Design and implementation of a face detection and location technique to crop faces from human images DB

Michal Szabo
Aleix Granero Mol

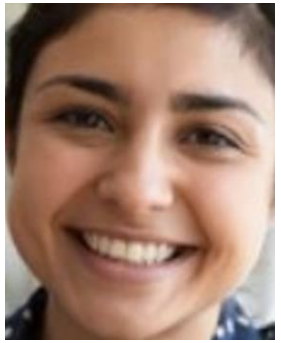
Contents

Introduction

Images database

Detect and crop faces

Cropped faces database

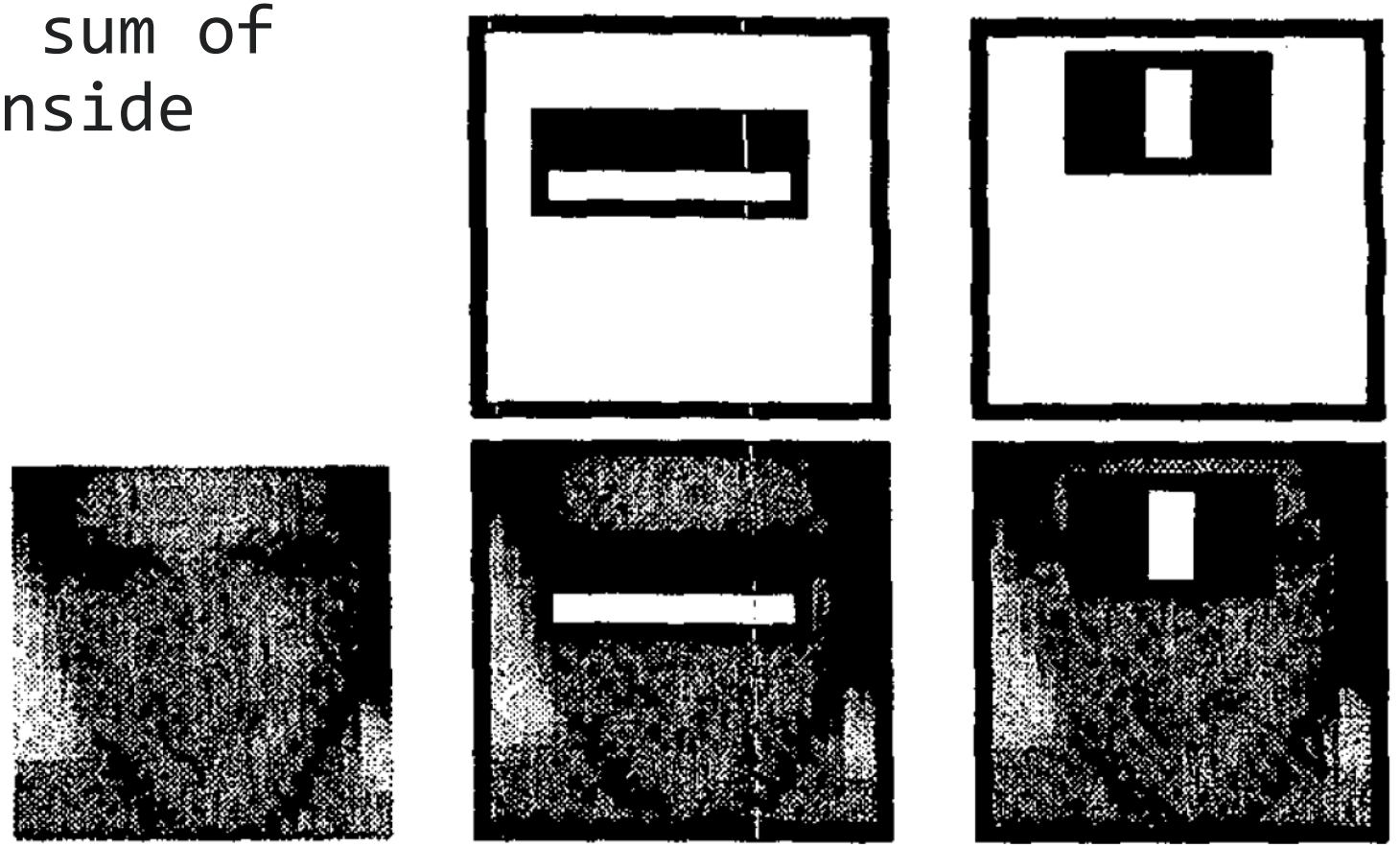


Viola-Jones Algorithm

- The Viola–Jones object detection framework is an object detection framework which was proposed in 2001 by Paul Viola and Michael Jones.

Haar Features

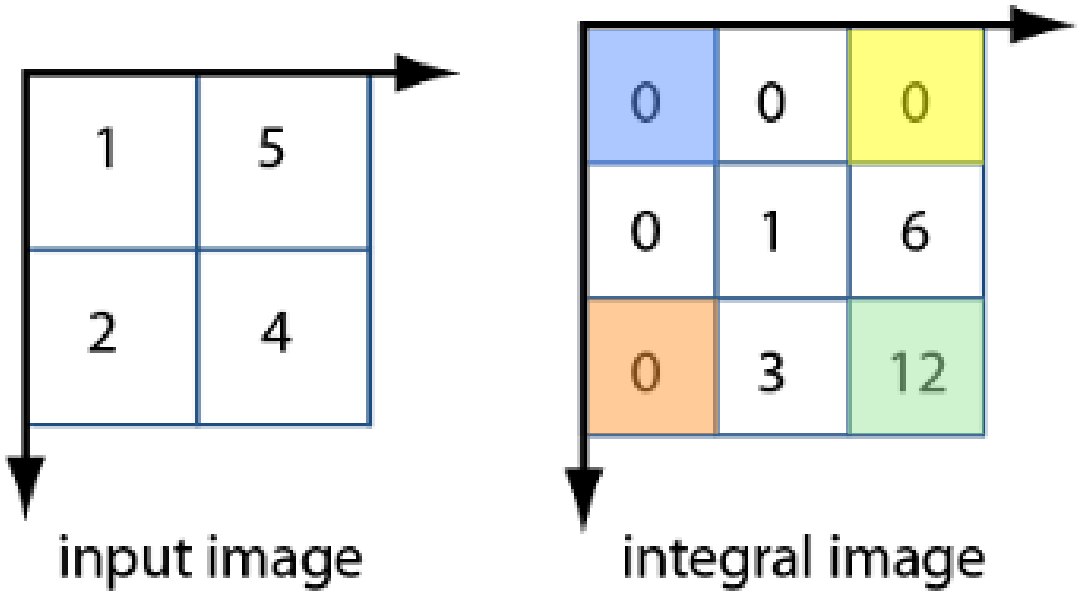
- Difference of the sum of pixels of areas inside the rectangle



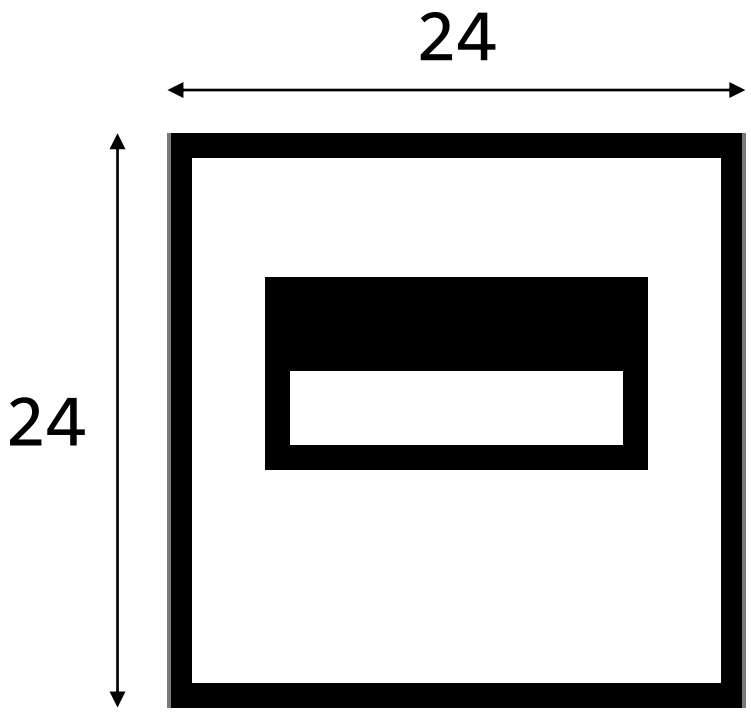
Integral Image

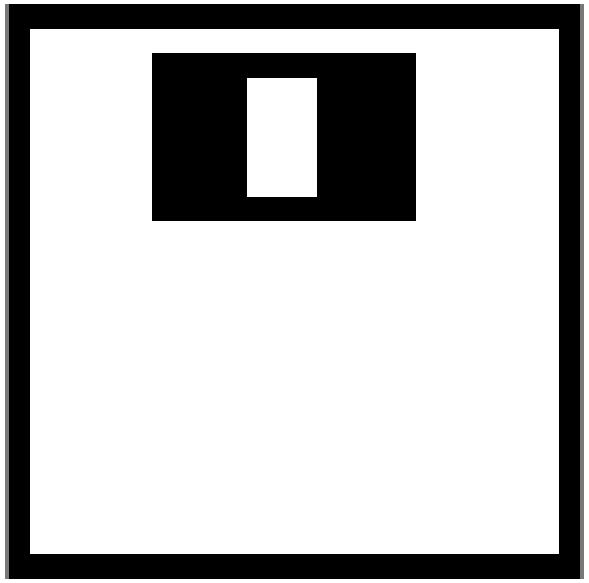
- Represents the cumulative sum of a corresponding input pixel with all pixels above and left of the input pixel

