The Viola Jones algorithm for face detection

P. Viola, M. J. Jones, “Robust Real-Time Face Detection”, International Journal of Computer Vision 57(2), 137–154, 2004

Three main ideas:

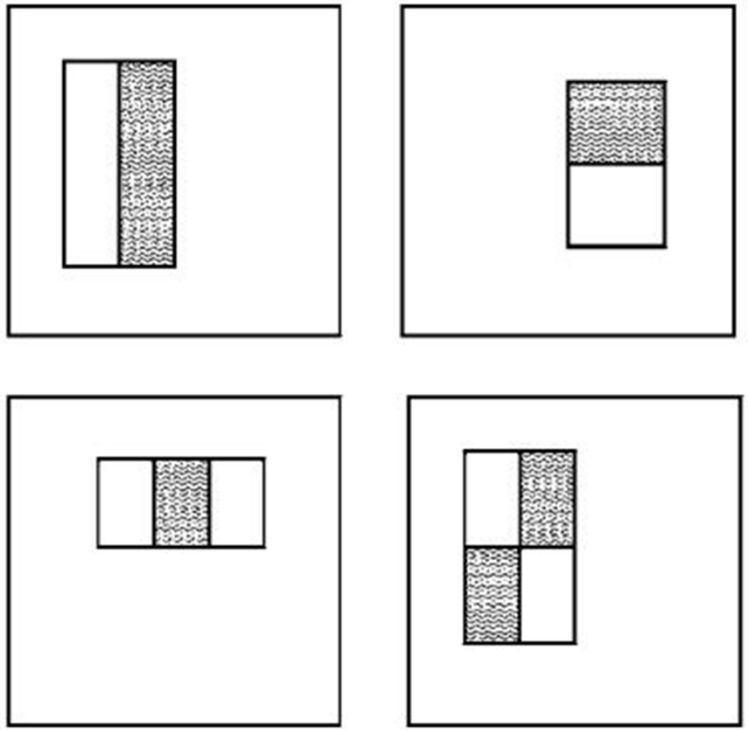
- introduction of a new image representation called the *Integral Image*
- simple and efficient classifier which is built using the *AdaBoost* learning algorithm
- method for combining classifiers in a “cascade” which allows background regions of the image to be quickly discarded

Operating on 384x288 pixel images, faces
are detected at 15 frames per second on a 700 MHz Intel Pentium III

Features

The simple features used are reminiscent of Haar basis functions

- *two-rectangle feature*: difference between the sum of the pixels within two rectangular regions
- *three-rectangle feature*: sum within two outside rectangles subtracted from the sum in a center rectangle
- *four-rectangle feature*: difference between diagonal pairs of rectangles.



Given that the base resolution of the detector is 24×24, the exhaustive set of features is quite large, 160,000

- a very efficient way to compute them is needed -> *integral image*
- only the useful ones have to be used -> *AdaBoost*

Integral image

The *integral image* at location *x* , *y* contains the sum of the pixels above and to the left of *x* , *y*, inclusive:

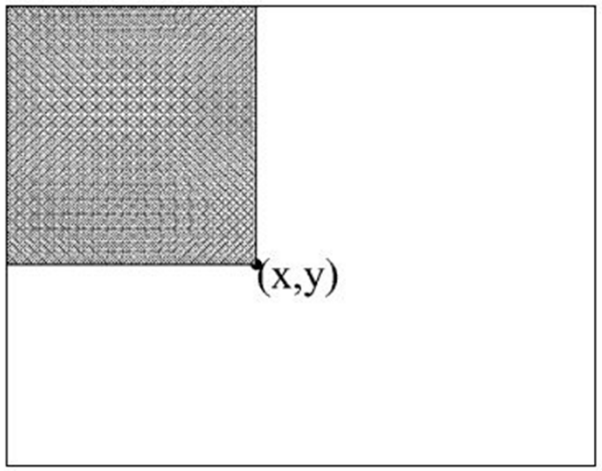
$$ii(x, y) = \sum_{x' \le x, y' \le y} i(x', y'),$$

Using the following pair of recurrences:

$$s(x, y) = s(x, y - 1) + i(x, y)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

where *s(x , y)* is the cumulative row sum, the integral image can be computed in one pass over the original image.

