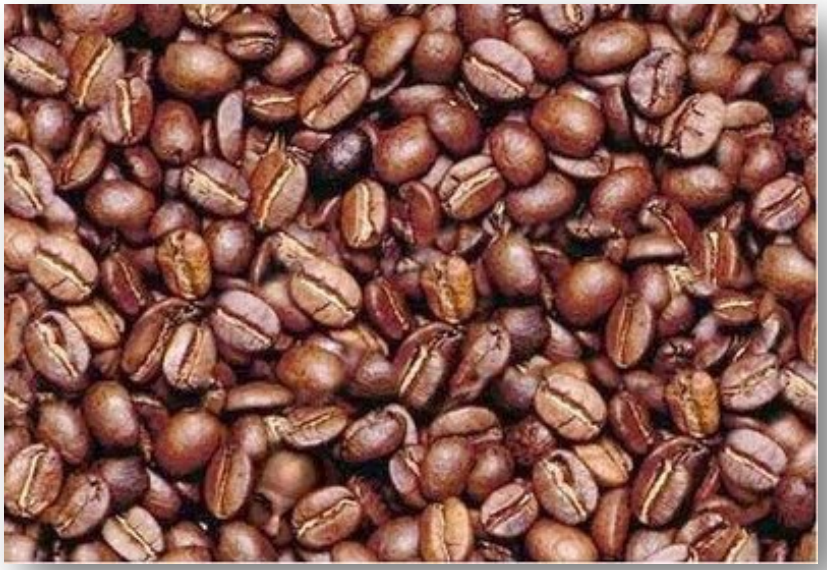
COMPUTATIONAL VISION:
Face Detection

Outline

1. Basic Concepts
2. Visual features
3. Ensemble learning
4. Cascade of classifiers

Face detection



“Hidden Face by Coffee Beans”

Face detection

Face detection

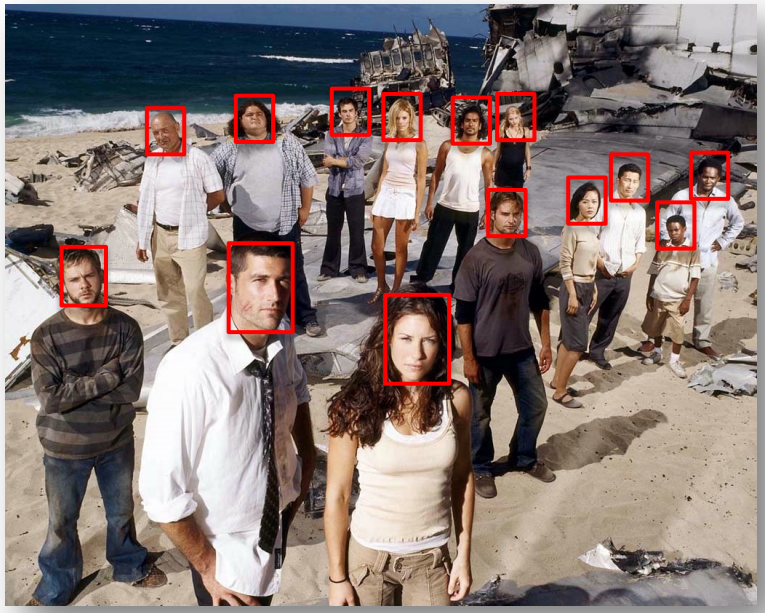
- The first step to face analysis
  - Given an image, where are the faces?

Identify and locate human faces in an image regardless of their position, scale, in plane rotation, orientation, pose and illumination.

→A very difficult problem

Face detection and recognition

Detection



versus

Recognition

Jack

Kate

Sawyer

?

Face detection and recognition

- Intra-class variability

- Inter-class confusion

Paul Newman/Marlon Brando

Face detection

- Classical approach for face detection:

  P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features*. Proc. CVPR, 1:511-518, 2001.

Face detection

Objectives of Viola & Jones method:

- Accurate detection of faces
- Fast Algorithm
- Real-time detection (video processing)

With a camera, we do not want to wait too much time to take a picture!

Face detection

Viola & Jones method

- Posed as a standard *pattern recognition problem*
- The main steps of a pattern recognition problem are:
  1. Feature extraction from the image
  2. Training of a classifier
  3. Test of new images using the trained classifier

Pattern recognition scheme

Training images

TRAINING

Features Extraction

Training Classifier

Test image

Features Extraction

TESTING

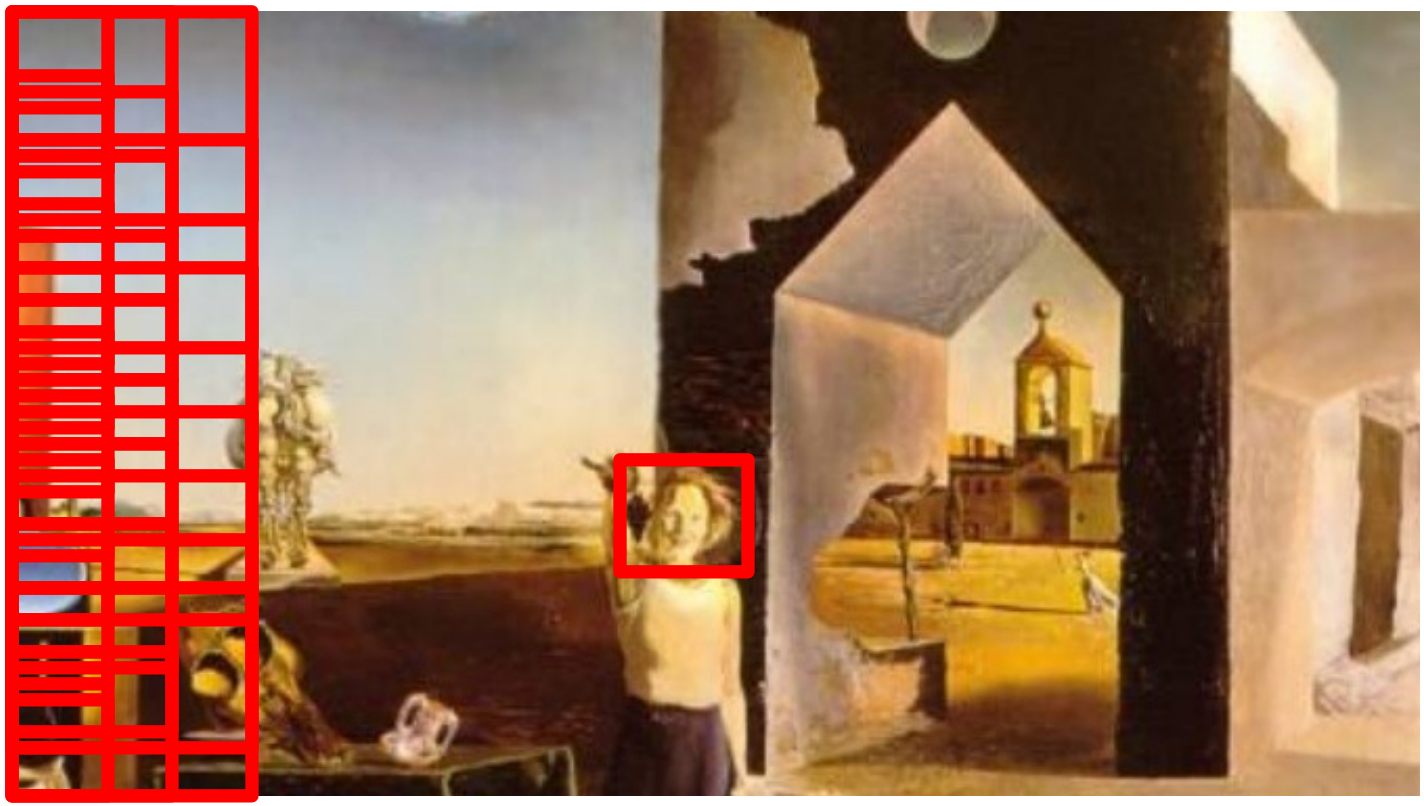
Face detection

Basic concepts for understanding Viola & Jones method:

1. “Rectangular” features (Haar-like features)
2. Integral Images
3. AdaBoost
4. Cascade of classifiers