= d-c-b -a = 12



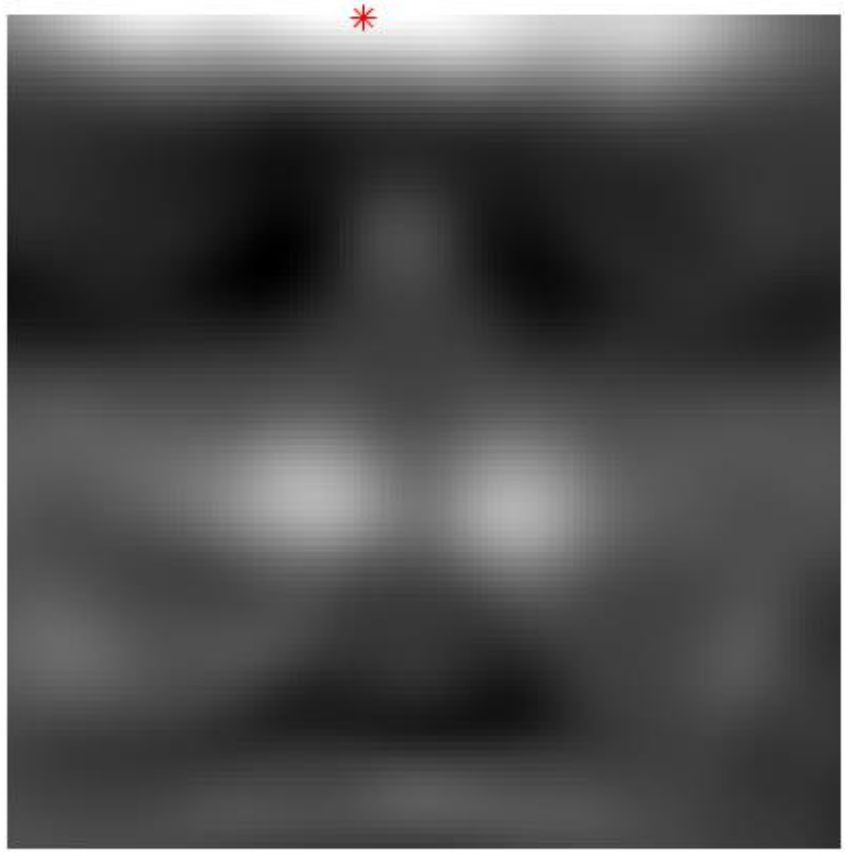
First Results



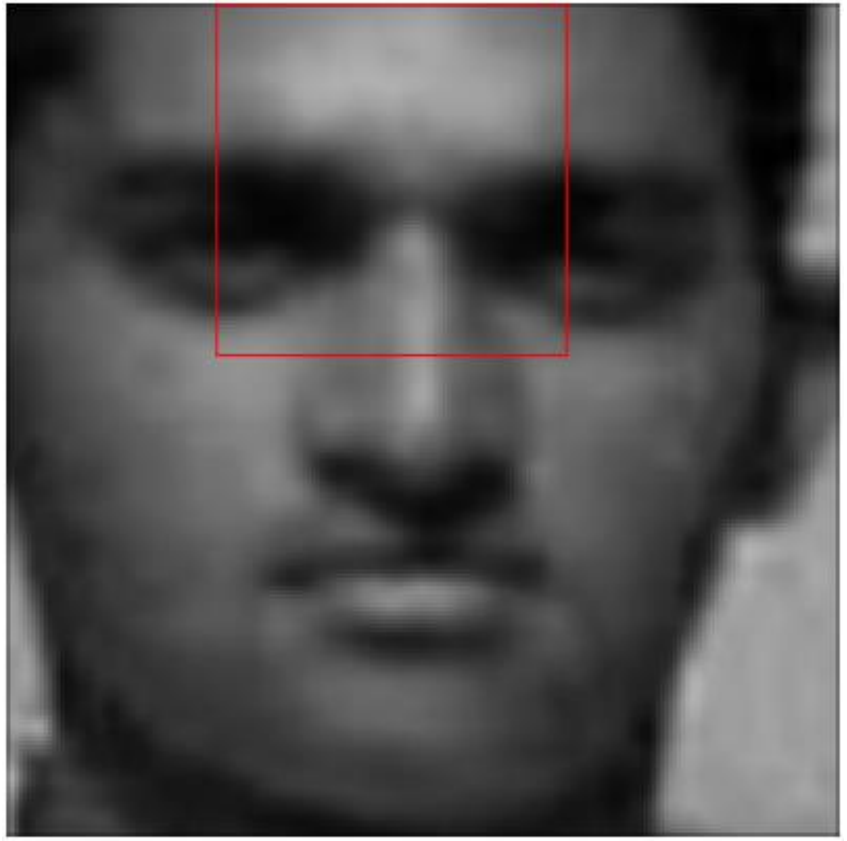


First Results

Intensity Map

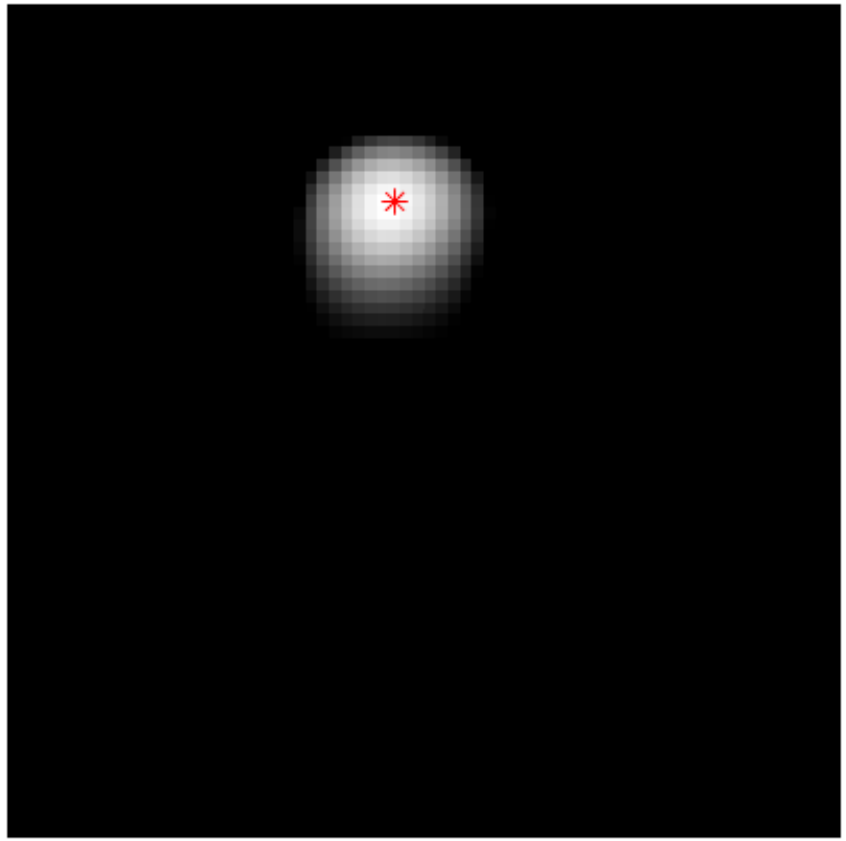


Detection

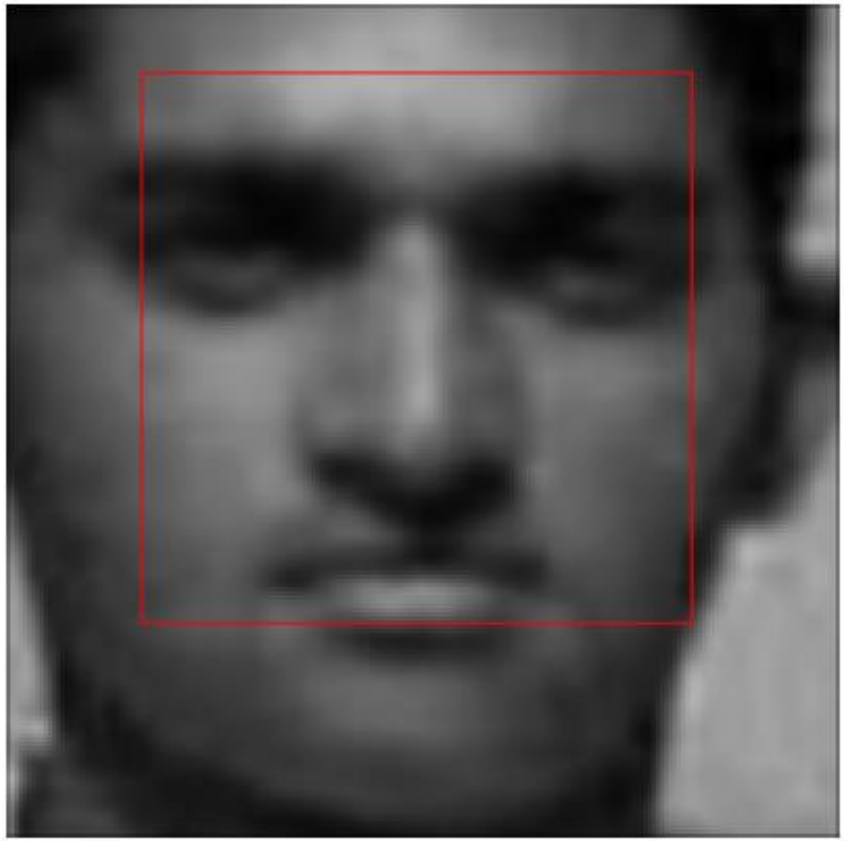


First Results

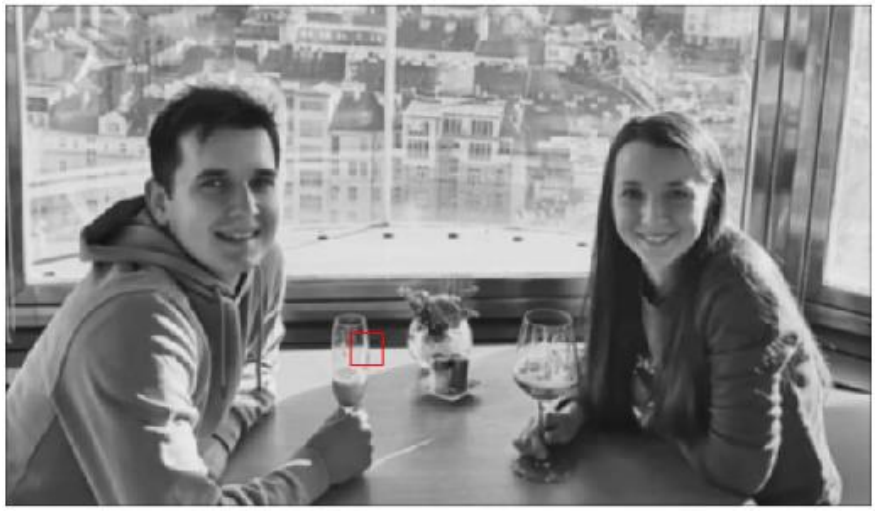
Intensity Map

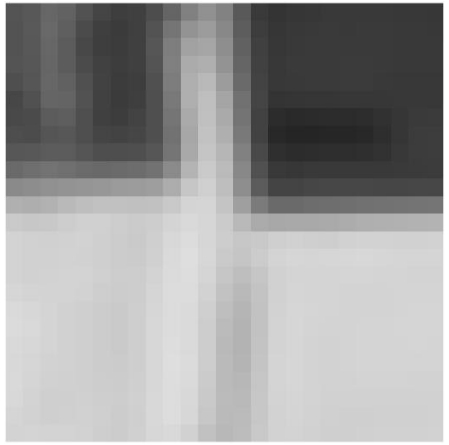


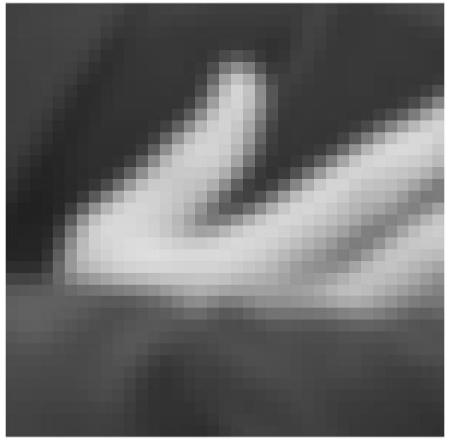
Detection