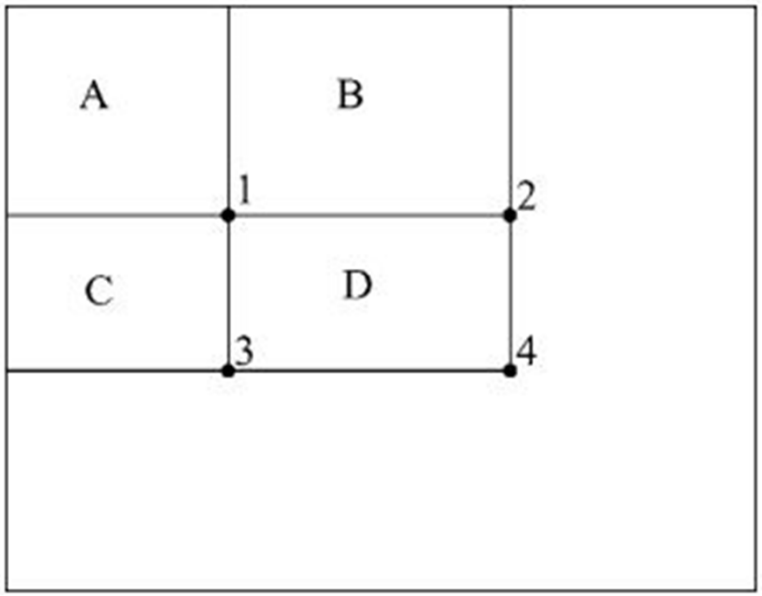


Integral image

The sum of the pixels within rectangle D can be computed with four array references:

- The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D.
- The sum within D can be computed as 4 + 1 – (2 + 3).

A two-rectangle feature can be computed in six array references – *for any scale*!

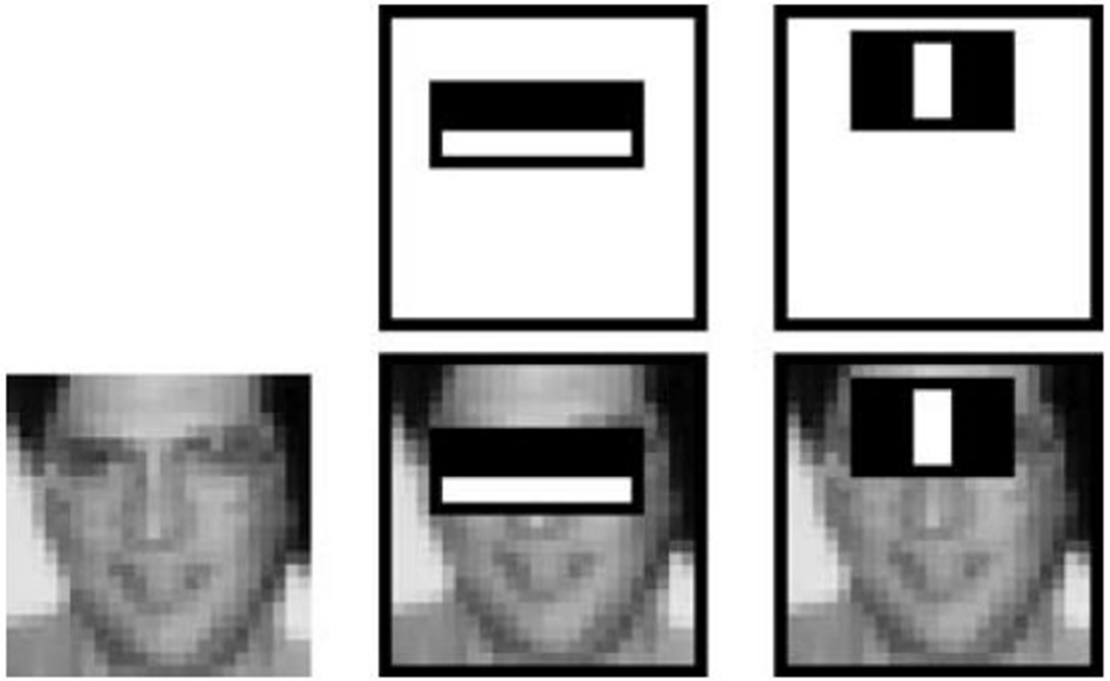


A very small number of these features can be combined to form an effective classifier. The main challenge is to find these features.

Avariant of AdaBoost is used both to select the features and to train the classifier.