Face detection

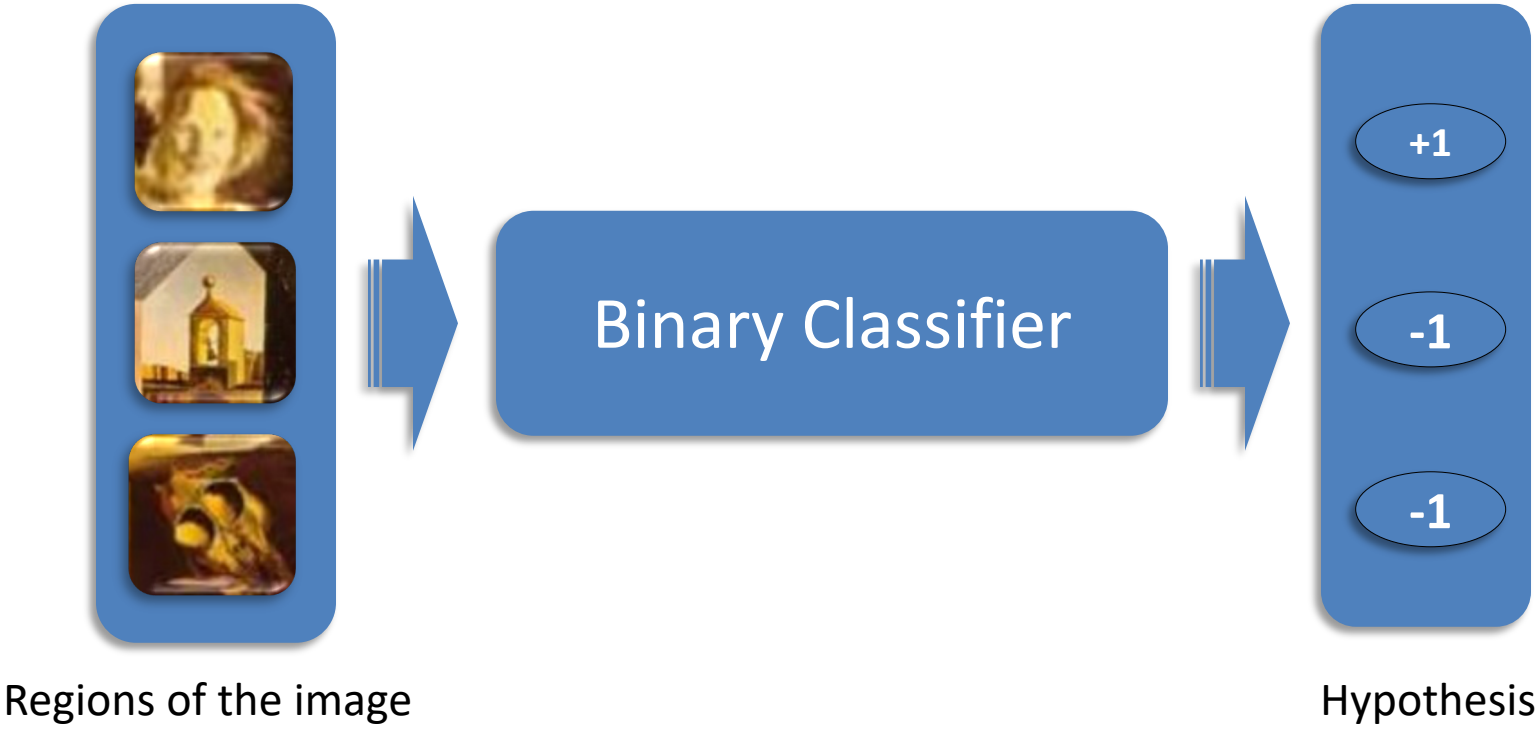
- Windowing strategy



Binary classification problem

Face detection

- Basic Concepts

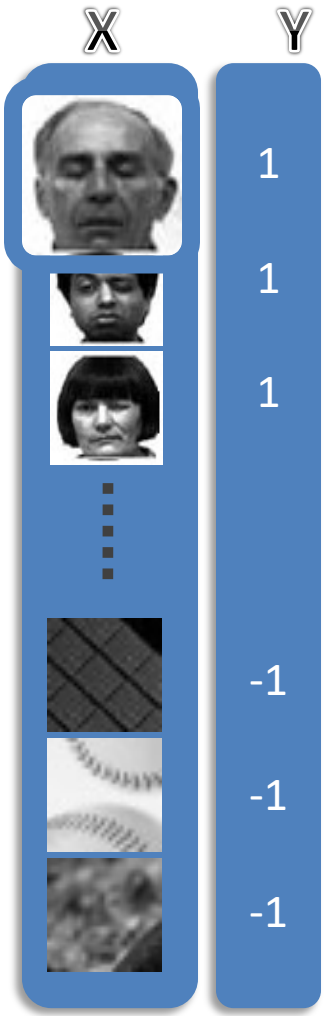


Face detection

- Restrictions
  - Large number of regions to analyze (after windowing strategy)
  - Unbalanced problem
    - Most of the regions are from non-face class
    - Only few windows or none have a face

Face detection

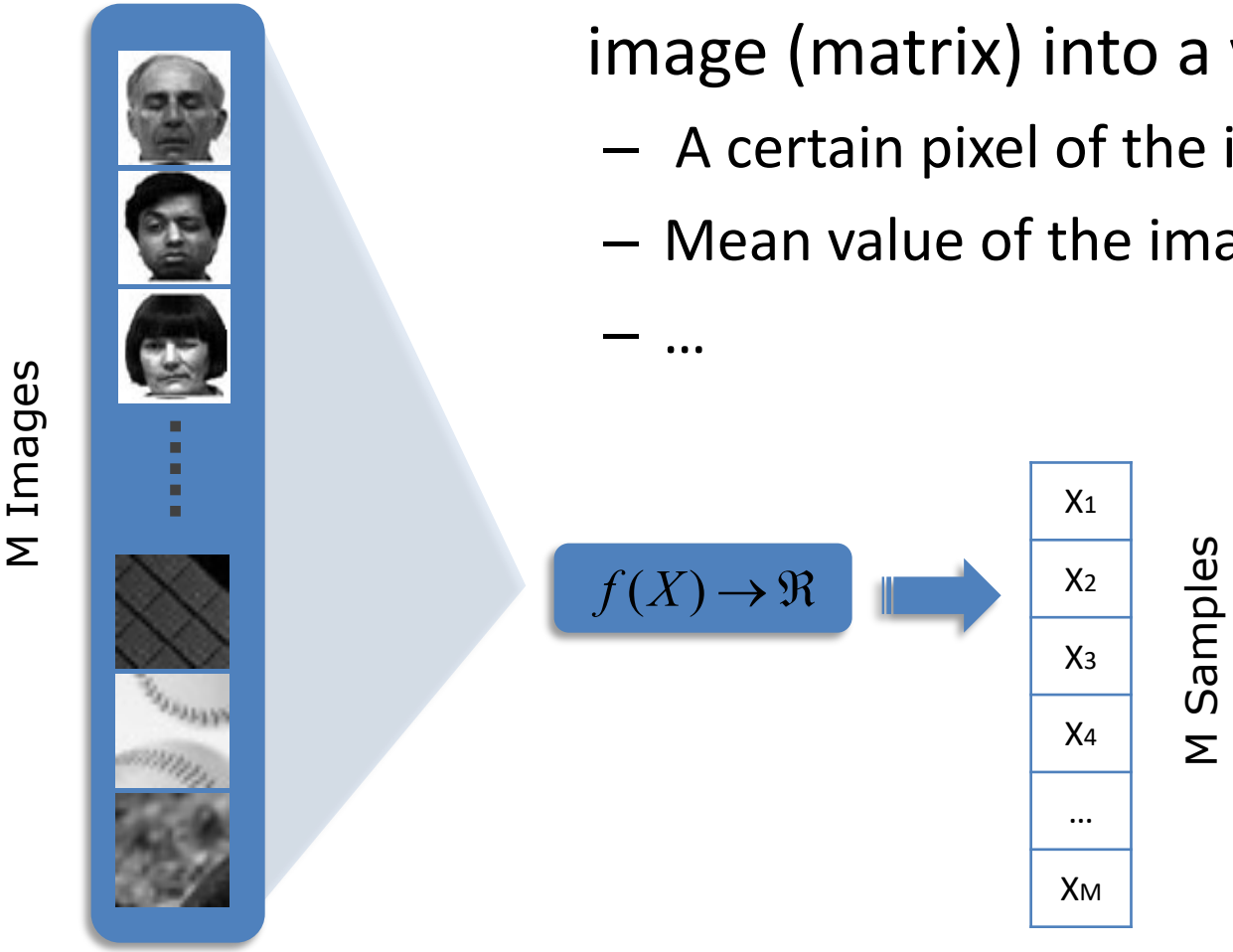
- Samples set
  - Positive and negative image samples (X)
  - Labels (Y)
- An image is a matrix



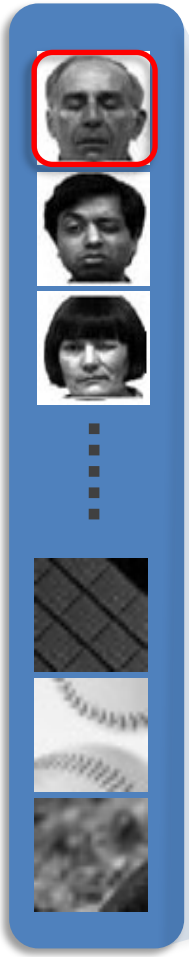
140	140	140	121	121	140
141	142	120	121	121	142
143	121	121	200	200	50
121	121	204	201	200	50
121	204	202	198	250	2
204	203	198	150	250	5

Face detection

- *Feature*: function that transforms an image (matrix) into a value
  - A certain pixel of the image
  - Mean value of the image
  - …



Face detection



- *Feature Set*
  - All pixels of the image
  - Family of functions
    - Filter banks
  - SIFT descriptor

$$f_1(X) \to \Re$$
$$f_2(X) \to \Re$$
$$\dots$$
$$f_N(X) \to \Re$$

N features

X_{11}	X_{12}	X_{13}	...	...	X_{1N}
X_{21}	X_{22}	X_{23}	...	...	X_{2N}
X_{31}	X_{32}	X_{33}	...	...	X_{3N}
X_{41}	...	...	...	...	X_{4N}
...	...	...	...	...	...
X_{M1}	...	...	...	...	X_{MN}

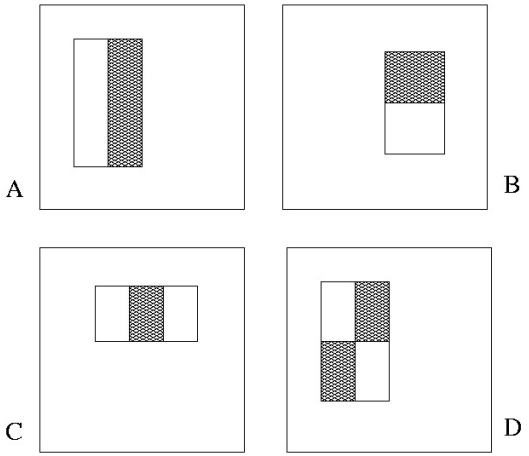
M Samples

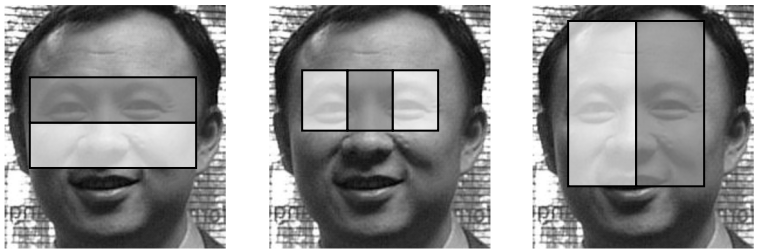
Face detection

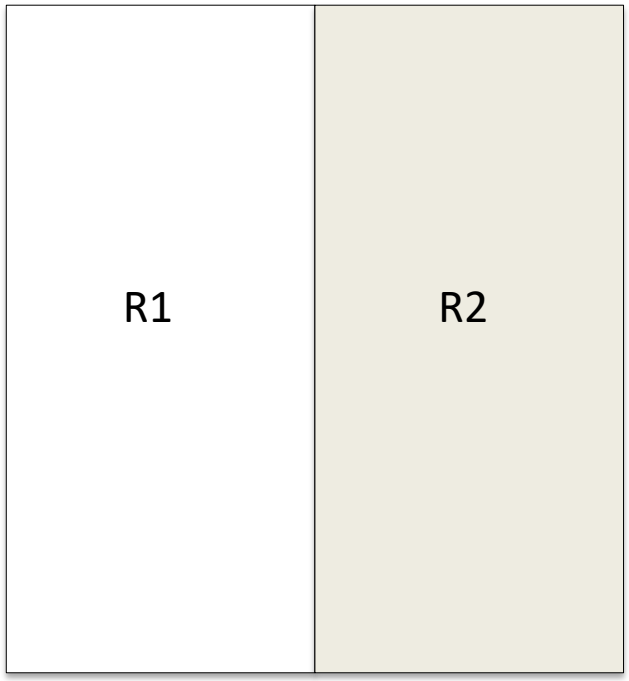
- Visual Features
  - Designed to be sensitive to visual artifacts in the objects
    - Contrasted regions
      - Edges
      - Corners
      - …