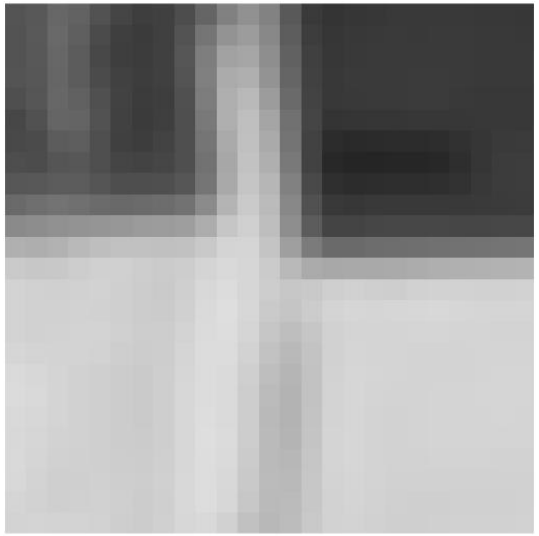
First Results

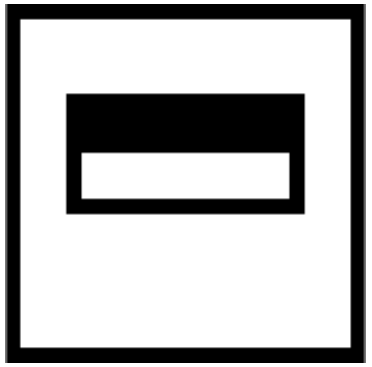


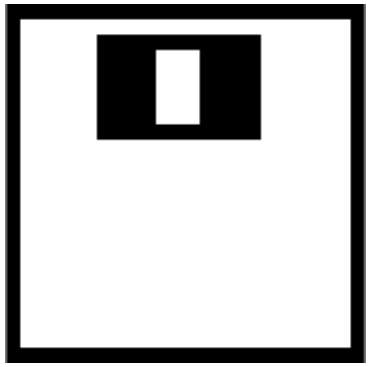




First Results





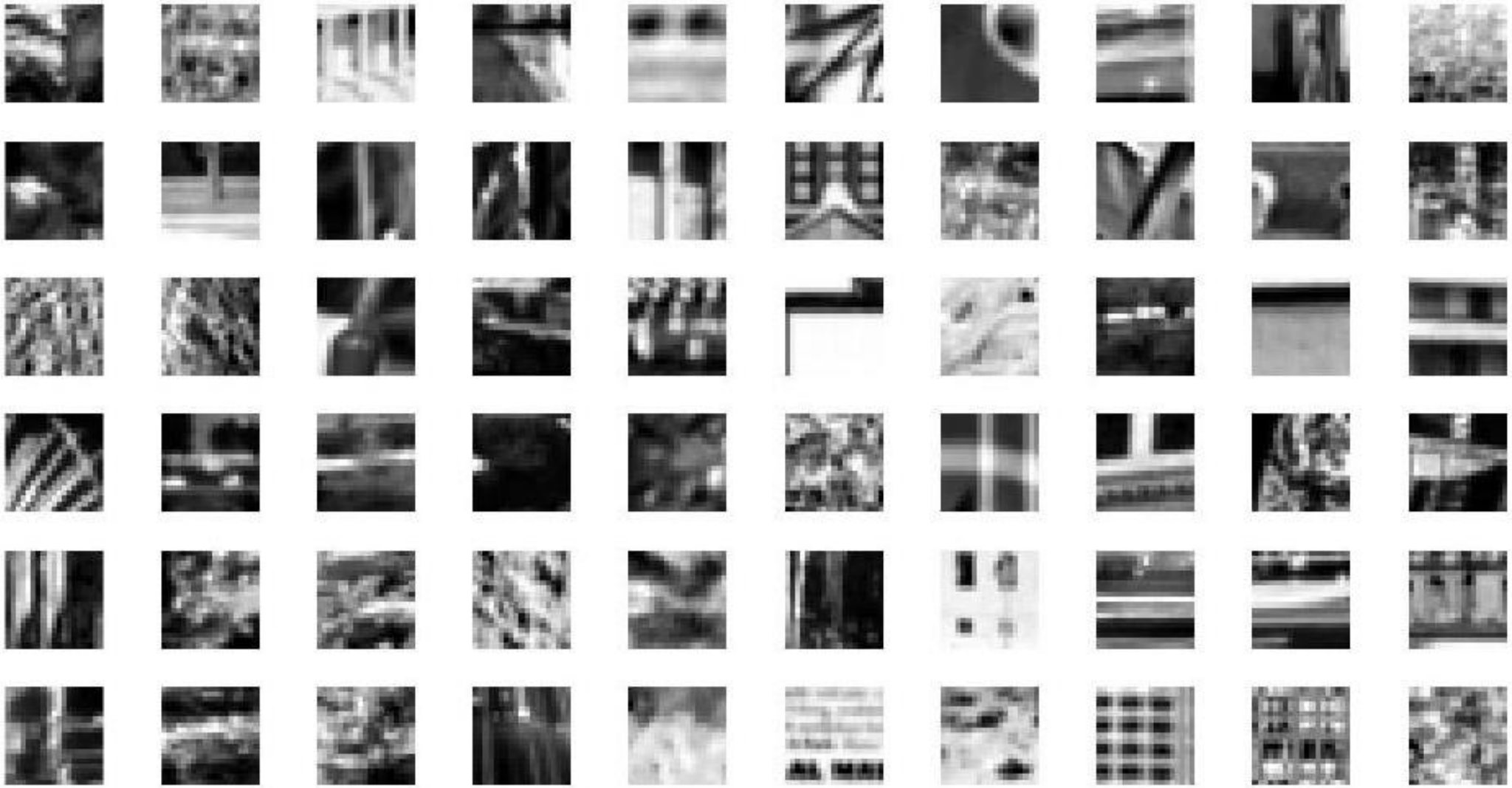


Training our own Haar features

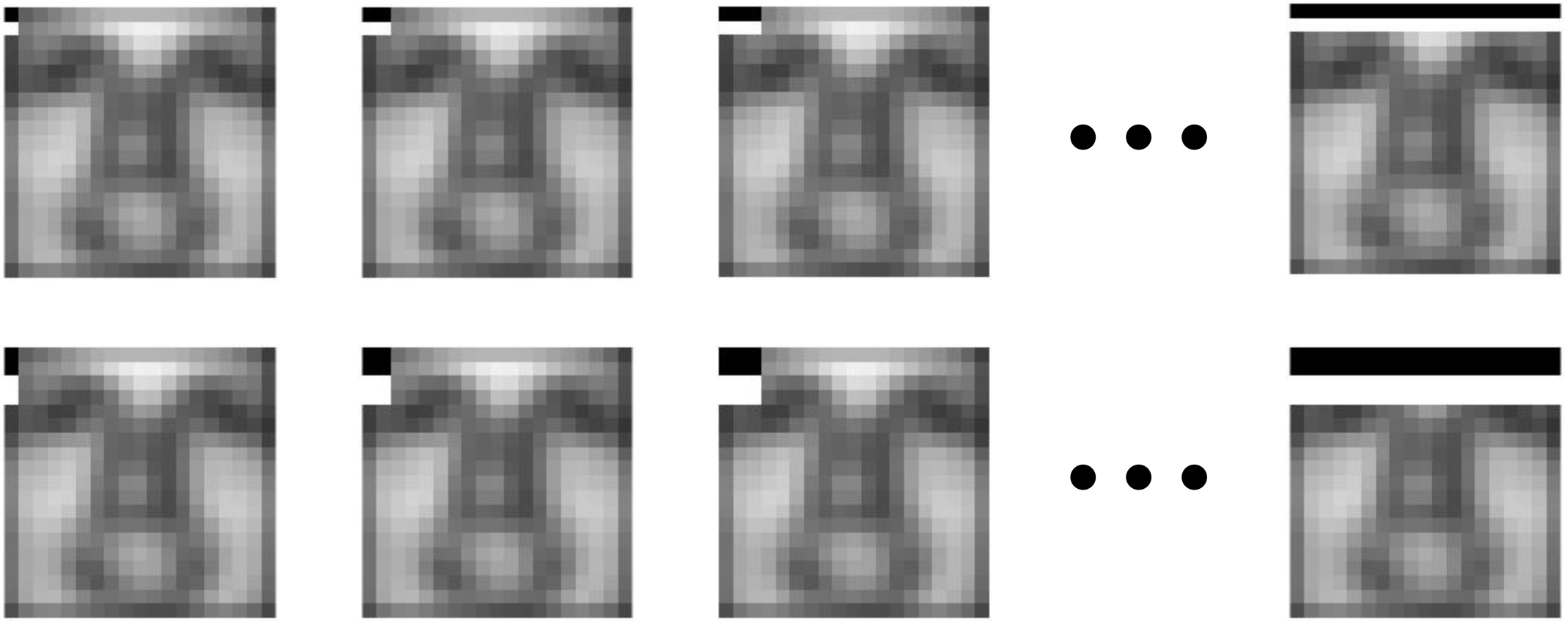
- Dataset:
  - 2429 face images
  - 4547 non-face images
  - All images already in grayscale with size 19x19 pixels
- Calculate Haar feature values
- AdaBoost

