Robust Real-Time Face Detection

Viola&Jones, Int. J. of Comp. Vision 2004

France ROSE, Mar 2017

Overview

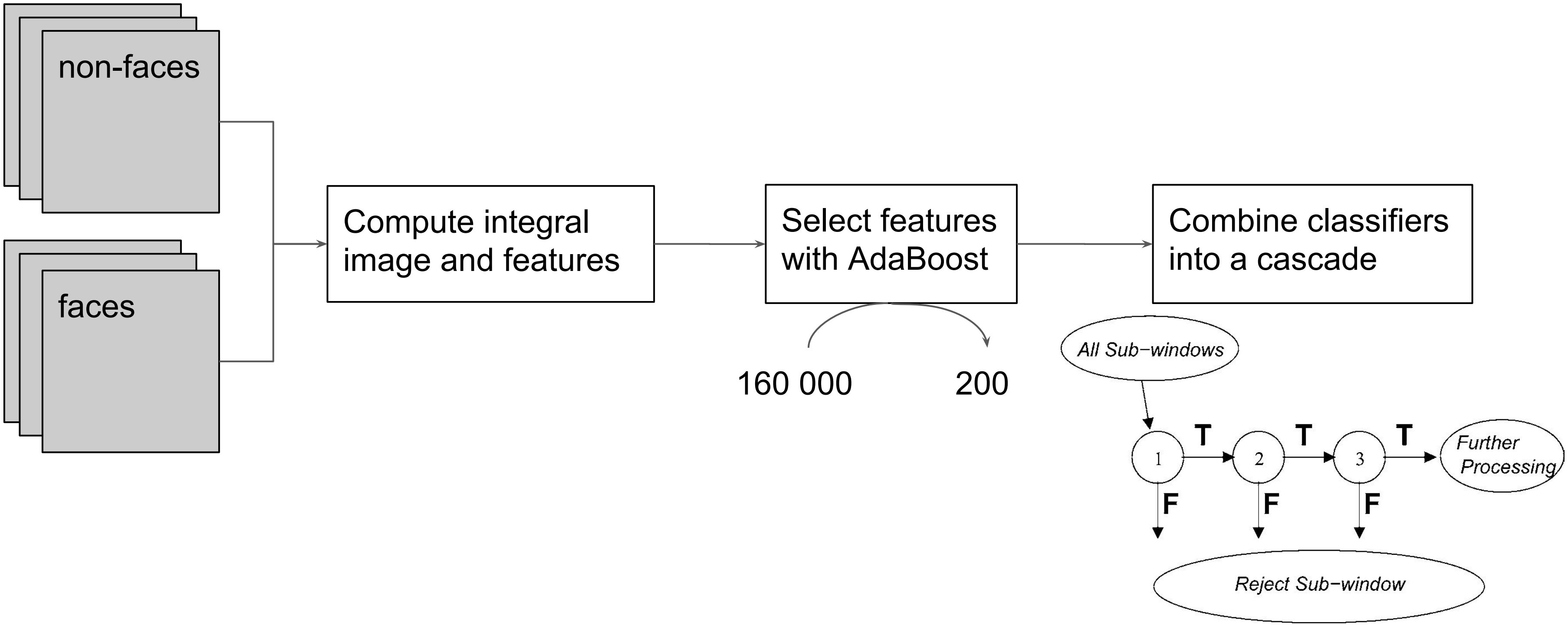
Goal:

- Find faces in an image and put boxes around them
- Do it “real-time”
- Achieve a very low False Detection Rate

3 major contributions:

- Integral image to compute quickly features on the image
- Features filtering by AdaBoost learning algorithm
- “Cascade” of classifiers to reject early the easy boxes with no face

Overview


non-faces
faces
Compute integral image and features
Select features with AdaBoost
160 000
200
Combine classifiers into a cascade
All Sub−windows
1
2
3
T
T
T
F
F
F
Further Processing
Reject Sub−window

Ensemble methods

Basic idea: put many models instead of only one

Bagging (bootstrap aggregating):

- generate random variations of the training set by resampling
- learn a classifier on each
- combine the results by voting

Ensemble methods

Basic idea: put many models instead of only one

Bagging

Boosting:

- Weight training examples
- Weights are varied so that each new classifier focuses on the examples the previous ones tended to get wrong (sequential)