Face detection

- *Rectangle features*





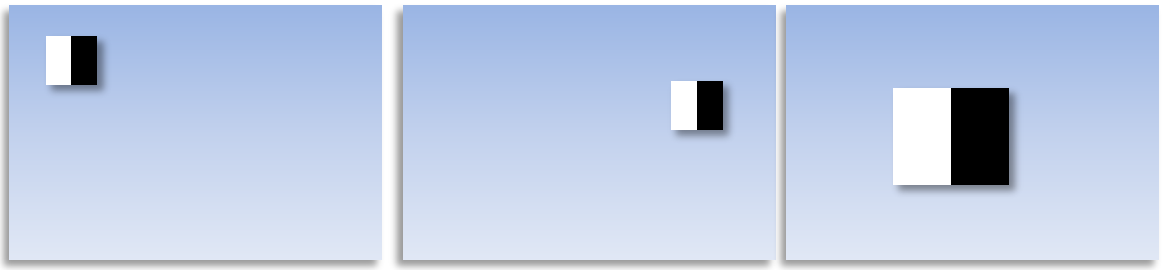


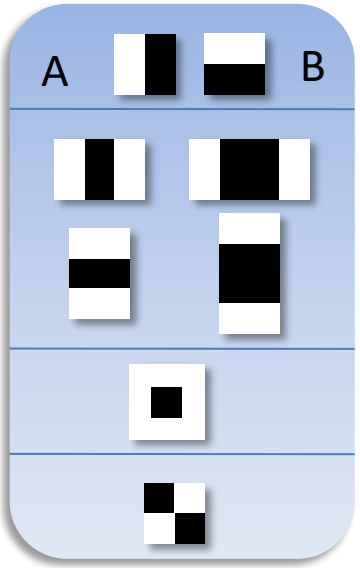
The k-th feature:

$$F_k = \sum_{(i,j)\in R1} I(i,j) - \sum_{(i,j)\in R2} I(i,j)$$

Face detection

- Set of features: different sizes, shapes and positions of the regions with respect to the window.
- Region configurations to detect
  - Edges
  - Lines
  - Center-surround structures
- Definition in a training window



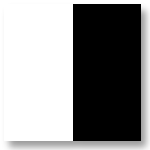


- For a 24x24 detection region, the number of possible rectangle features (A and B) is over 180,000.

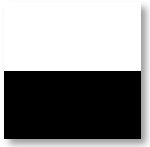
Face detection

- Feature interpretation:

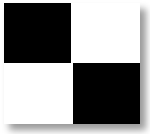
  Which structures of the image are they showing?



Horizontal changes



Vertical changes

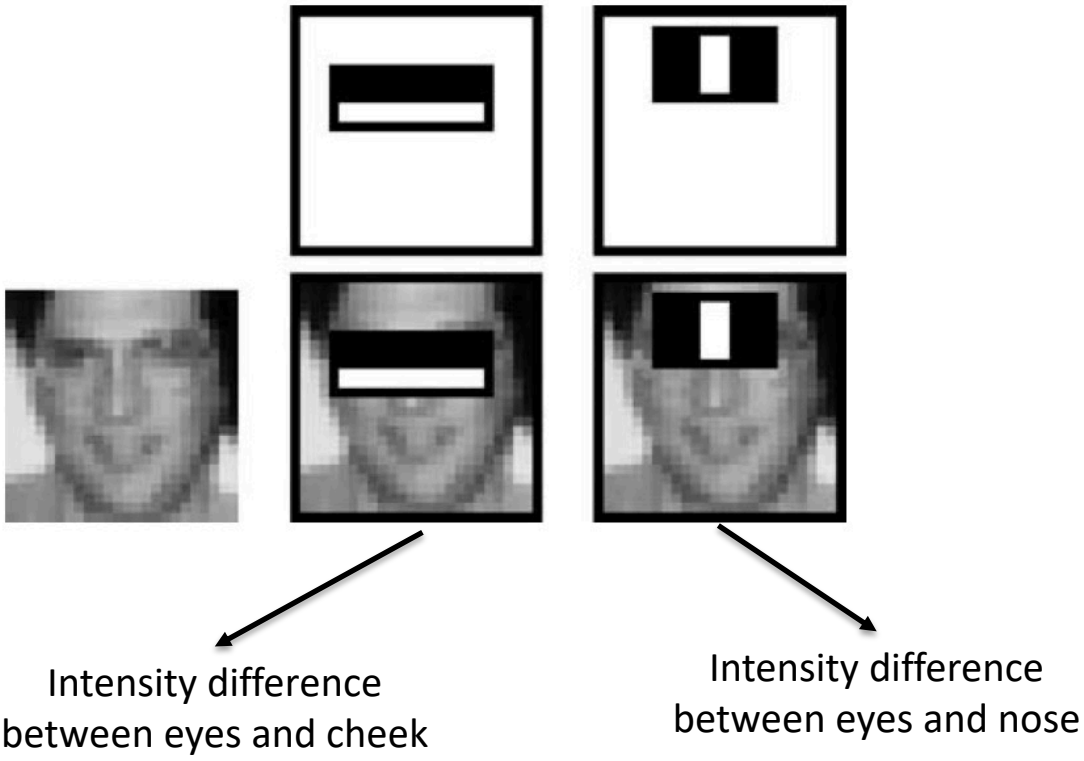


Diagonal changes

Face detection

- Feature interpretation:

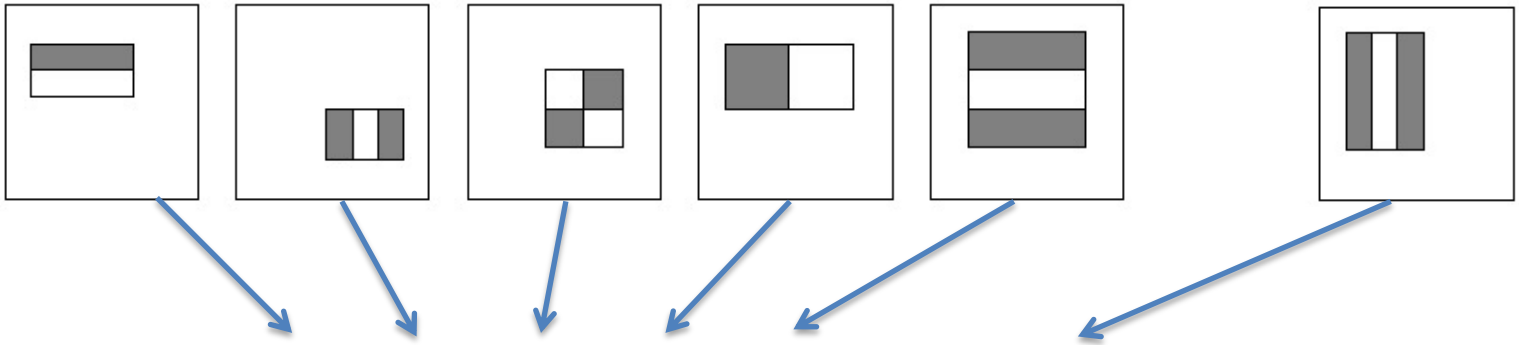
  And apply to faces, which information are they providing?



Face detection

- Feature extraction:

  Given the set of masks, a feature vector of the window is built:



$$\mathbf{X} = [\, F_1, F_2, F_3, F_4, F_5, \ldots, F_K]$$

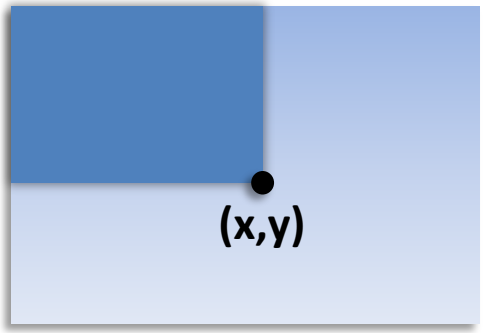
The feature vector describes the content of the window and it is used to train the classifier and to detect faces.