Face Images

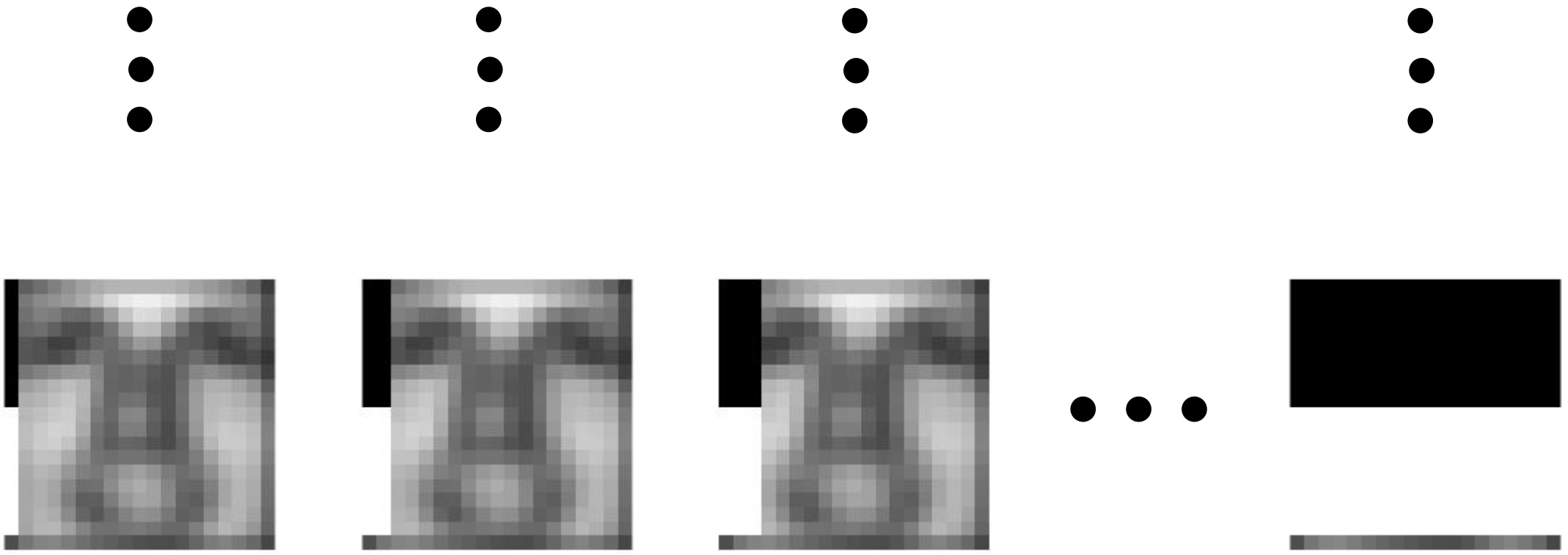
Non-Face Images



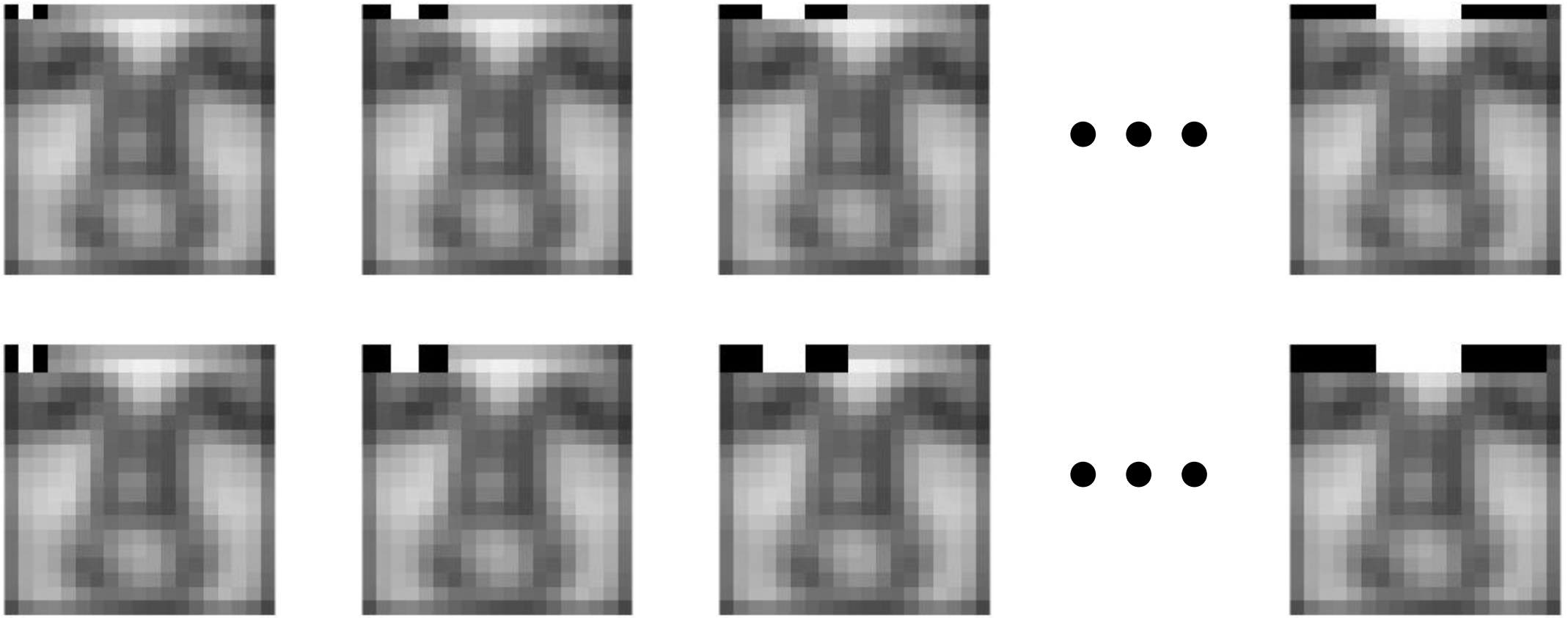
Calculating Haar feature values



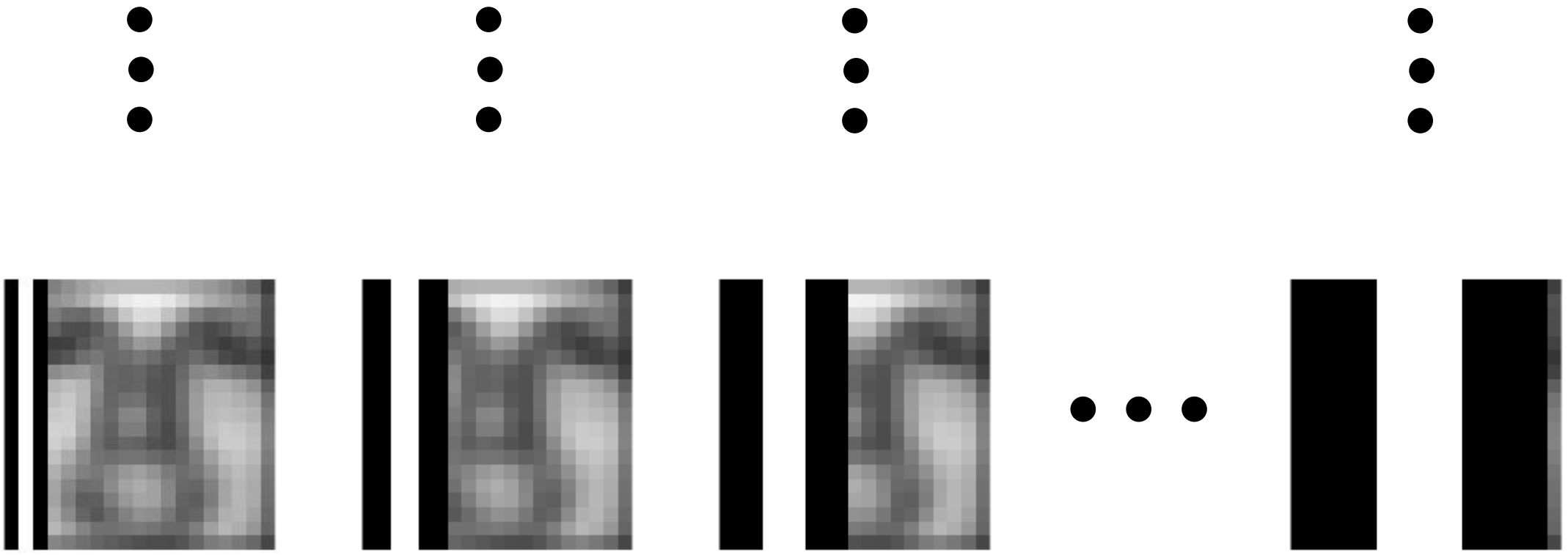
Calculating Haar feature values



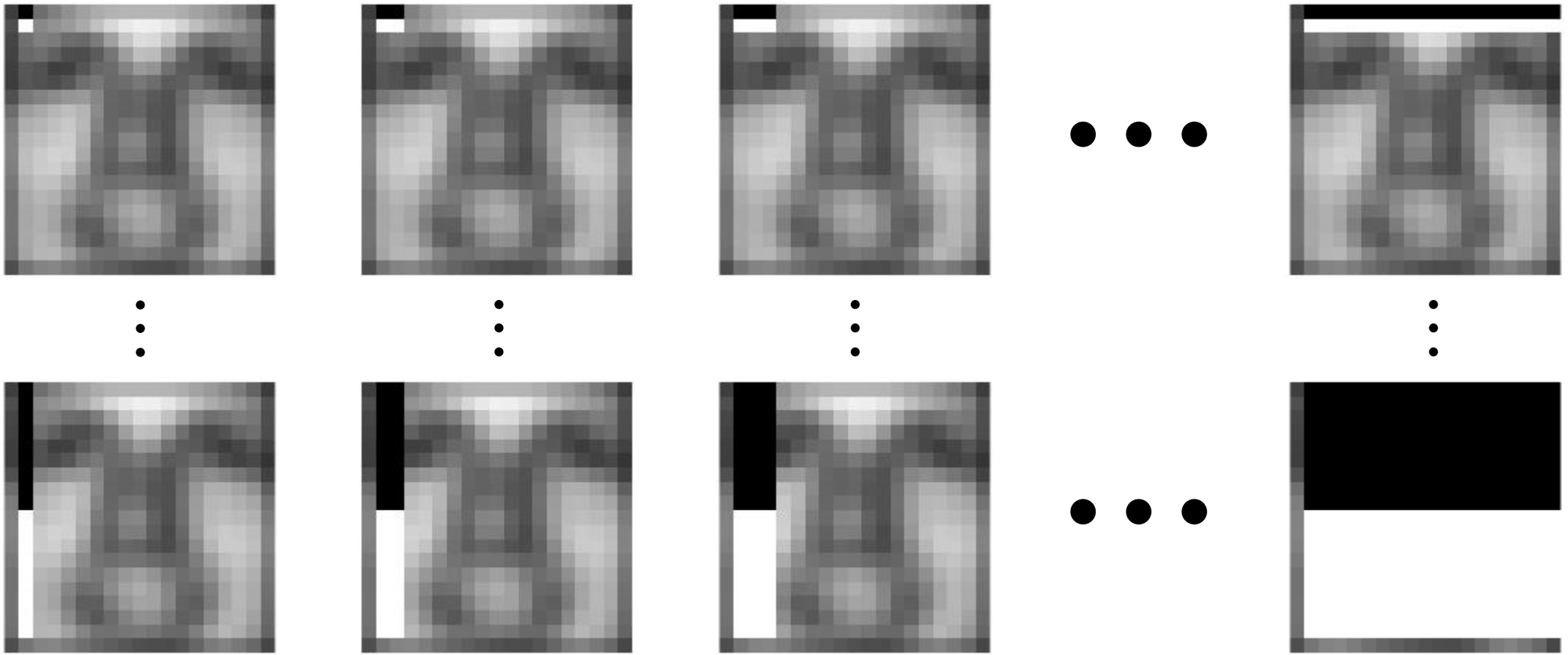
Calculating Haar feature values



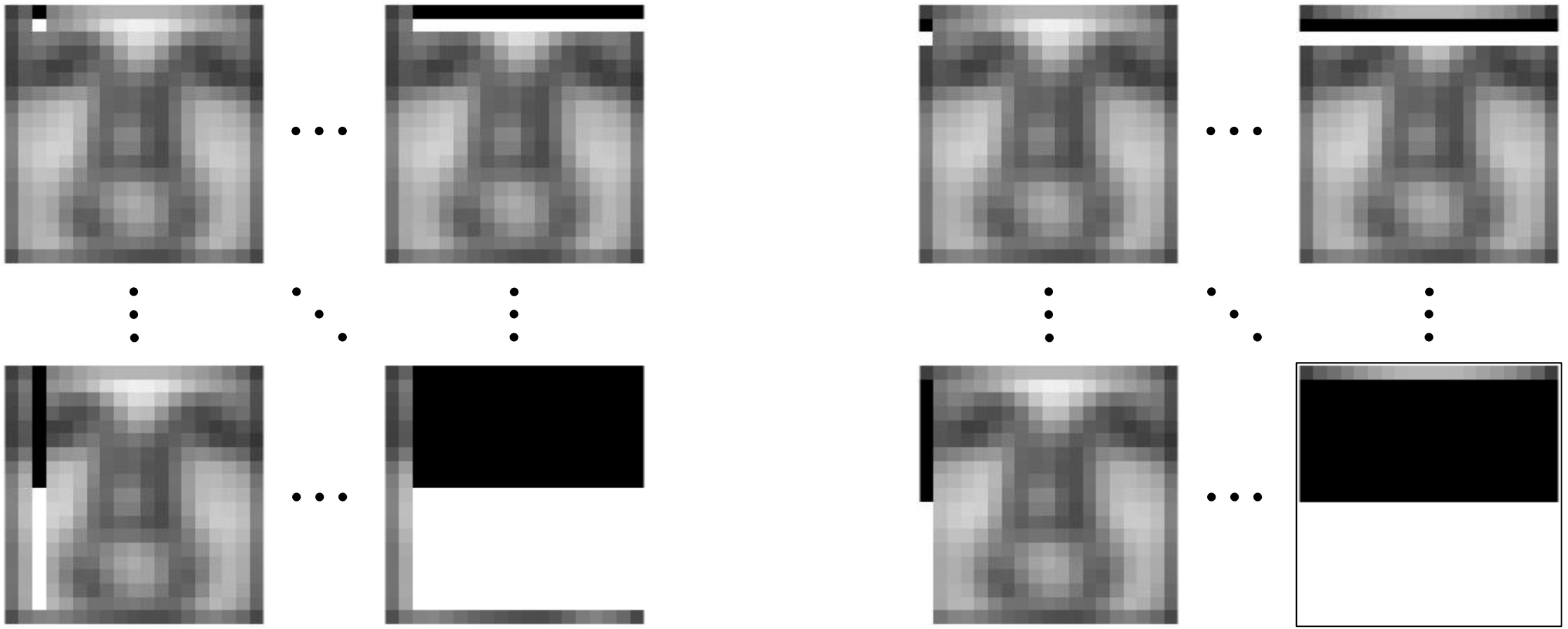
Calculating Haar feature values



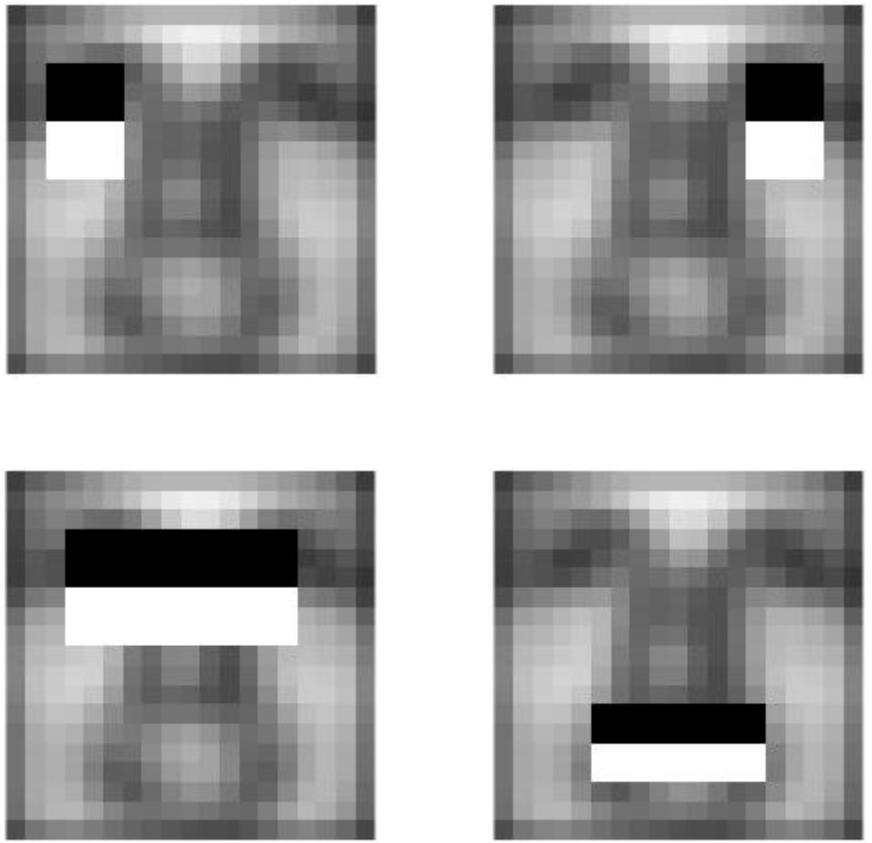