The first two features selected by AdaBoost for the task of face detection are easily interpreted:

- first feature: the region of the eyes is often darker than the region of the nose and cheeks
- second feature: the eyes are darker than the bridge of the nose.



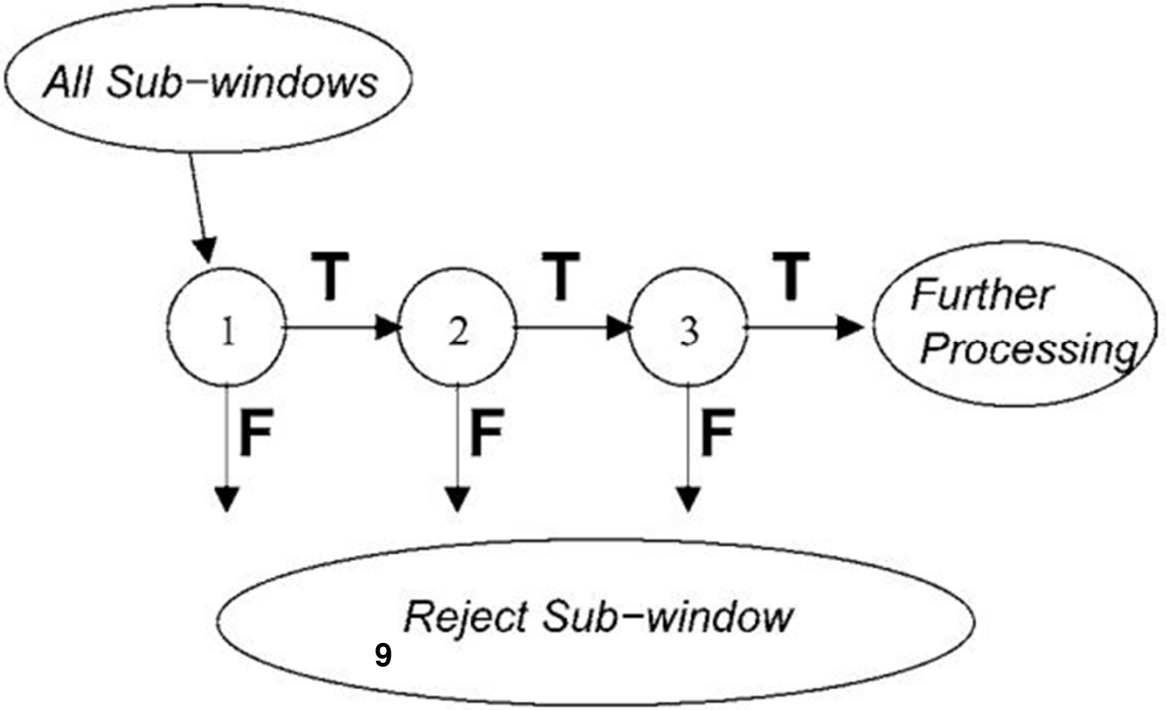
So, the systems is essentially based *on correlation*!

Cascaded classifier

Smaller, and therefore more efficient, classifiers can be constructed which reject many of the negative subwindows while detecting almost all positive instances:

- simpler classifiers are used to reject the majority of subwindows
- then, more complex classifiers are called upon to achieve low false positive rates.

T: true
F: false



The cascaded classifier for face detection

… has 38 layers and 6060 features:

- first classifier: two features, rejects about 50% of non-faces (while correctly detecting close to 100% of faces)
- second classifier: ten features, rejects 80% of non-faces
- 3rd and 4th: 25 features
- …

Training

- Base resolution: 24x24 pixels
- several thousands faces and non-faces
- training time: *weeks* on a 466 MHz AlphaStation XP900

Position and scale invariance

The final detector is scanned across the image *at multiple scales and locations*:

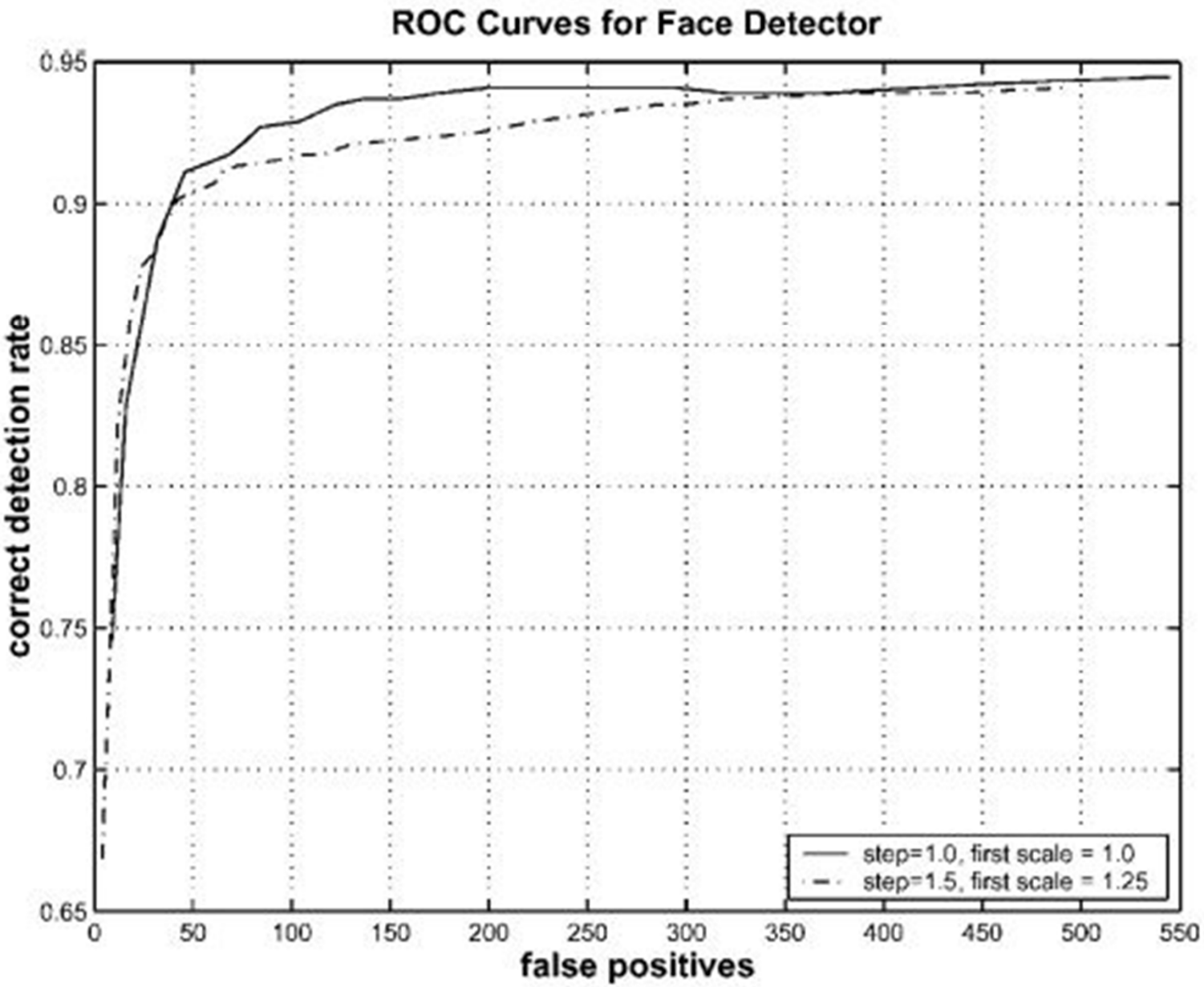
- Scaling is achieved by scaling the detector itself, rather than scaling the image (features can be evaluated at any scale with the same cost)
- The detector is also scanned across location, by shifting the window some number of pixels

Preprocessing

- Sub-windows were variance normalized during training to minimize the effect of different lighting conditions.
- The same is done during detection as well.

Final postprocessing

- It is useful to postprocess the detected sub-windows in order to combine overlapping detections into a single detection.


ROC Curves for Face Detector
correct detection rate
false positives
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0
50
100
150
200
250
300
350
400
450
500
550
step=1.0, first scale = 1.0
step=1.5, first scale = 1.25

Examples of output

Failure modes

Various failure modes

- The face detector can detect faces that are tilted up to about ± 15 degrees in plane and about ± 45 degrees out of plane (toward a profile view).
- Problems with harsh backlighting and occlusions

Final comments

- The same approach can be used to detect other objects (new training is needed of course)
- In pattern recognition it is normally a good idea to have smart features. Here,
  - features are *very simple*
  - but they are *very numerous*, and the classifier is *very smart*