Face detection

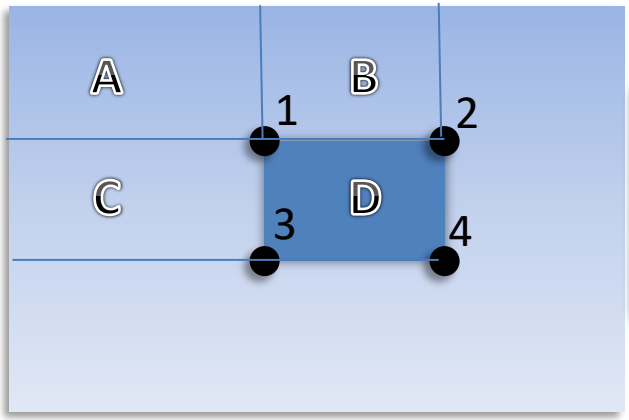
How rectangle features are computed in a fast way:

- Integral Image

$$II(x,y)=\sum_{i=1}^{x}\sum_{j=1}^{y}I(i,j)$$



We use the integral image to efficiently compute rectangle sums:



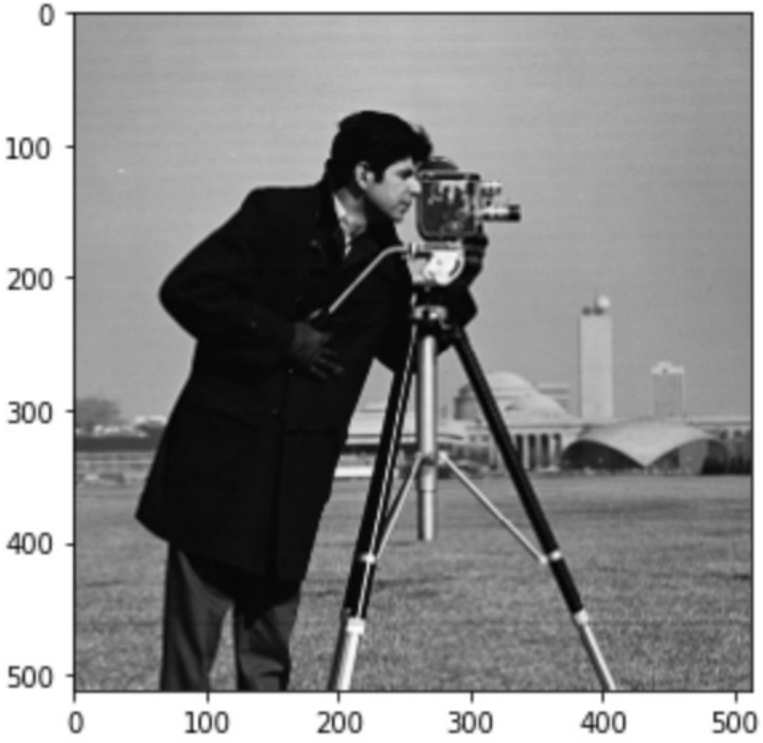
$$\begin{aligned} D &= 1+4-(2+3) \\ &= A+(A+B+C+D)-(A+B+A+C) \\ &= D \end{aligned}$$

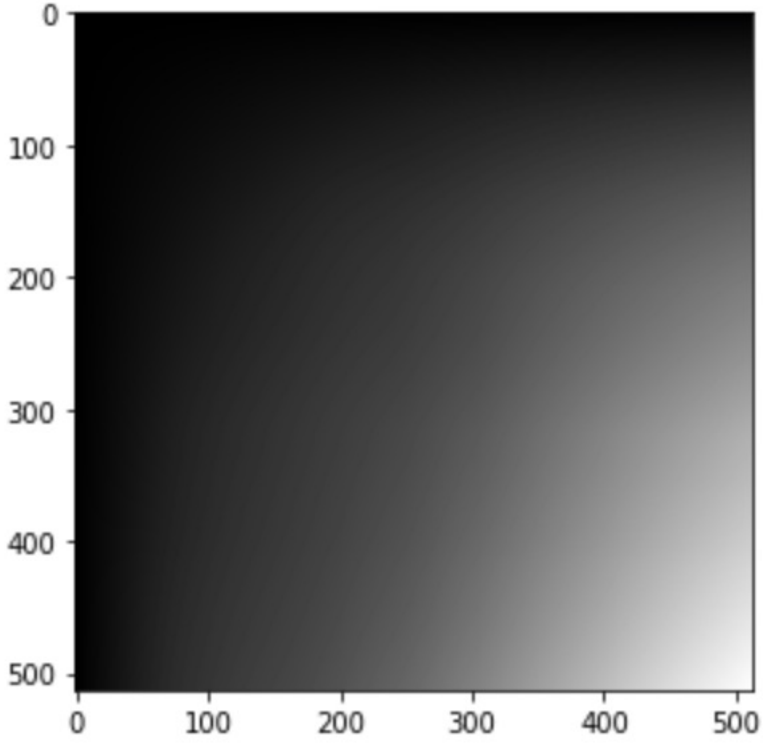
Any rectangular sum can be computed in constant time and it is independent of the size of the rectangular area.

Face detection

In Python:

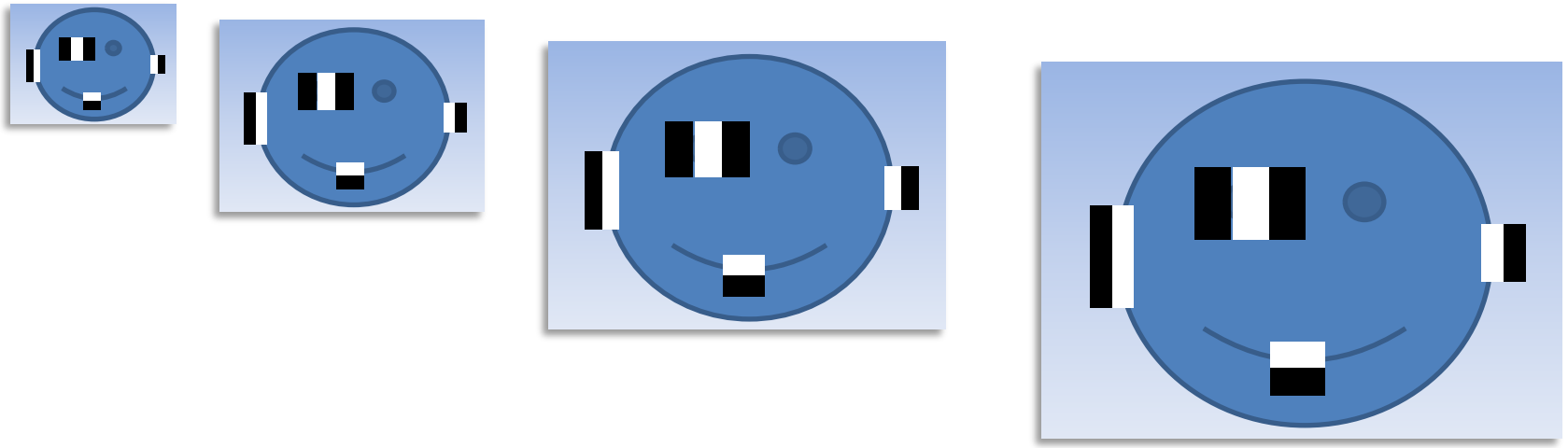
- `np.cumsum(np.cumsum(img_arr, axis=0), axis=1);`





Face detection

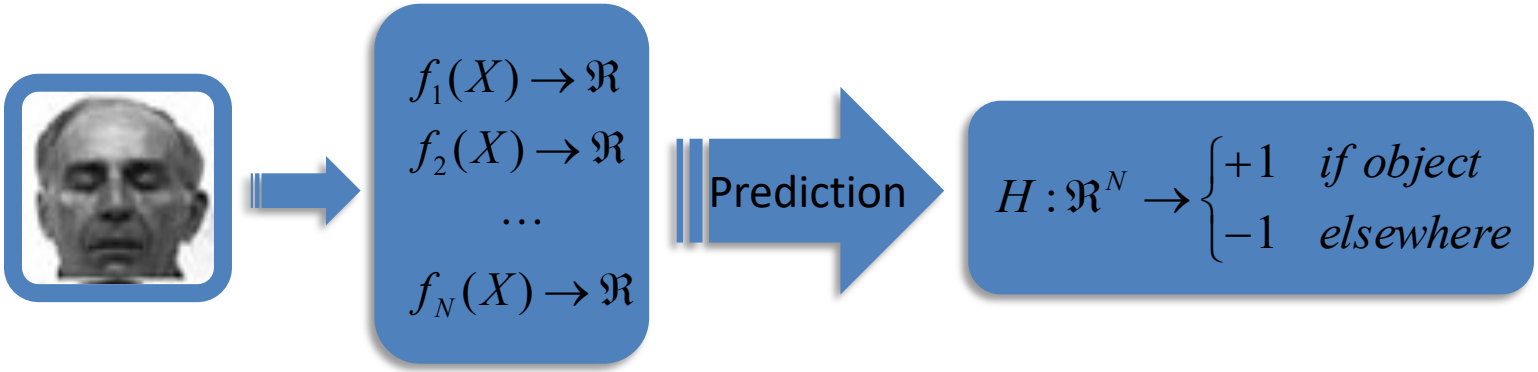
- Integral Image enables to evaluate all rectangle sizes in constant time. Therefore, no image scaling is necessary.



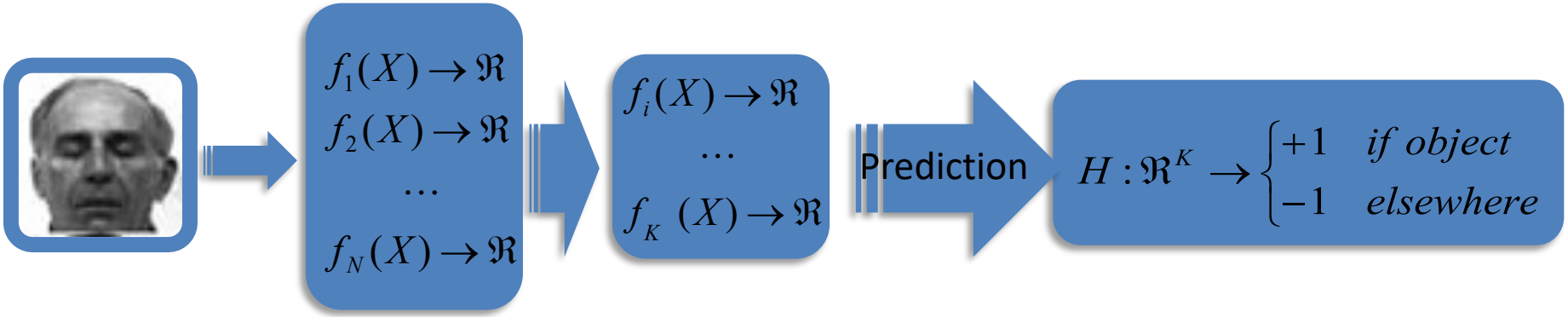
- Multi-scale detection by means of region scaling
  - Just scale the parameters of the features

Face detection

- Given an image (region), predicts its class
  - Using all the N features



  - Using a subset of K features: **Feature selection**



Face detection

- Feature selection
  - Remove noisy features
  - Time restriction
    - Repetitions of the method (Object detection)
    - Large feature sets
- Alternative: Classification algorithm

  AdaBoost (Adaptive Boosting)
  - Feature selection
  - Strong classifiers from simple classifiers
    - Ensemble of classifiers

Face detection

AdaBoost

- Introduced by Freund & Schapire in 1999.