Ensemble methods

Basic idea: put many models instead of only one

Bagging

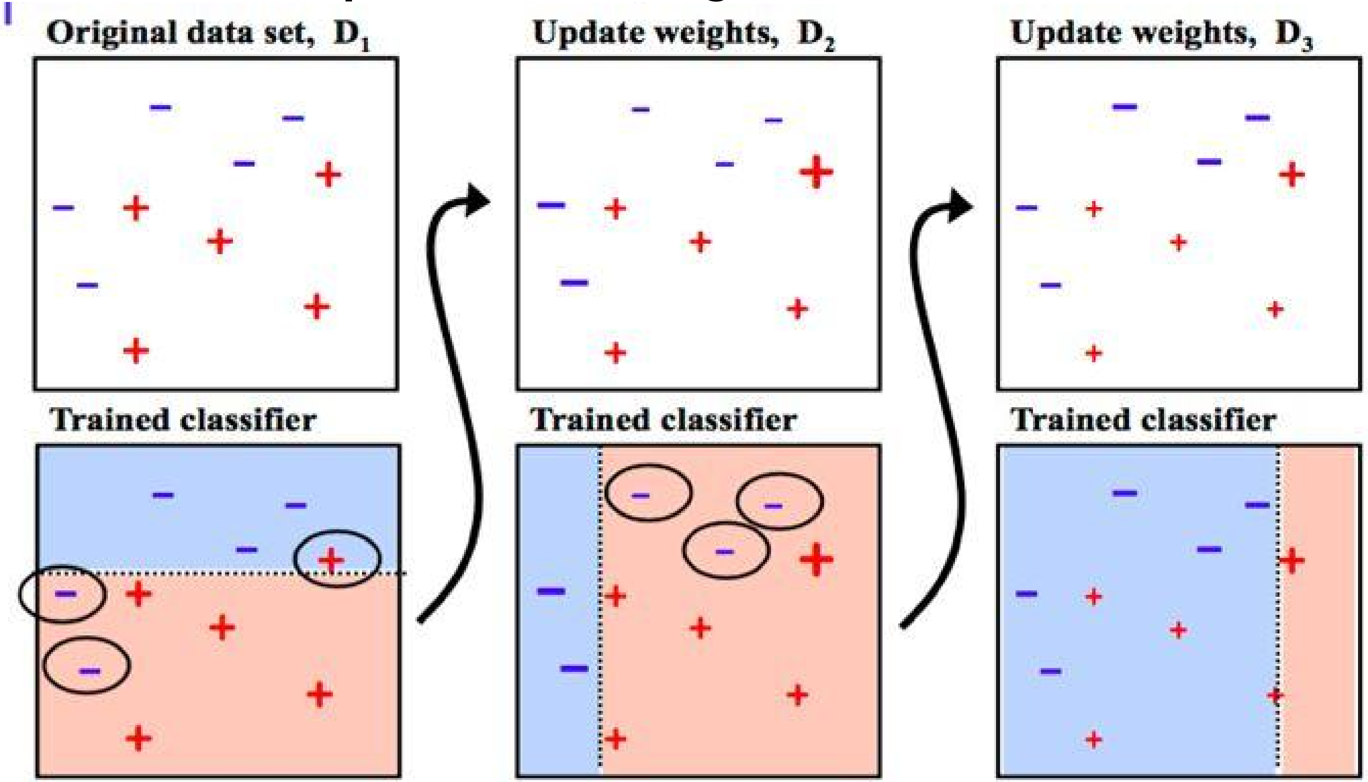
Boosting

Stacking:

- The outputs of individual classifiers become the input of a “higher level” learner that figures out how to best combine them

