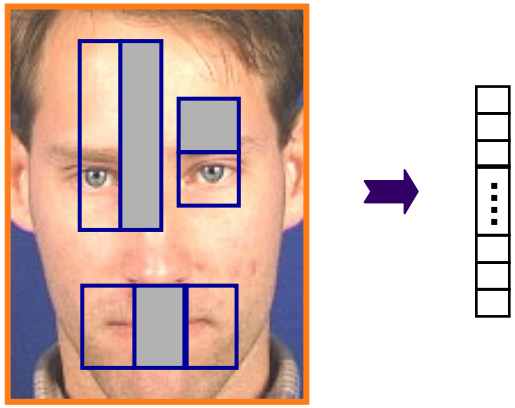
$$g(x) = \sum_t \alpha_t h_t(x)$$

Haar-like based weak classifiers

- Haar-like basis functions:



- Applying all masks at any scales and positions:

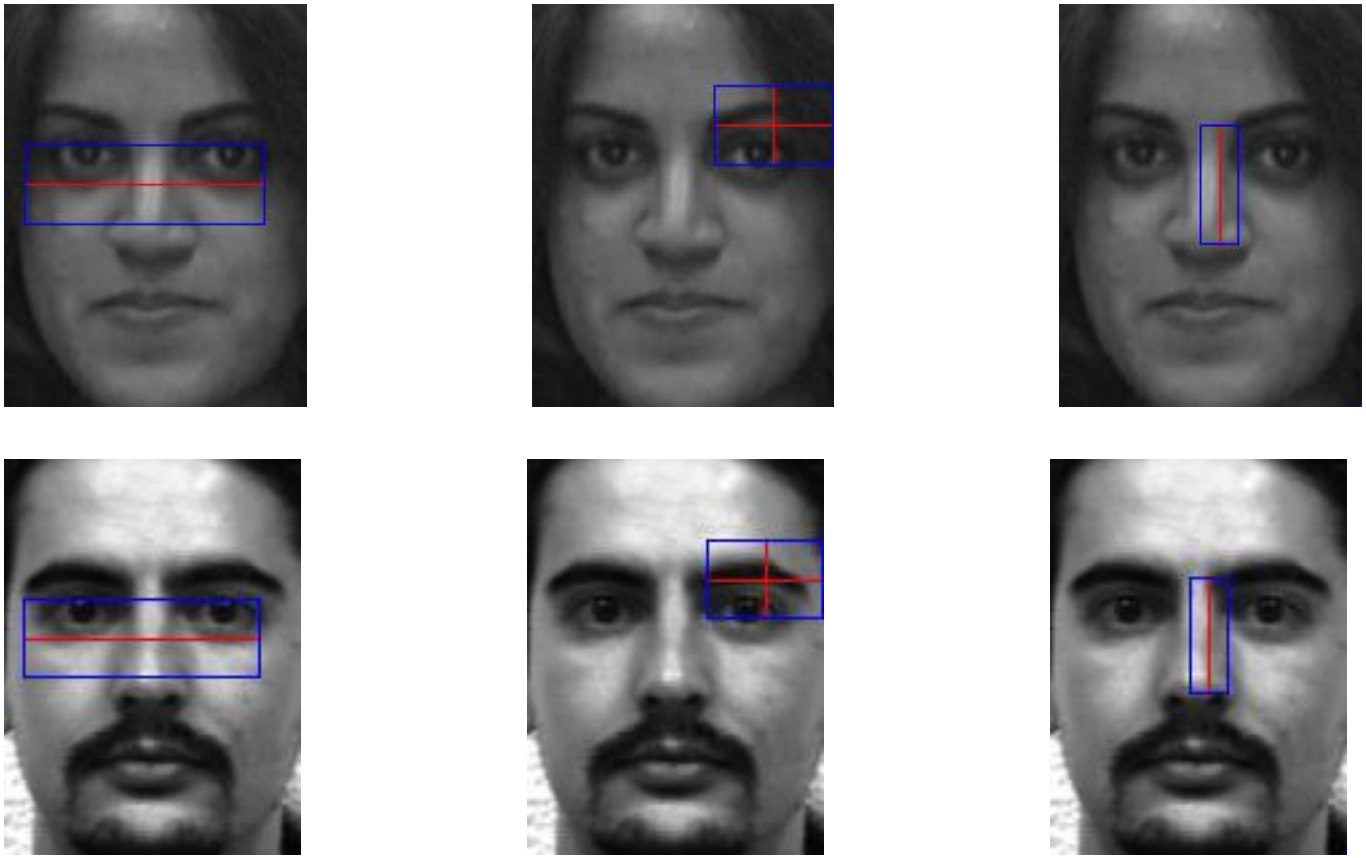


$$g(x) = \Sigma_t \alpha_t h_t(x)$$

Feature vector dimension between 1000 and 20000

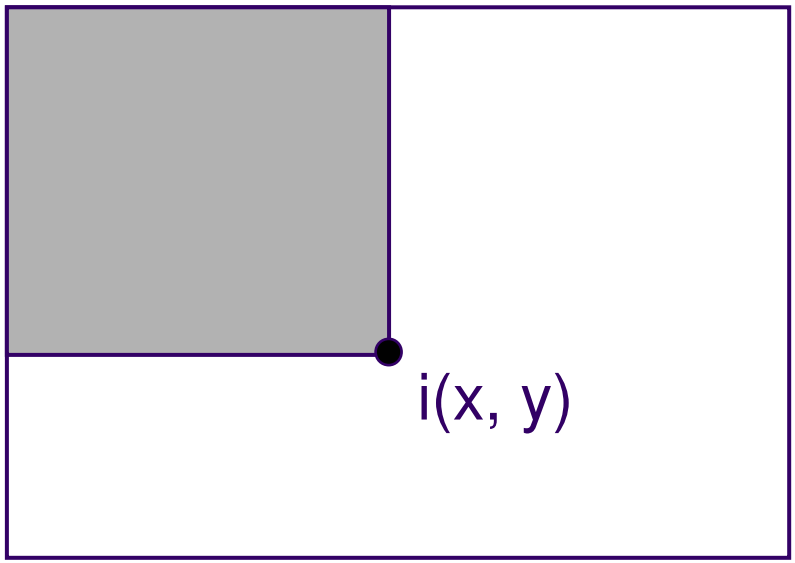
Haar-like based weak classifiers

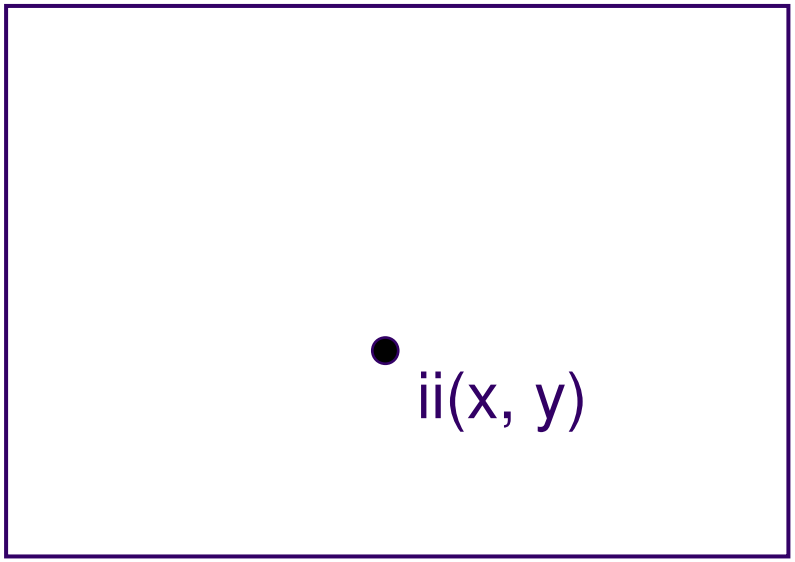
- AdaBoost selects a mask at a given position and size:



Integral Image (1/2)

- Def: The point (x, y) of the integral image is the sum of all the pixels in the upper-left corner of the original image

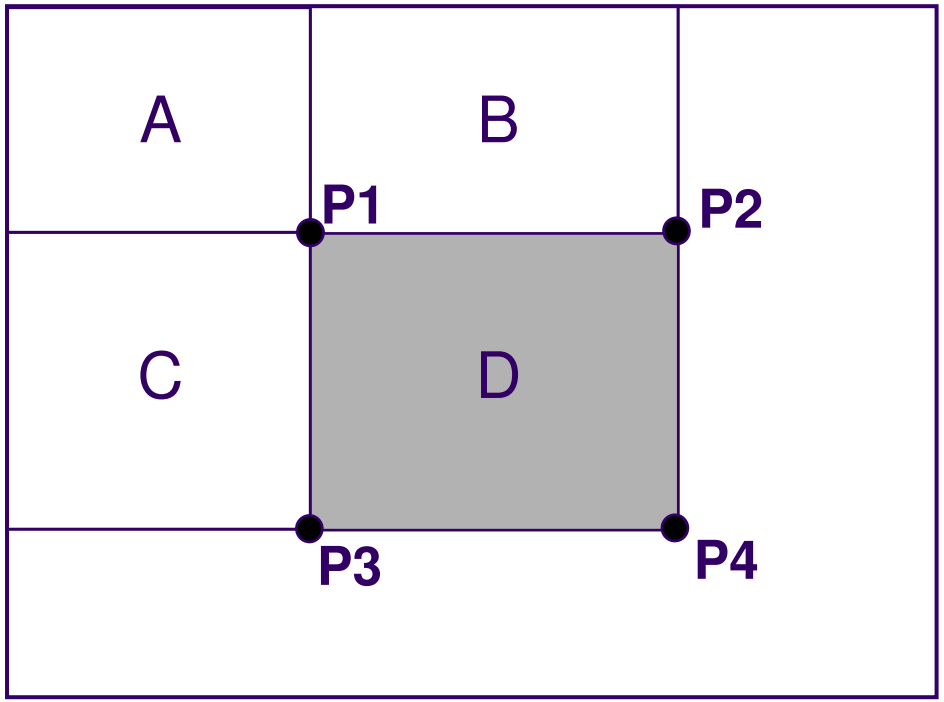




- Can be computed efficiently in 1 pass

Integral Image (2/2)

- The sum of pixels in D is P4 + P1 - (P2 + P3)



- Haar-like features are very fast to compute

Examples of detections

Demonstration

- Face detection in meetings:
  - FGnet dataset
  - video: 720x576 25fps
  - each frame is processed individually with Haar-like based weak classifiers

Conclusion

- Face detection software:
  - available to anybody at IDIAP
  - to process still images or videos
  - easy to modify
  - face classifiers are in **Torch3**
- Future work:
  - out-of-plane face models are missing
  - building cascade of weak-classifiers
  - evaluation on benchmark databases