- Combine several weak classifiers to build a single strong classifier

  >> BOOSTING

- Weak classifiers are defined in each iteration and are devoted to misclassified examples

  >> ADAPTIVE

Face detection

AdaBoost revision

- Weak classifier/hypothesis
  - Decision stumps
    - 1 feature + Threshold

X = 

$$f_i(X) \rightarrow \Re$$

$$h(X) = \begin{cases} +1 & if\ f(X) \geq Thr \\ -1 & elsewhere \end{cases}$$

- Strong classifier/hypothesis

X = 

$$h_1(X) \rightarrow \{-1,+1\}$$
$$h_2(X) \rightarrow \{-1,+1\}$$
$$\ldots$$
$$h_m(X) \rightarrow \{-1,+1\}$$

$$H(X) = sign\left(\sum_{i=1}^{m} h_i(X)\right)$$

Face detection

AdaBoost revision

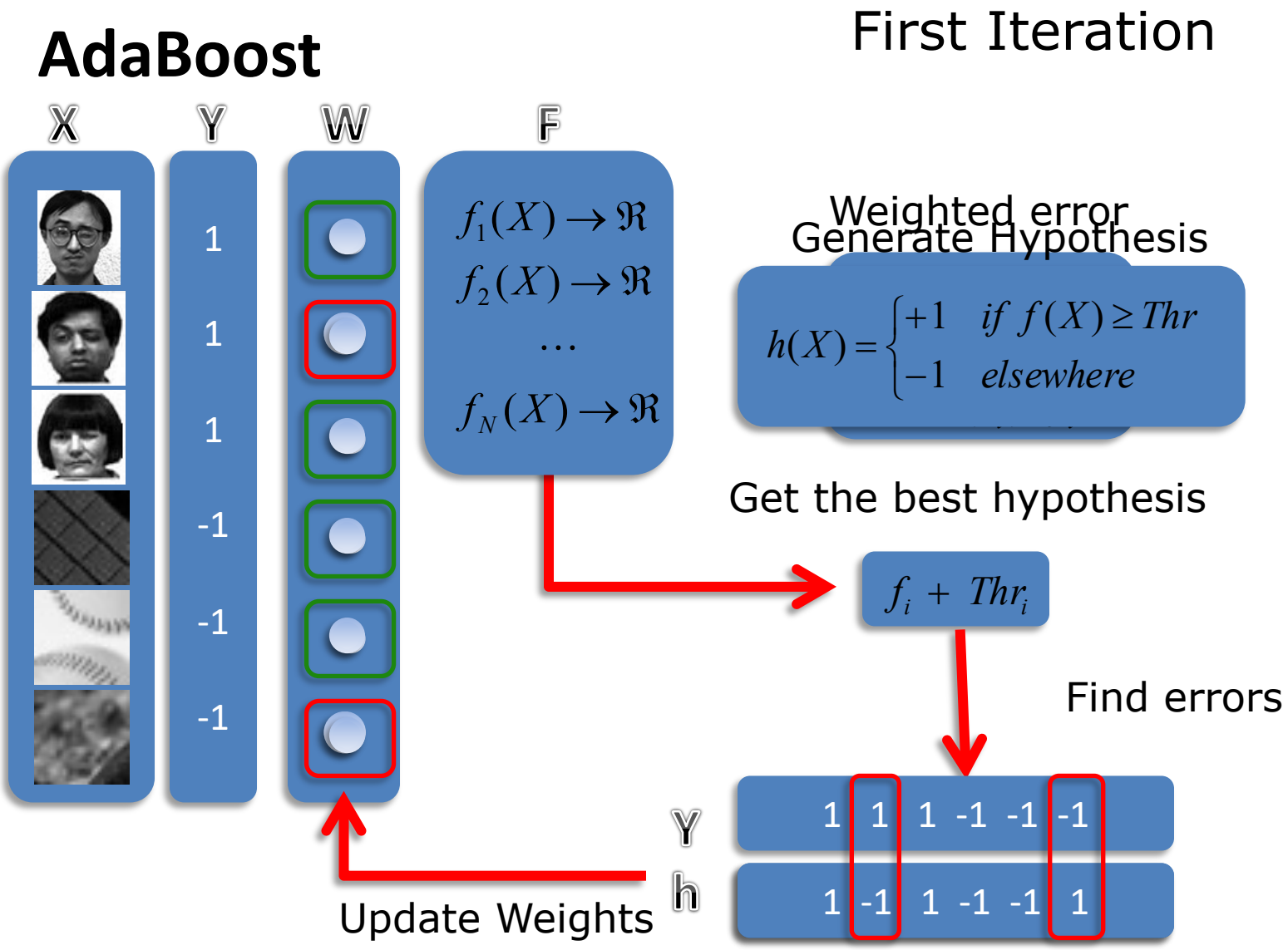
- Weighting strategy
  - Focus on **difficult samples** by adding a weight to each sample:

$$X = \{x_i \,|\, i = 1:N\}$$
$$Y = \{y_i \,|\, i = 1:N\}$$
$$W = \{\omega_i \,|\, i = 1:N\}$$
$$\sum_{i=1}^{N} \omega_i = 1$$

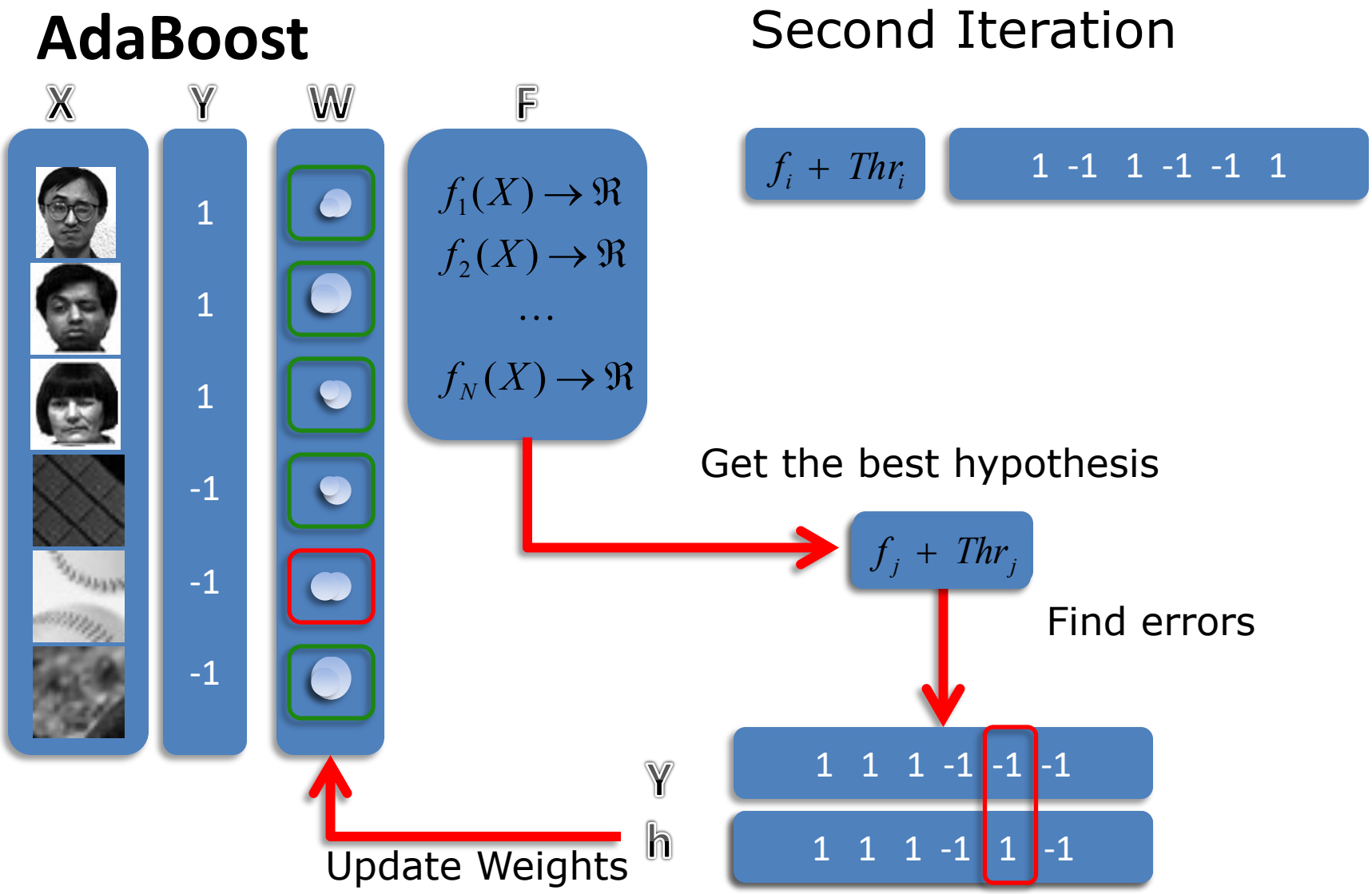
- Weak Learner
  - Find the weak hypothesis that minimizes the weighted error:

$$\varepsilon = \sum_{h(x_i) \neq y_i}^{N} \omega_i$$

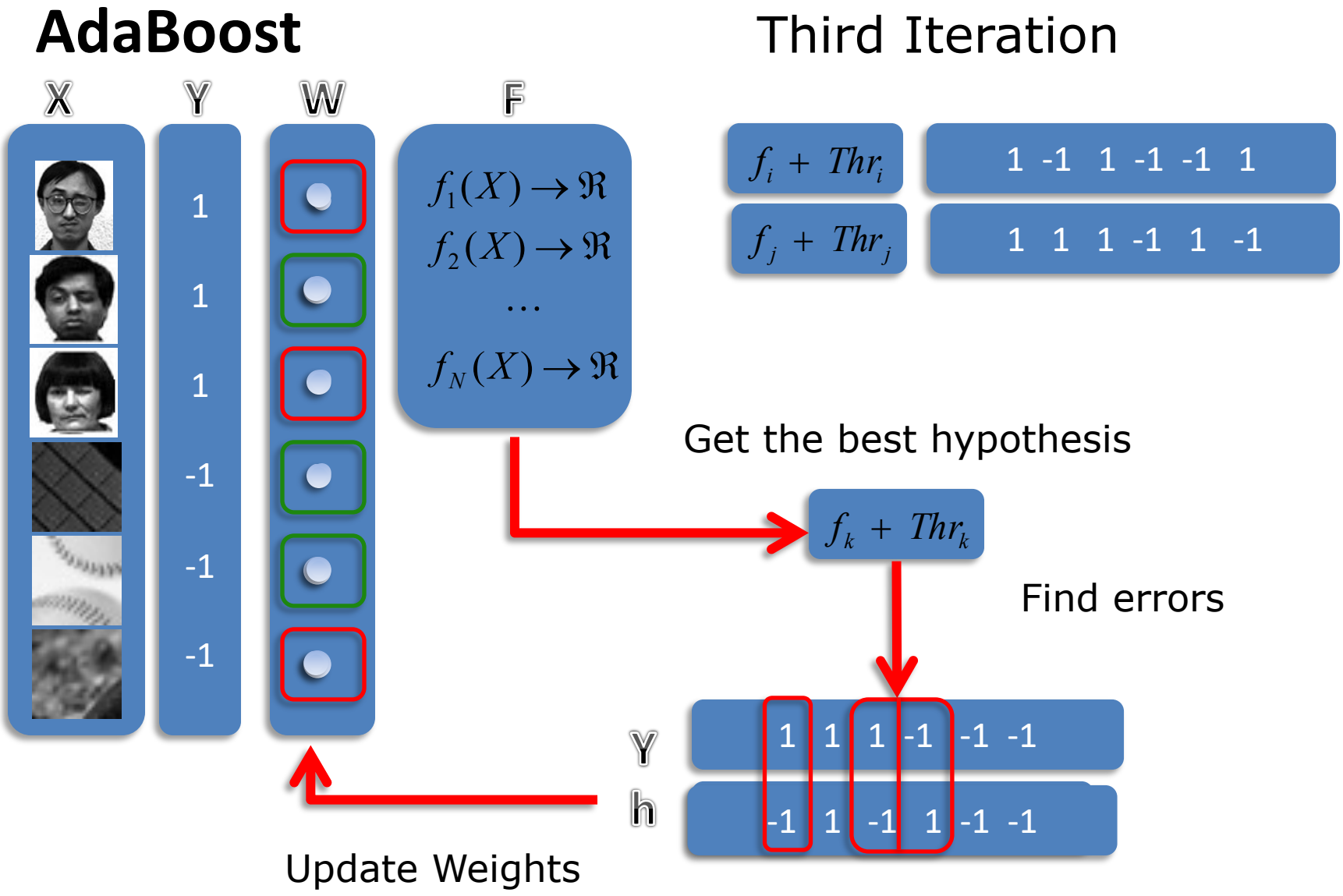
Face detection



Face detection

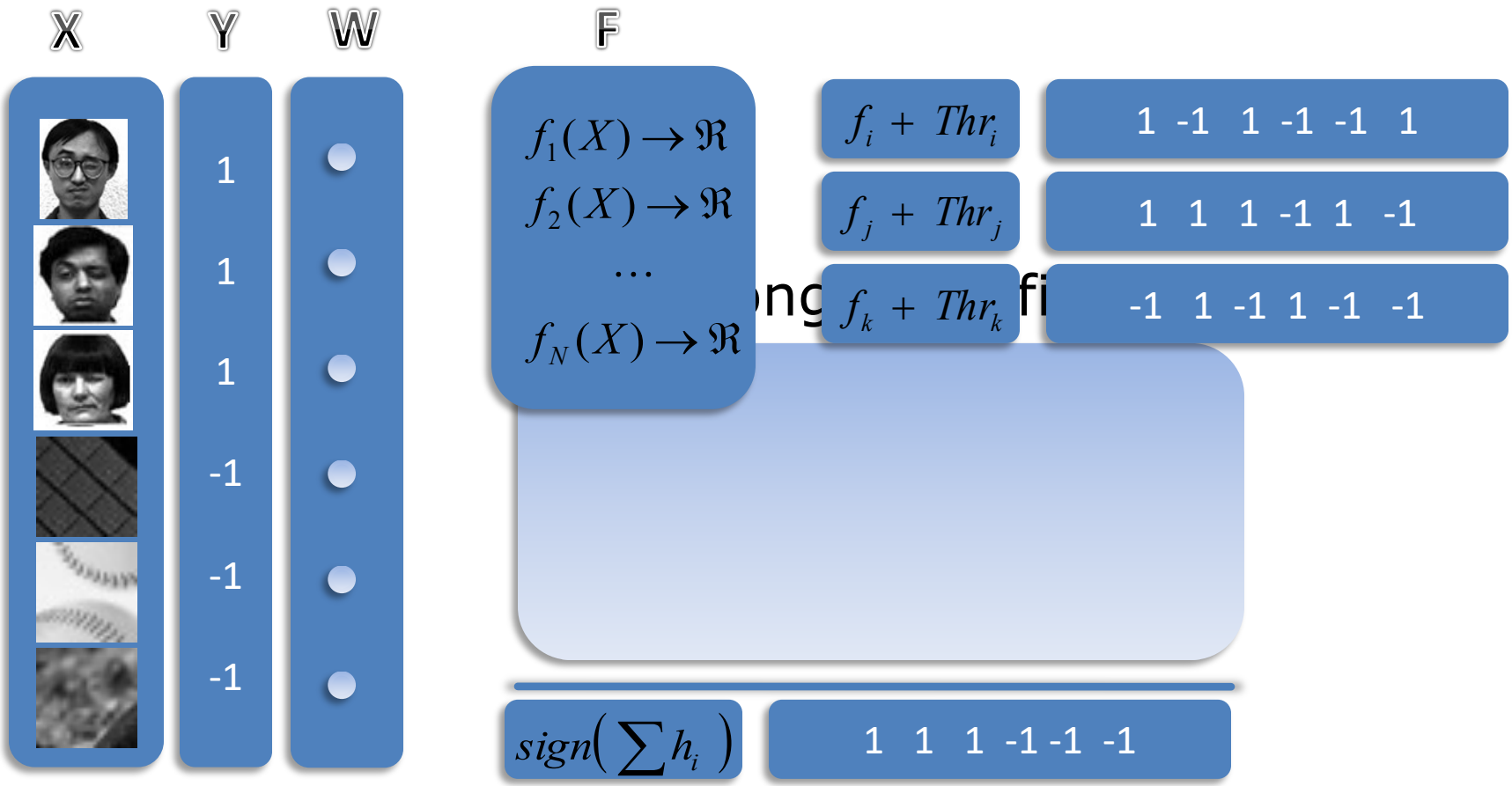


Face detection



Face detection

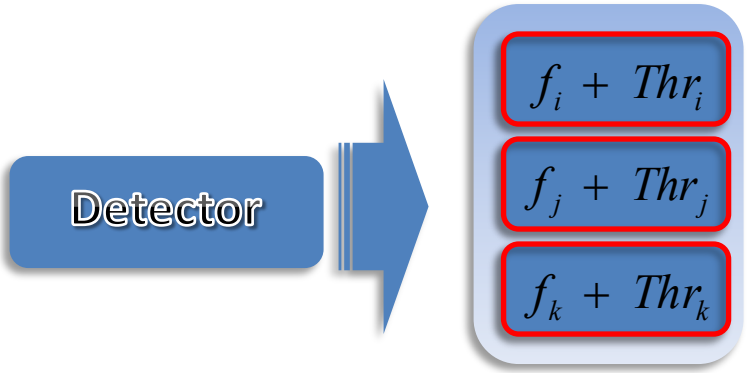
AdaBoost



Face detection

AdaBoost

Strong Classifier



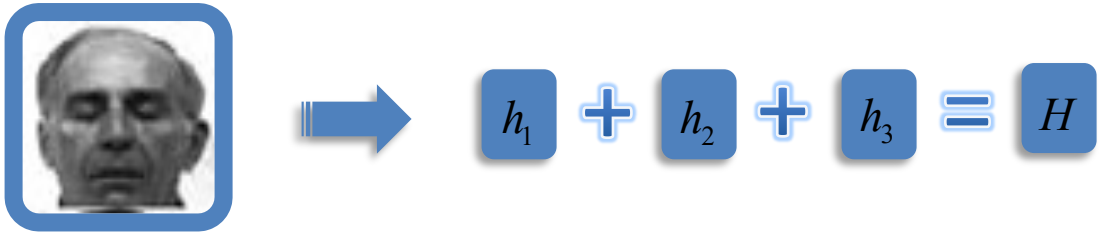
Hypothesis

$$h(X) = \begin{cases} +1 & if\ f(X) \geq Thr \\ -1 & elsewhere \end{cases}$$

Final decision

$$H(X) = sign\left(\sum_{i=1}^{m} h_i(X)\right)$$

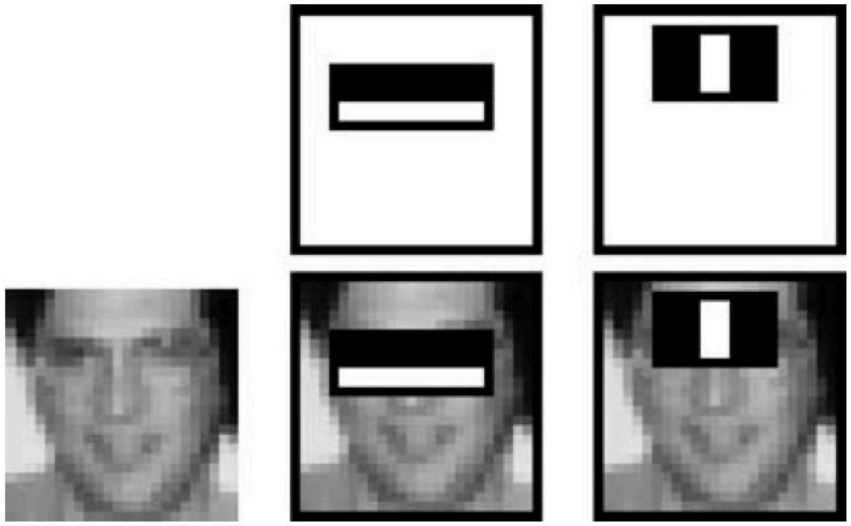
Input test image



Face detection

- After some experiments:

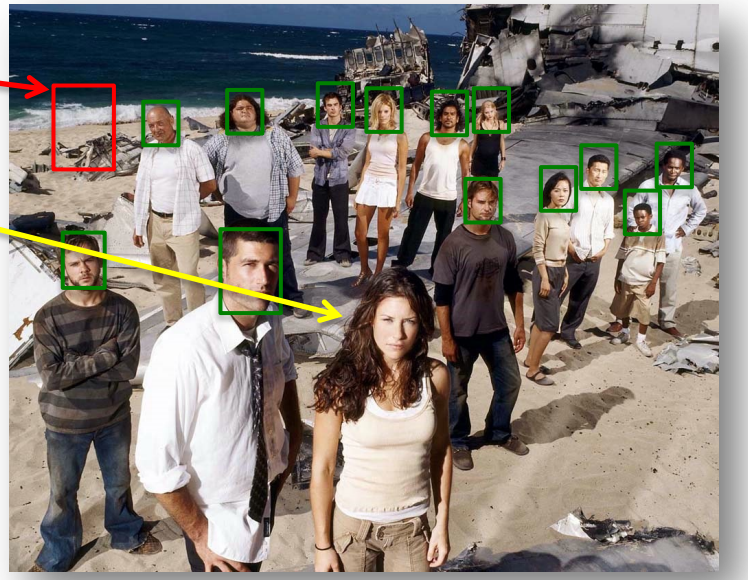
  The initial rectangle features selected by Adaboost are:



Face detection

In general, detection error is measured in:

- **False Positive** (FP)
- **False Negative** (FN)



Viola & Jones method accepts FP, but not a FN.

→**Faces can not be lost!!**

→**CASCADE**

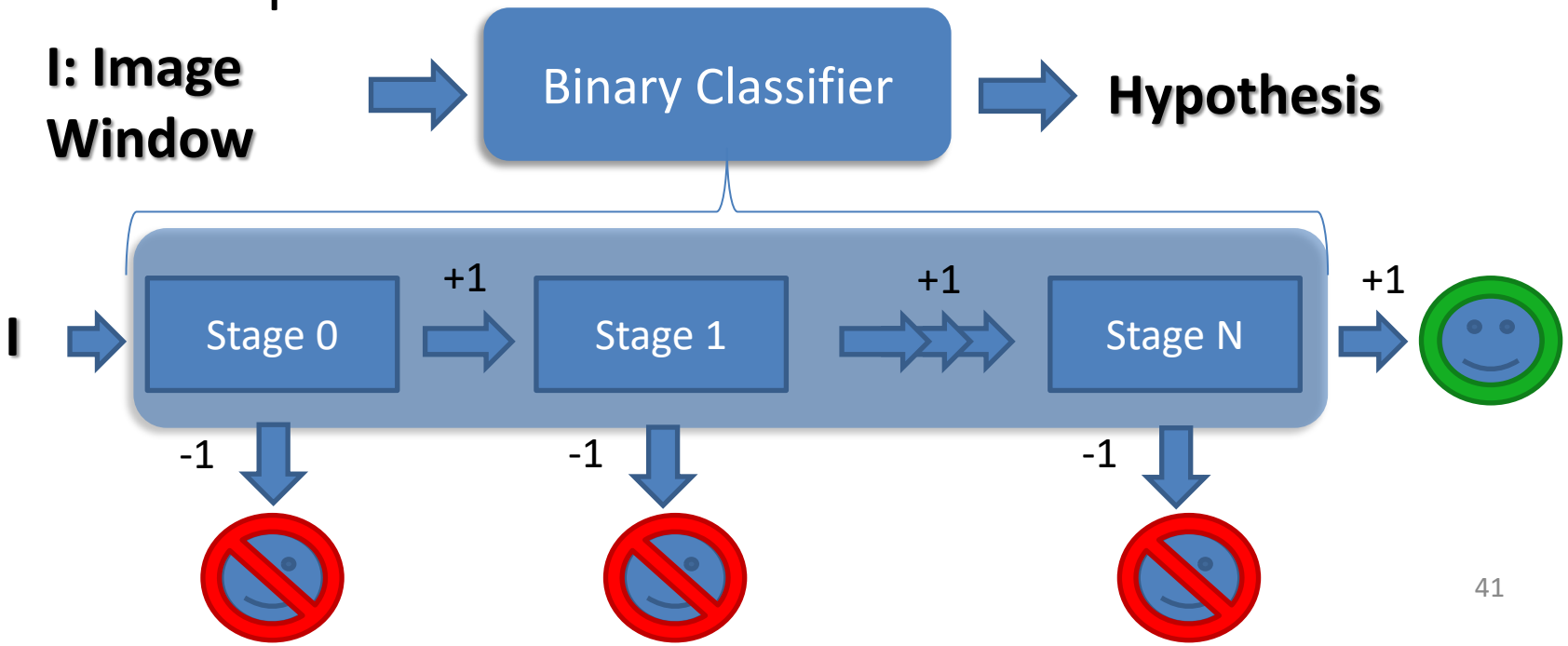
Face detection

Cascade of classifiers

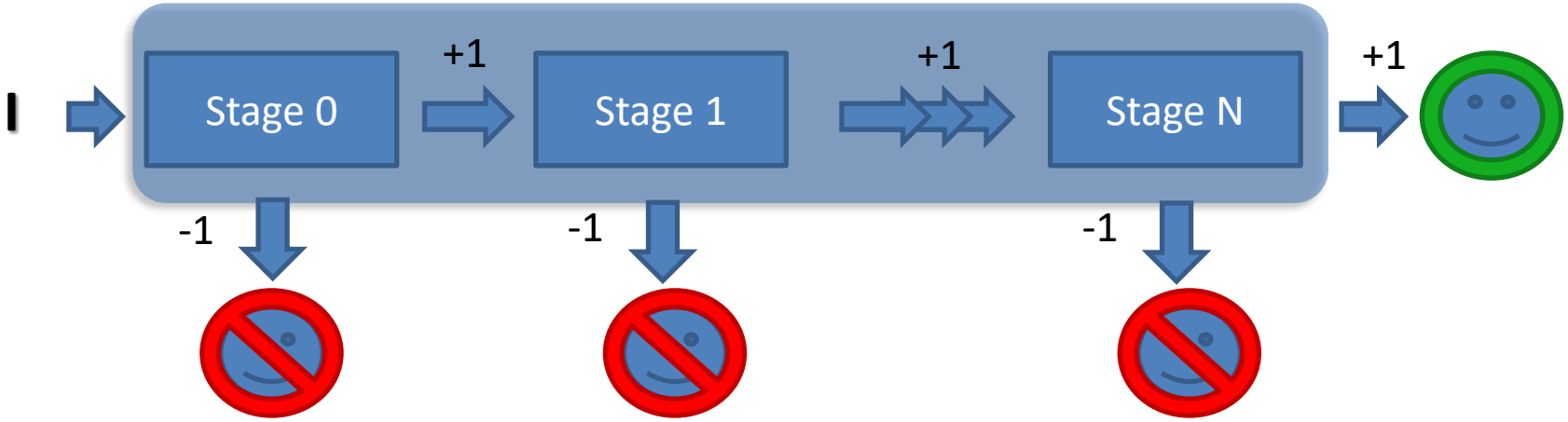
- Method to speed-up the detection process. Typical when windowing is used.
- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows

Face detection

- Each stage only process regions classified as faces by the previous stages.
  - Similarity between faces/non-faces regions increments each stage, incrementing the difficulty of the problem.



Face detection



Number of regions processed by each stage decrease exponentially:

#windows	Stage 0	Stage 1	Stage 5	Stage 10	Stage 15
FA = 0.5	1.000.000	500.000	31.250	976	30

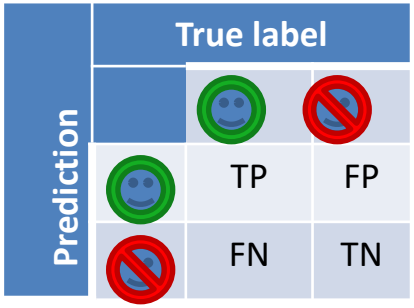
Face detection

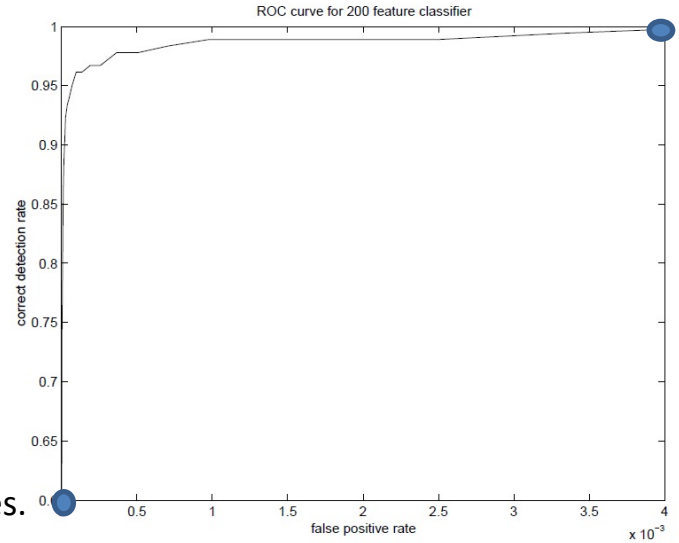
Basic evaluation concepts:

- Probability of detection (POD) or **Detection Rate** (DR):

  DR = TP/(TP+FN) = TP/P
- False alarm rate (FA) or **False Positive rate**:

  FA = FP/(TN+FP) = FP/N





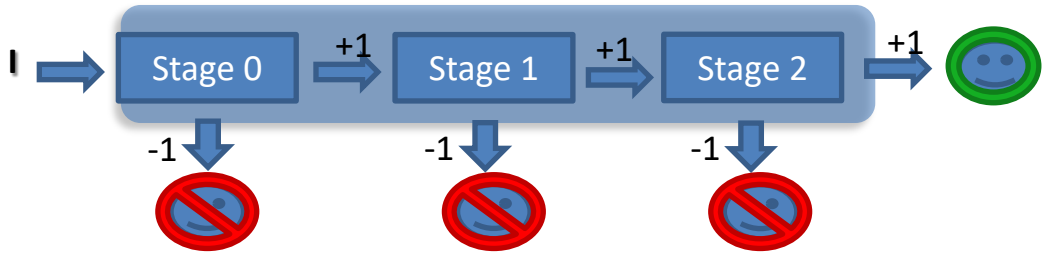
Prediction of all the samples as positives.