AdaBoost: adaptive boosting

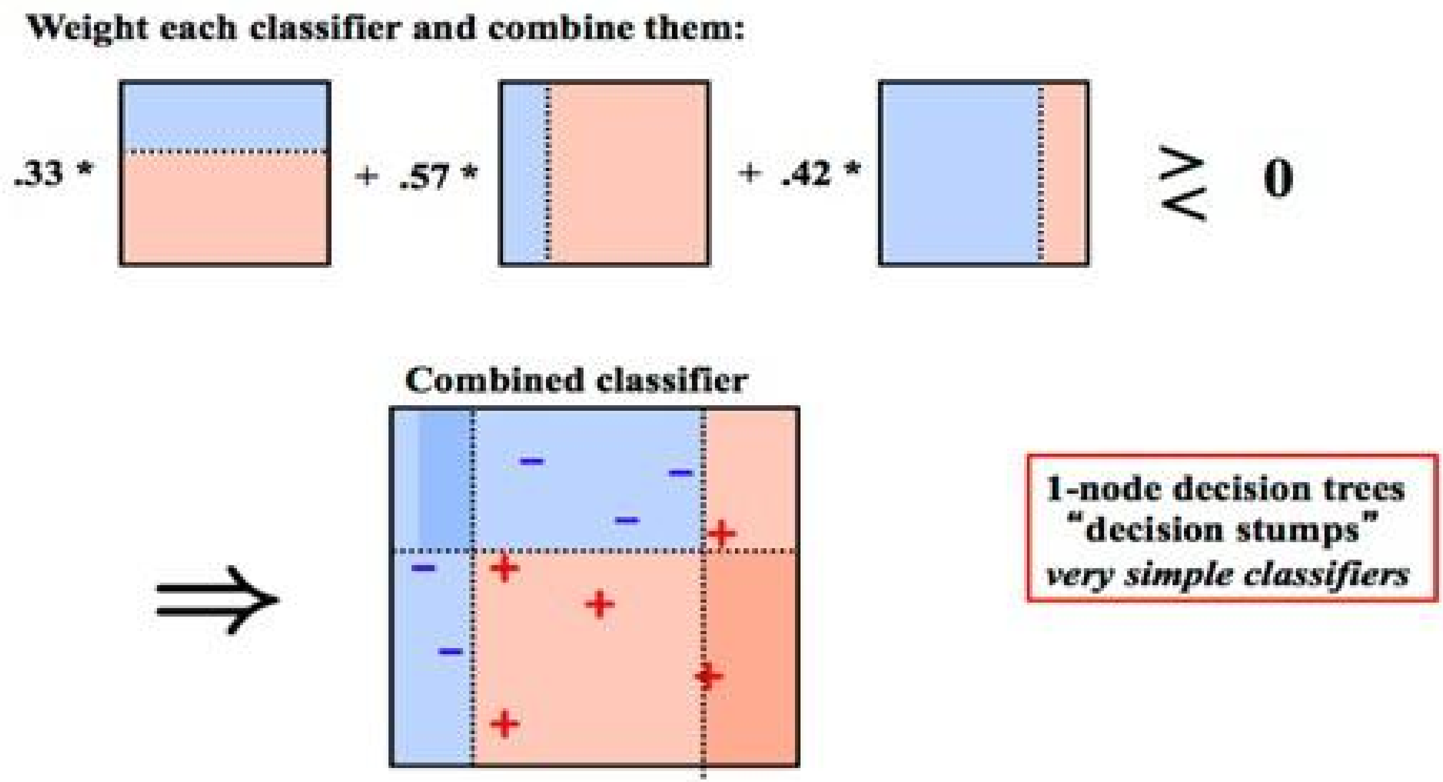
Weak classifier: doing better than random



courtesy to Alexander Ihler http://sli.ics.uci.edu/Classes/2012F-273a?action=download&upname=10-ensembles.pdf

Slide from Zhang&Khalil

AdaBoost



Slide from Zhang&Khalil

AdaBoost

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:

1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
2. Select the best weak classifier with respect to the weighted error

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i \,|\, h(x_i, f, p, \theta) - y_i \,| \,.$$

3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, p_t, and θ_t are the minimizers of ϵ_t.
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

AdaBoost

- Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \ldots, T$:

1. Normalize the weights, $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$
2. Select the best weak classifier with respect to the weighted error

$$\epsilon_t = \min_{f,p,\theta} \sum_i w_i \, | h(x_i, f, p, \theta) - y_i | .$$

3. Define $h_t(x) = h(x, f_t, p_t, \theta_t)$ where f_t, p_t, and θ_t are the minimizers of ϵ_t.
4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

Train

Predict & compute errors

Update weights of training examples

- The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Importance of each weak classifier related to its error