Calculating Haar feature values

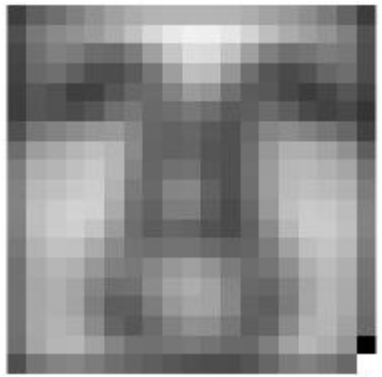


Calculating Haar feature values



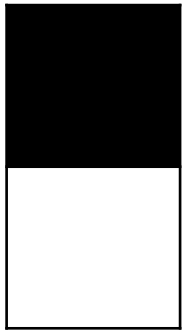
Calculating Haar feature values



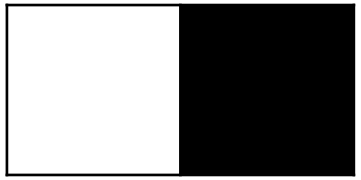


Types of Haar features

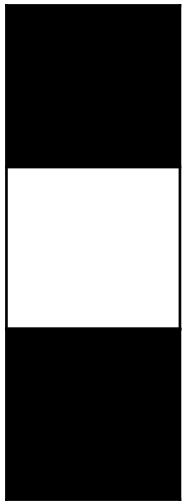
Type 1



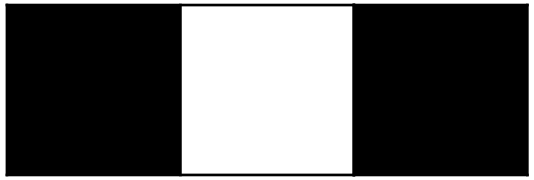
Type 2



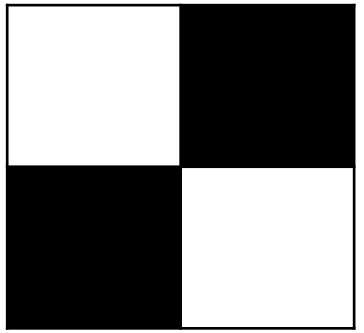