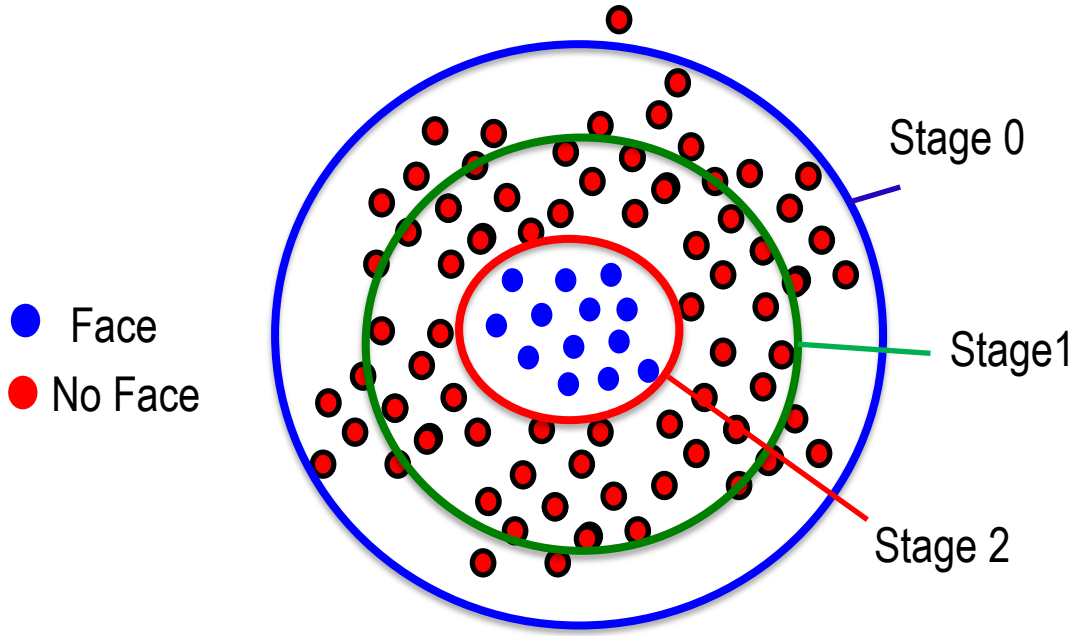
Prediction of all the samples as negatives.

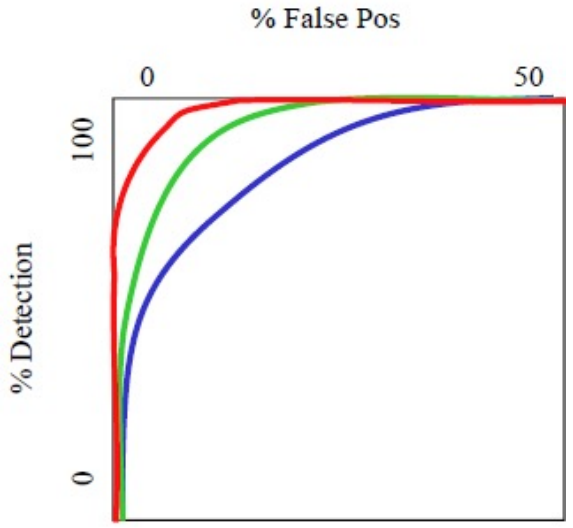
Receiver operating characteristic (ROC) curve

Face detection

- Chain classifiers that are progressively more complex and have lower false positive rates







Face detection

Criteria for cascade design:

1) Each classifier of the cascade is an AdaBoost
2) The first classifier C_1 is the simplest one
3) Following classifiers are more complex to refine the results of previous classifiers.

Face detection

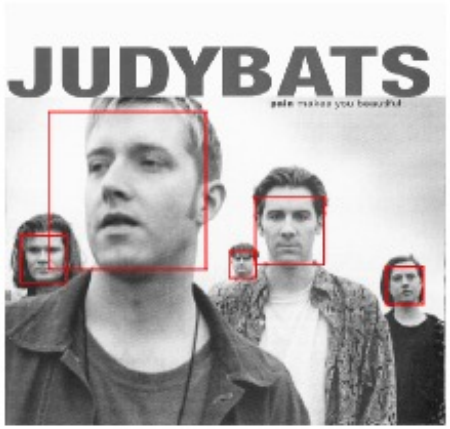
Training the cascade

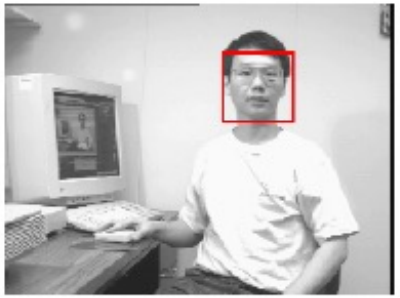
- Set target detection and false positive rates for each stage
- Keep adding features to the current stage until its target rates have been met
- Test on a *validation set*
- If the overall false positive rate is not low enough, then add another stage
- Use false positives from current stage as the negative training examples for the next stage

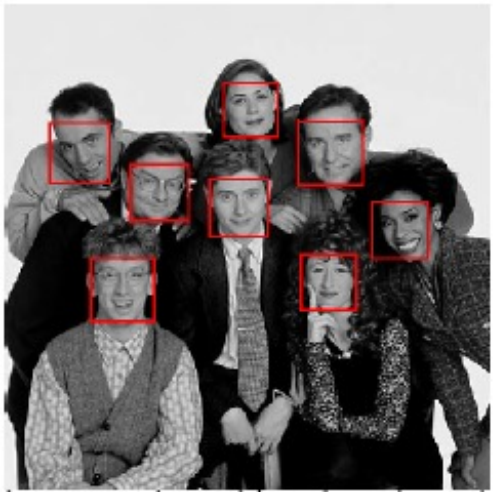
Face detection

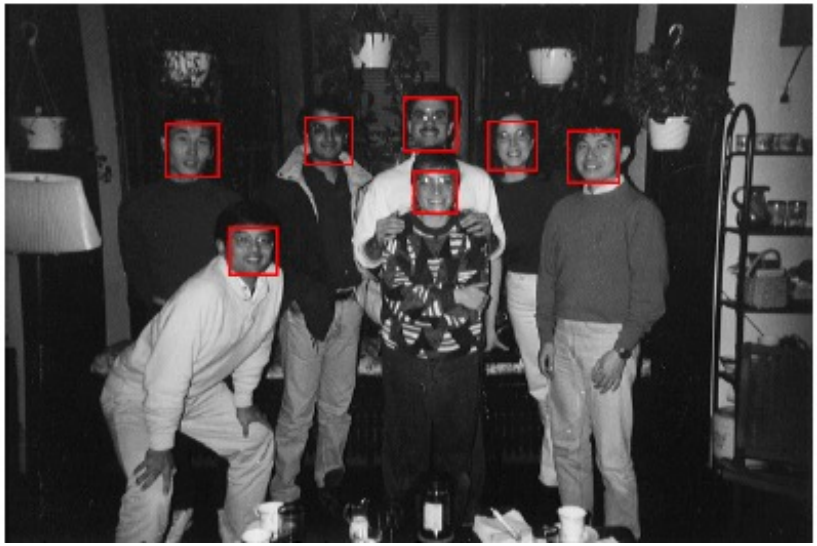
- Viola & Jones prepared their final Detector cascade:
  - 38 layers (stages), 6060 total features included
  - 1st classifier- layer, 2-features
    - 50% FP rate, 99.9% TP rate
  - 2nd classifier- layer, 10-features
    - 20% FP rate, 99.9% TP rate
  - Next 2 layers 25-features each, next 3 layers 50-features each and so on…
- Tested on the MIT+MCU test set
- Process a 384x288 image on a PC (dated 2001) took about 0.067 seconds

Results









Summary

- Viola & Jones algorithm is a method for automatic detection of faces in an image.
- **Rectangle Haar-like features** provide a description of the window features of the image
- By means of **integral images**, the rectangle features can be computed fast!
- Rectangle features are robust in front of noise
- **AdaBoost** is used for feature selection and classification
- Cascade of classifiers allows to obtain a very low false negative rate detecting faces at **real-time**!

Bibliography

Bibliography:

- Szelisky, “Computer Vision: algorithms and applications”.
- P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”.
- Proc. CVPR, 1:511-518, 2001. P. Viola and M. Jones: ”Robust Real-time Object Detection”, IJCV 2001.

- Article: https://realpython.com/traditional-face-detection-python/
- Videos:
  - http://www.youtube.com/watch?NR=1&v=lvBvFHEX-CY
  - http://www.youtube.com/watch?NR=1&v=JyBMxeVCQkc
  - http://videolectures.net/lmcv04_verri_clafa1/

Lecture videos

Face Detection

- From 762 - Intro Until 772 – End.

Face Recognition:

- From 735 - Eigenfaces Until 740 – End.
  - Except 738 - An Old Cast of Characters

From *Introduction to Computer Vision*:

https://www.udacity.com/course/introduction-to-computer-vision--ud810

COMPUTATIONAL VISION:
Face Detection