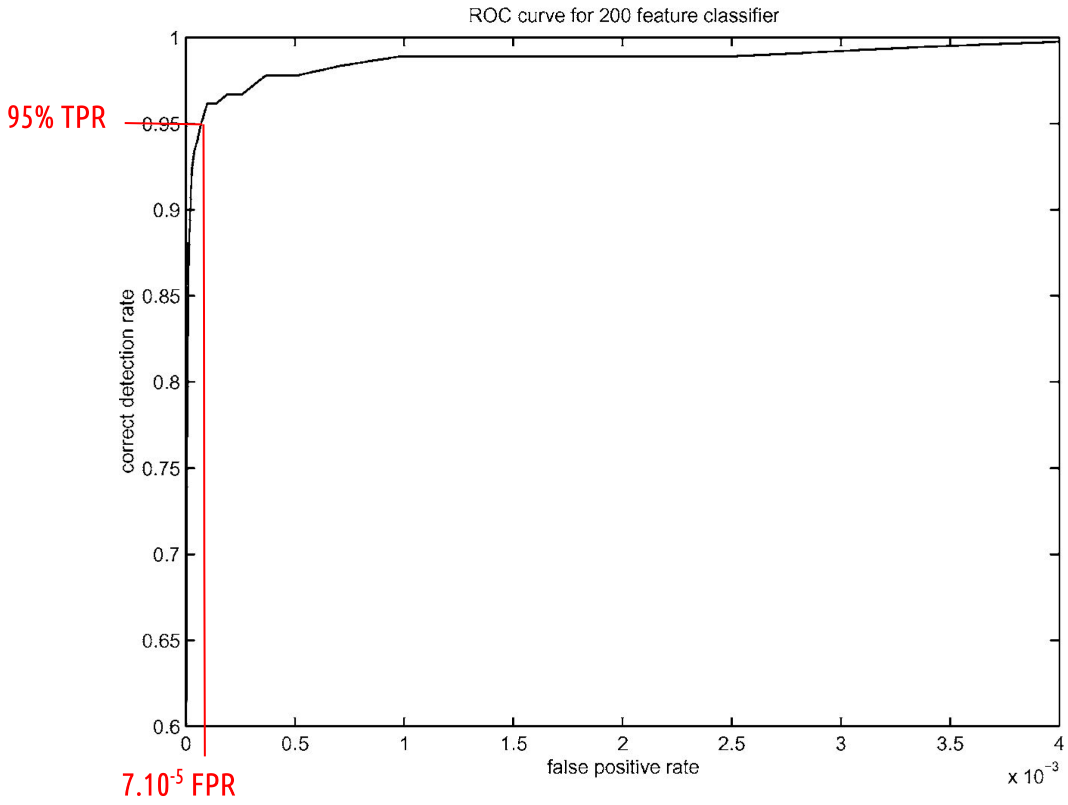
AdaBoost as a feature selection process

- Each weak learner is restricted to **one feature**
- A selected feature(= a good weak classifier) will have a **large weight** in the strong classifier
- "For each feature, the weak learner determines the **optimal threshold classification function**, such that the minimum number of examples are misclassified."

$$h(x, f, p, \theta) = \begin{cases} 1 & \text{if } pf(x) < p\theta \\ 0 & \text{otherwise} \end{cases}$$

x: subwindow
f: feature
p: polarity (+/-1)
Θ: threshold

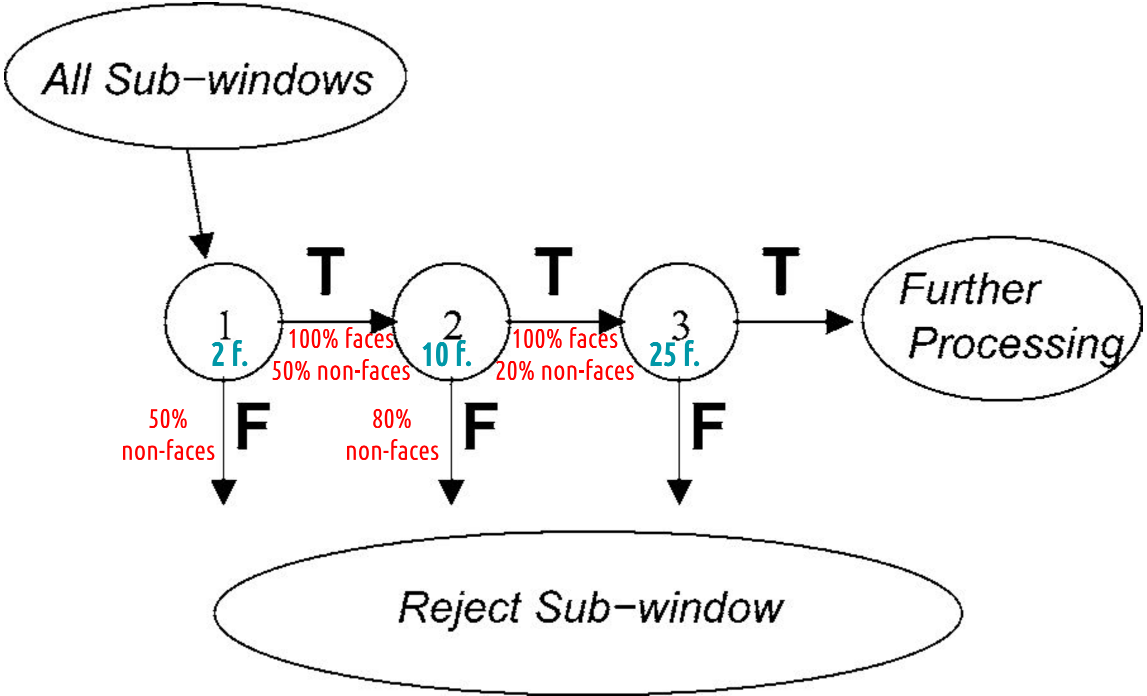
A 200-features strong classifier from AdaBoost



Objective: 10^{-6} FPR

The Attentional Cascade

“Smaller [...] boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances.”