Type 3



Type 4



Type 5



AdaBoost

1.Step: Sorting

fval	sign
2	1
4	1
-1	1
1	-1
-3	-1
-5	-1

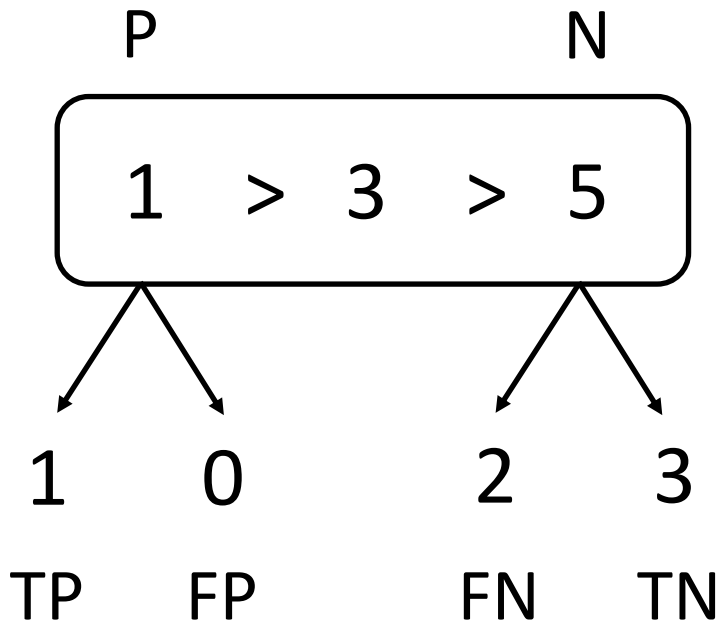
Sorting

fval	sign
4	1
2	1
1	-1
-1	1
-3	-1
-5	-1

2.Step: Threshold

fval	sign
4	1
2	1
1	-1
-1	1
-3	-1
-5	-1

- (4+2)/2 = 3
- 1.5
- 0
- -2
- -4



Gini impurity

Gini impurity for a leaf = 1 − (the probability of face)² - (the probability of no face)²