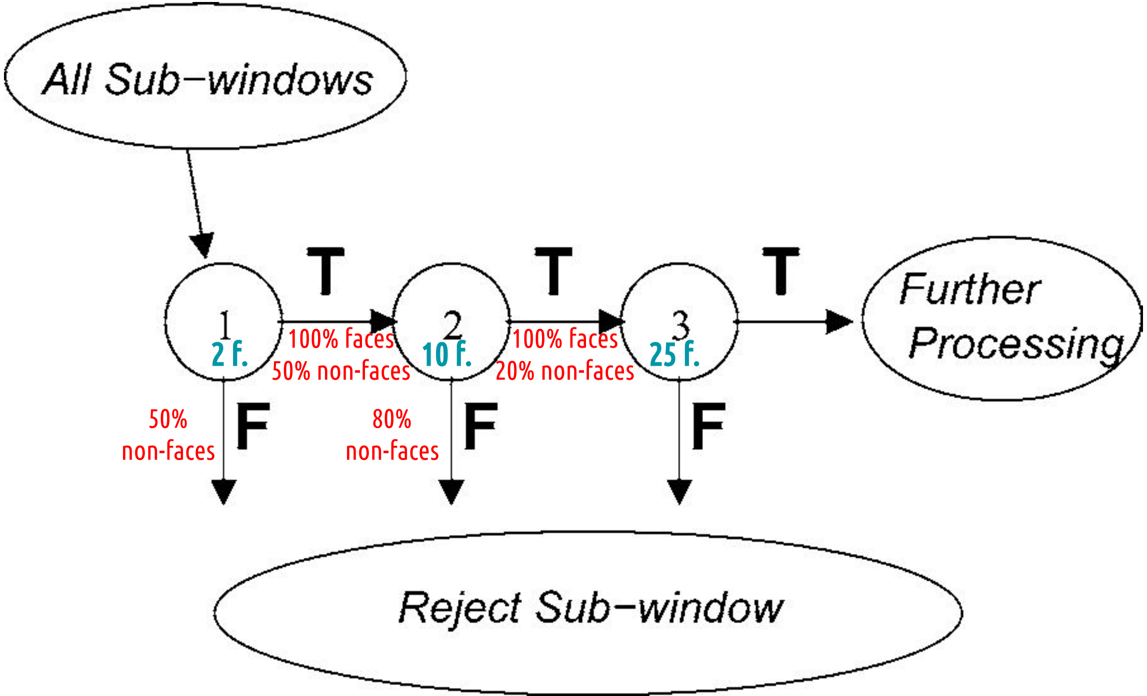
Each stage (1,2,3) is a classifier constructed by AdaBoost

Number of features in each classifier

Percentage of images presented to each classifier that are accepted/rejected by the classifier

The Attentional Cascade

"Smaller [...] boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances."



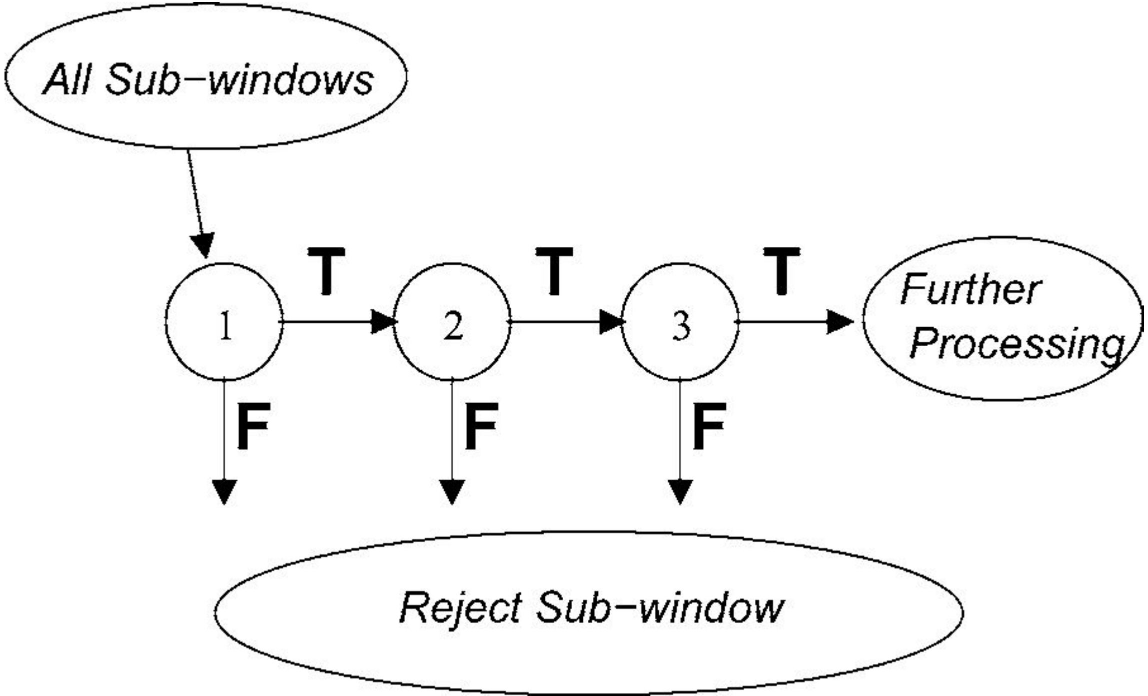
Expected number of evaluated features:

$$N = n_0 + \sum_{i=1}^{K} \left(n_i \prod_{j<i} p_j \right)$$

n_i: number of evaluated features of the i^{th} stage
p_j: positive rate of the i^{th} classifier

The Attentional Cascade

“Smaller [...] boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances.”



For a 10-stages cascade:

A **Detection Rate** of each stage of 0.99 will give 0.99^{10} = **0.9**

A **False Positive Rate** of each stage of 0.30 will give 0.30^{10}= **6.10^{-6}**

Final training code

- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User selects target overall false positive rate, F_{target}.
- P = set of positive examples
- N = set of negative examples
- $F_0 = 1.0$; $D_0 = 1.0$
- $i = 0$
- while $F_i > F_{target}$
  - $i \leftarrow i + 1$
  - $n_i = 0$; $F_i = F_{i-1}$
  - while $F_i > f \times F_{i-1}$
    * $n_i \leftarrow n_i + 1$
    * Use P and N to train a classifier with n_i features using AdaBoost
    * Evaluate current cascaded classifier on validation set to determine F_i and D_i.
    * Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$ (this also affects F_i)
  - $N \leftarrow \emptyset$
  - If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N