P:

$$= 1 - \left(\frac{1}{1+0}\right)^2 - \left(\frac{0}{1+0}\right)^2$$

$$= 1 - 1 - 0$$

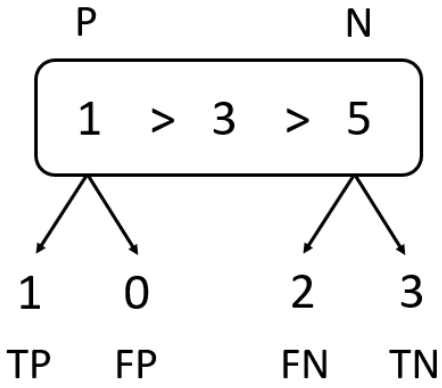
$$= 0$$

N:

$$= 1 - \left(\frac{2}{2+3}\right)^2 - \left(\frac{3}{2+3}\right)^2$$

$$= 1 - 0.16 - 0.36$$

$$= 0.48$$



Total Gini impurity = weighted average if Gini impurities for the leaves

$$= \left(\frac{1}{1+5}\right) \cdot 0 + \left(\frac{5}{1+5}\right) \cdot 0.48 = 0 + 0.4 \qquad = \mathbf{0.4}$$

Determining threshold

fval	sign
4	1
2	1
1	-1
-1	1
-3	-1
-5	-1

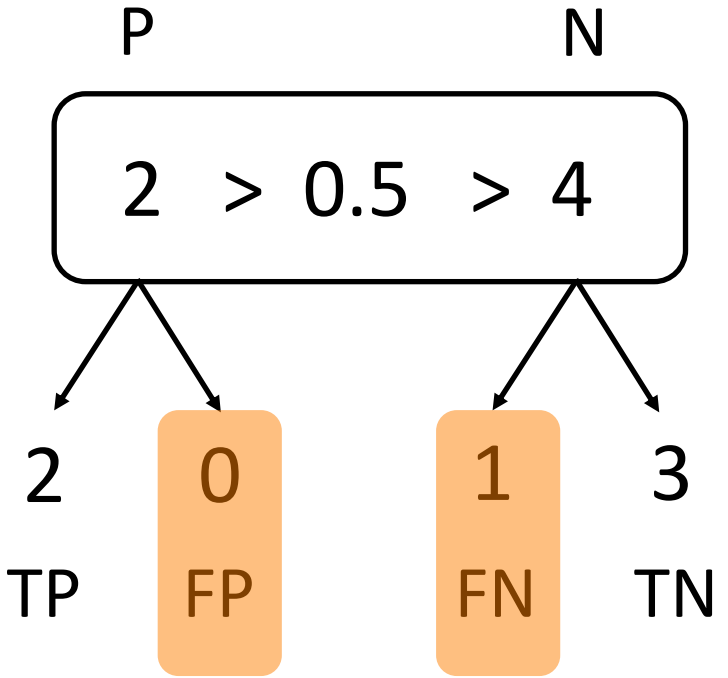
3 => 0.4

0.5 => 0.25

0 => 0.44

-2 => 0.25

-4 => 0.4



Incorrectly classified

3.Step: Calculating New Sample weights

fval	sign	weight
4	1	1/6
2	1	1/6
1	-1	1/6
-1	1	1/6
-3	-1	1/6
-5	-1	1/6

$$\text{Total error} = \frac{\text{sum of weights of incorrectly classified samples}}{} = \frac{1}{6} = 0.16\overline{6}$$

$$\text{Amount of say} = \frac{1}{2}\log\left(\frac{1-\text{Total error}}{\text{Total error}}\right) = \frac{1}{2}\log\left(\frac{1-1/6}{1/6}\right) = 0{,}805$$

$$\text{New Sample weight} = \text{Sample weight x } e^{\text{Amount of say}}$$

$$= \frac{1}{6}\cdot e^{0.805} = \mathbf{0.373}$$

Calculating New Sample weights

fval	sign	weight
4	1	0.166
2	1	0.166
1	-1	0.166
-1	1	0.373
-3	-1	0.166
-5	-1	0.166
		1.206

$$\times \frac{1}{1.206}$$

fval	sign	weight
4	1	0.138
2	1	0.138
1	-1	0.138
-1	1	0.309
-3	-1	0.138
-5	-1	0.138
		1

4.Step: New feature values

fval	sign	weight	csum
4	1	0.138	0.138
2	1	0.138	0.276
1	-1	0.138	0.415
-1	1	0.309	0.723
-3	-1	0.138	0.862
-5	-1	0.138	1

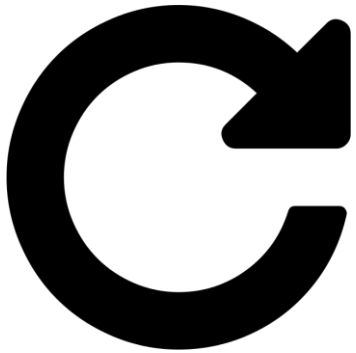
Random number between 0-1
0.1067
0.5265
0.7749
0.3998
0.8173
0.6032

fval	sign
4	1
-1	1
-3	-1
2	1
-3	-1
-1	1

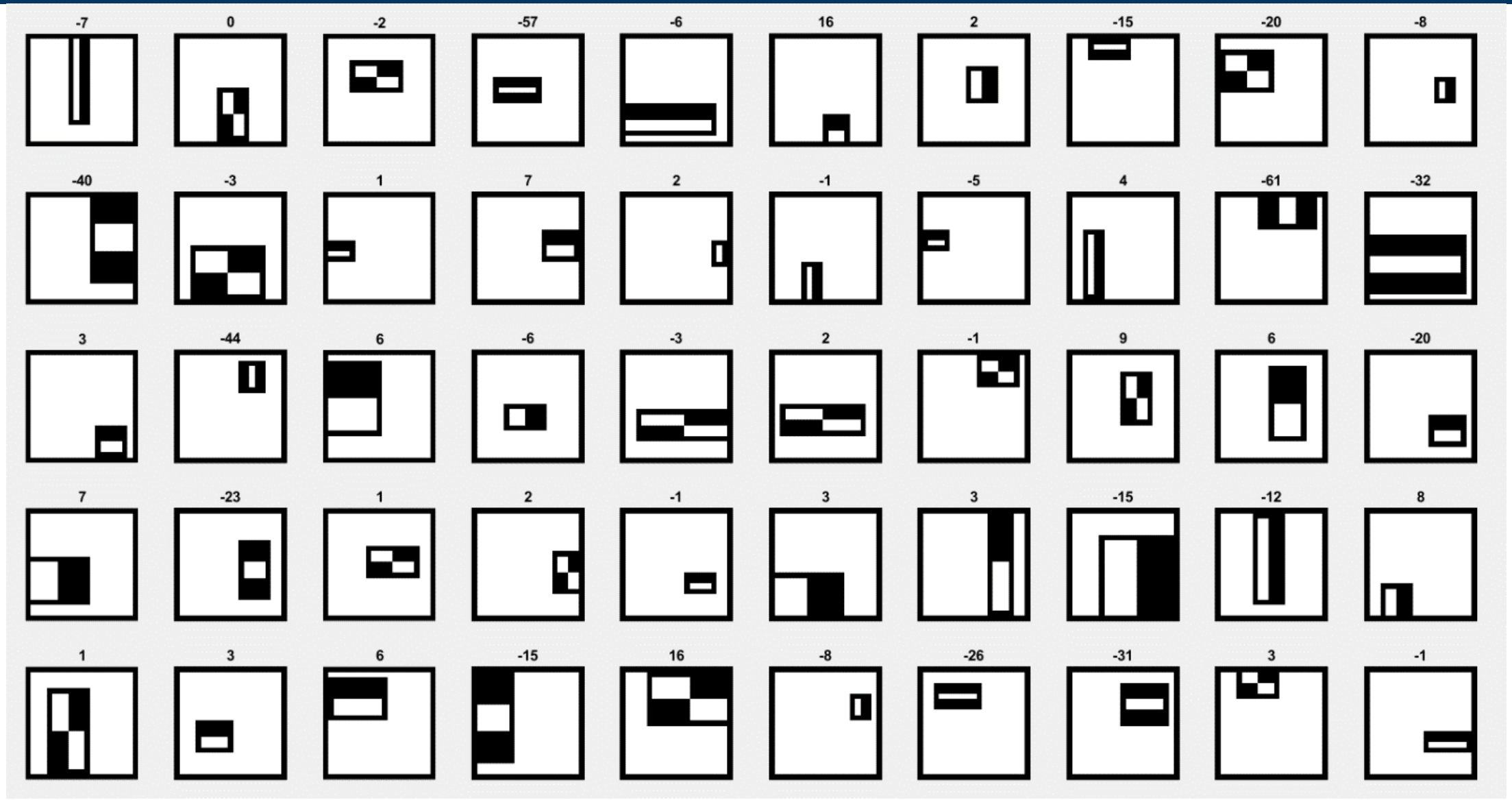
5.Step: Repeat

fval	sign
4	1
-1	1
-3	-1
2	1
-3	-1
-1	1

Repeat for a number of iterations



Results from AdaBoost (50 itr)