Final training code

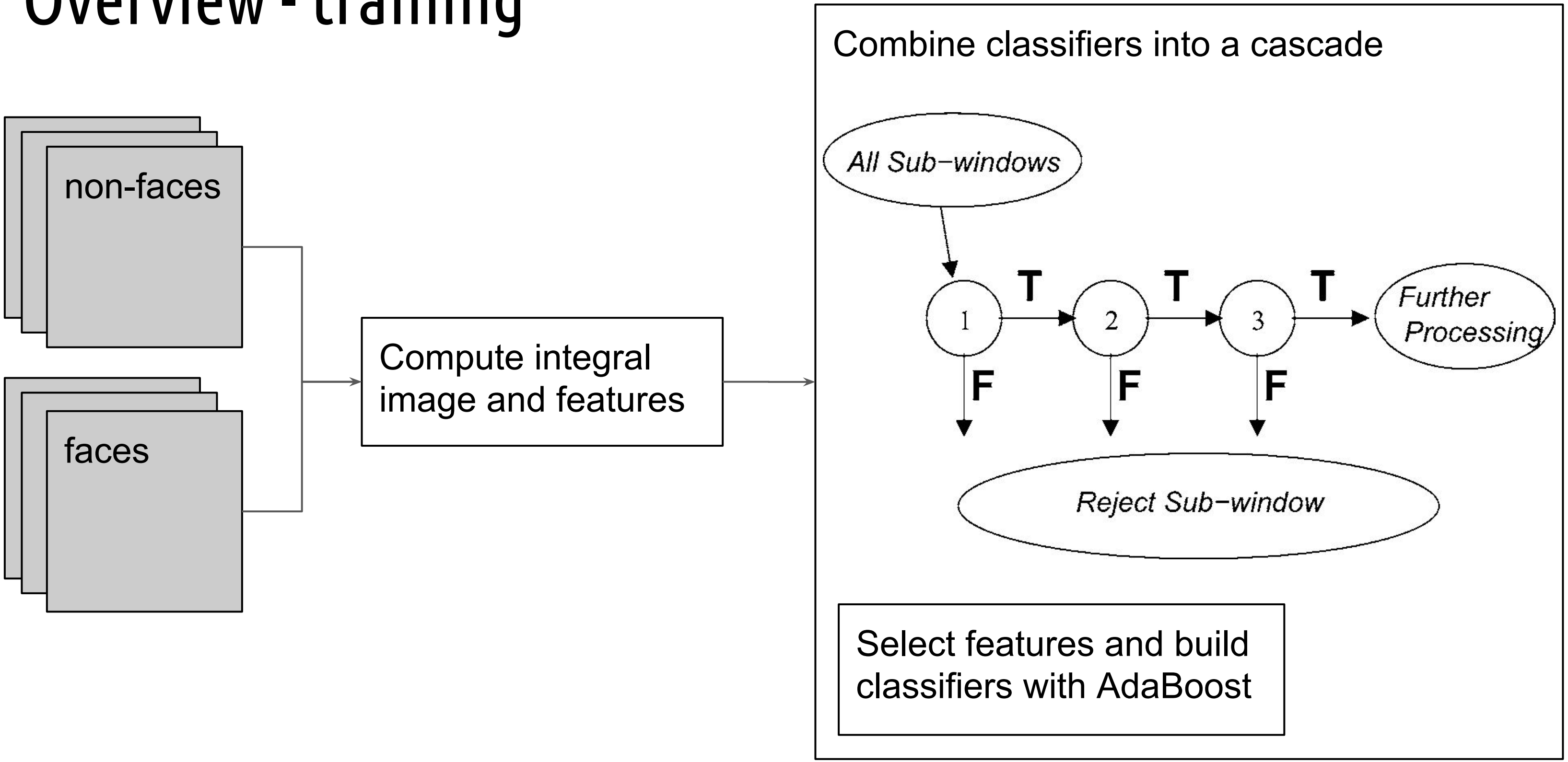
- User selects values for f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer.
- User selects target overall false positive rate, F_{target}.
- P = set of positive examples
- N = set of negative examples
- $F_0 = 1.0$; $D_0 = 1.0$
- $i = 0$
- while $F_i > F_{target}$
  - $i \leftarrow i + 1$
  - $n_i = 0$; $F_i = F_{i-1}$
  - while $F_i > f \times F_{i-1}$
    * $n_i \leftarrow n_i + 1$
    * Use P and N to train a classifier with n_i features using AdaBoost
    * Evaluate current cascaded classifier on validation set to determine F_i and D_i.
    * Decrease threshold for the ith classifier until the current cascaded classifier has a detection rate of at least $d \times D_{i-1}$ (this also affects F_i)
  - $N \leftarrow \emptyset$
  - If $F_i > F_{target}$ then evaluate the current cascaded detector on the set of non-face images and put any false detections into the set N

Initialization

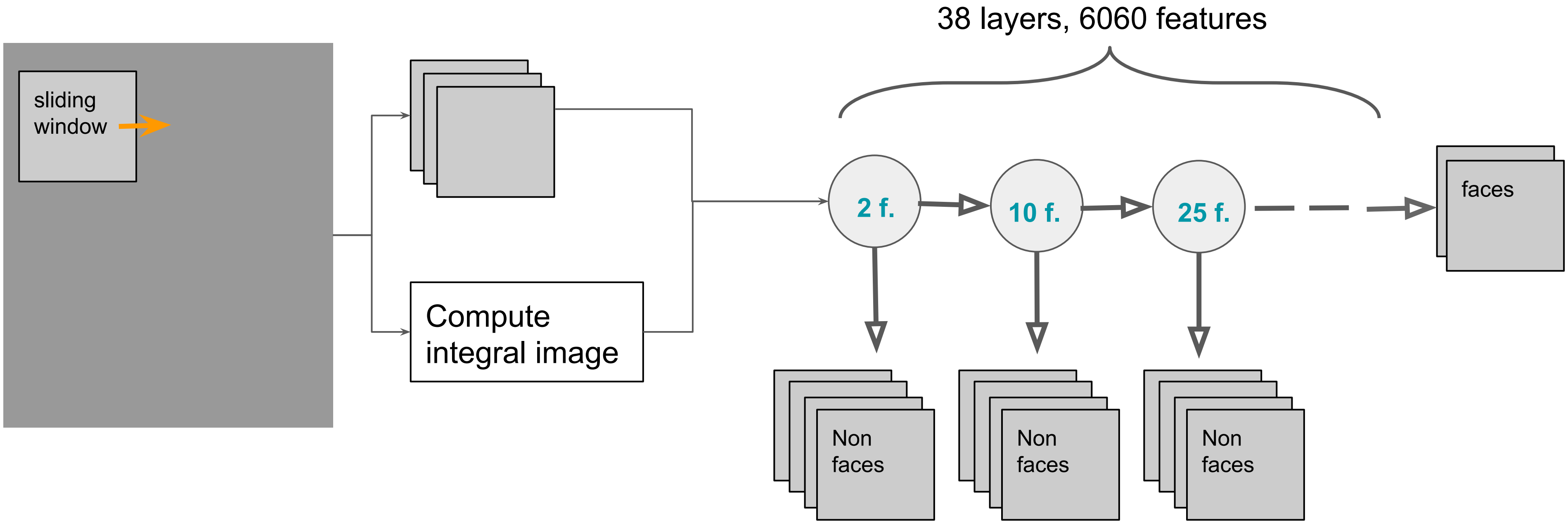
Adding a layer

Adding a feature to the layer:
- More features achieve higher DR and lower FPR
- More features require more time to compute

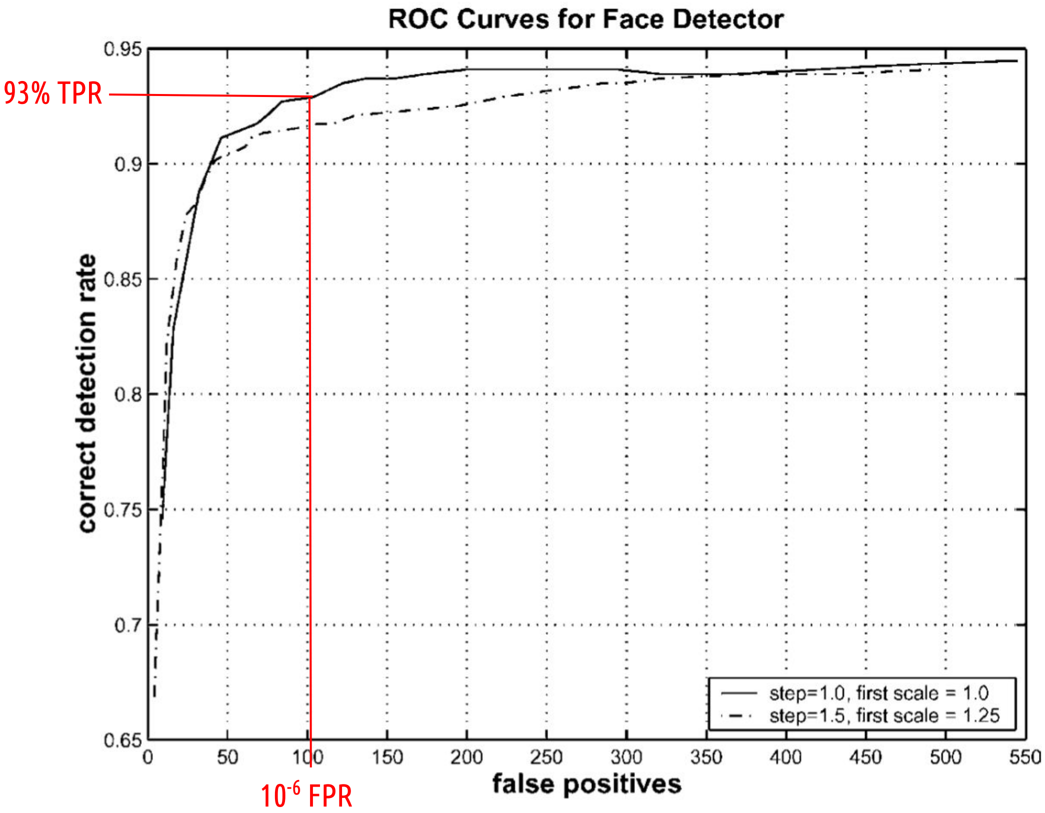
Overview - training



Overview - detection



Results


ROC Curves for Face Detector
93% TPR
10^{-6} FPR
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

References

- Viola and Jones. Robust Real-time Face Detection. International Journal of Computer Vision, 2004.
- Introduction to Adaboost, by Yongli Zhang Nacer Khalil (slides).
- Youtube video on AdaBoost (Alexander Ihler)
- Youtube video on Face Detection
- Classifier ensembles, by Pier Luca Lanzi (slides)