Final Results

Applied Haar features

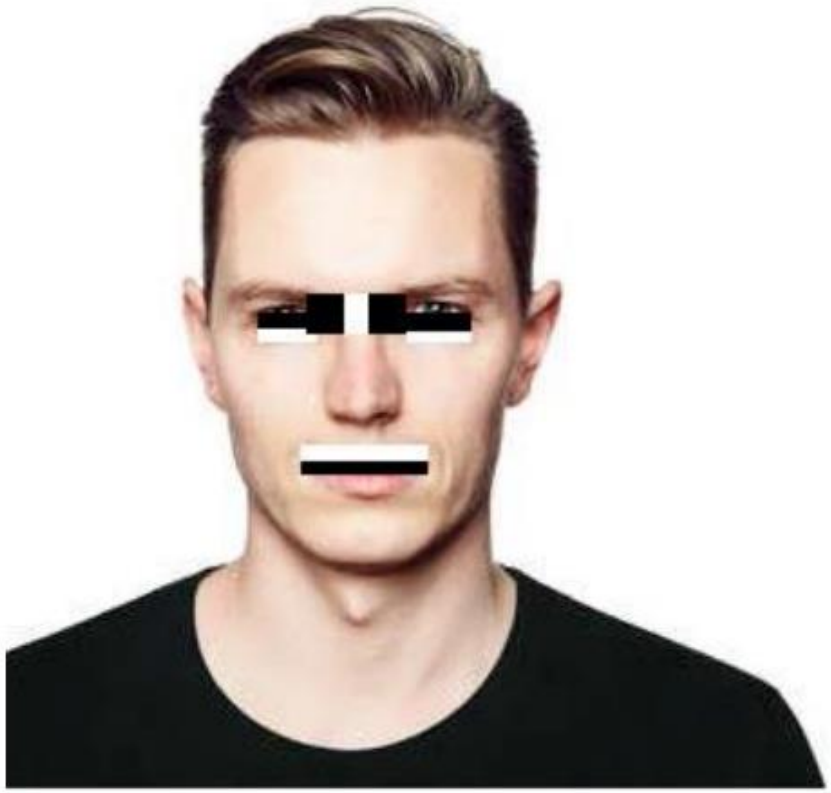
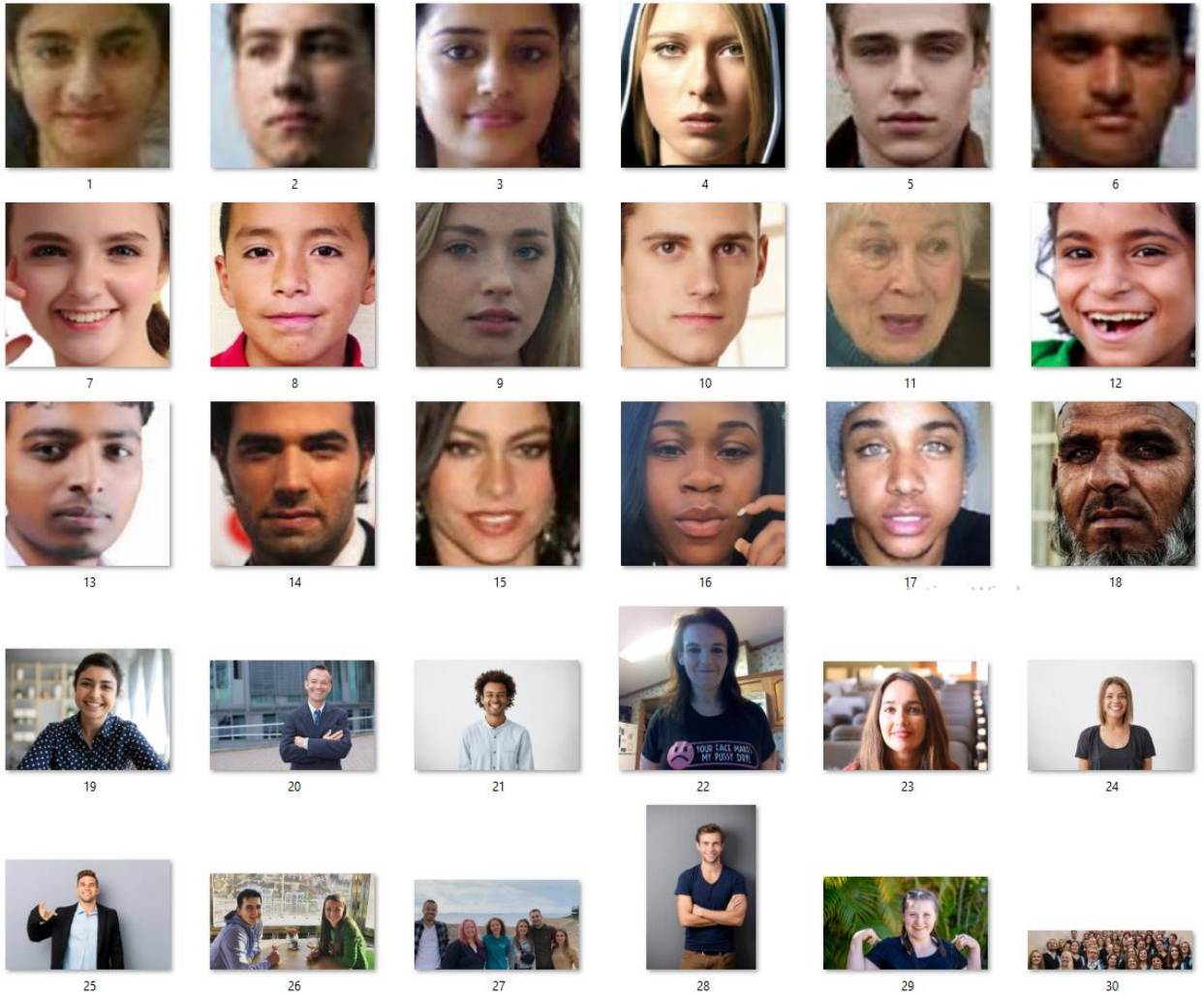
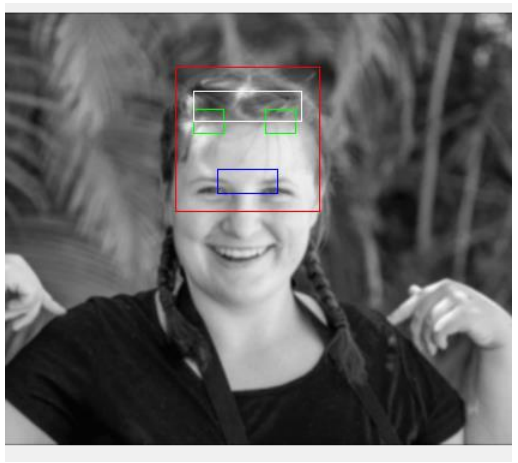


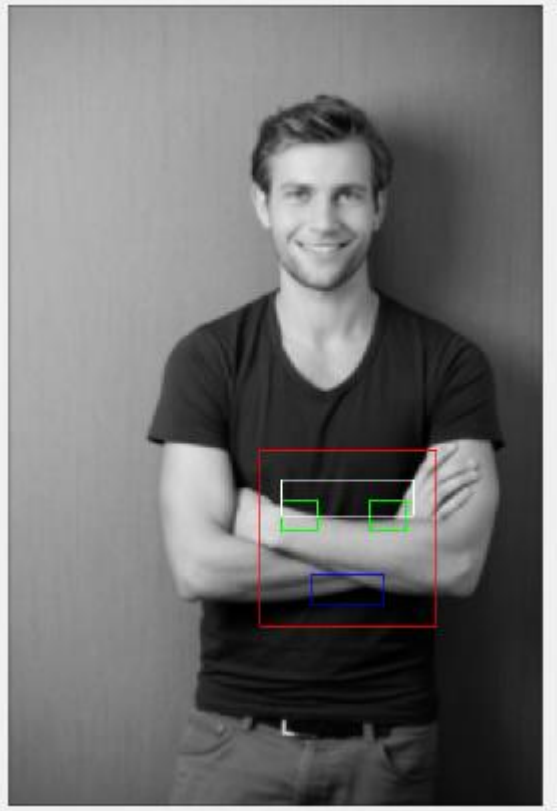
Image database

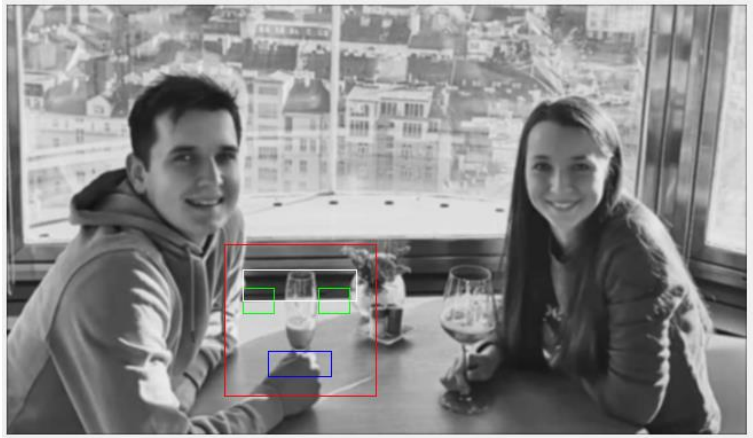


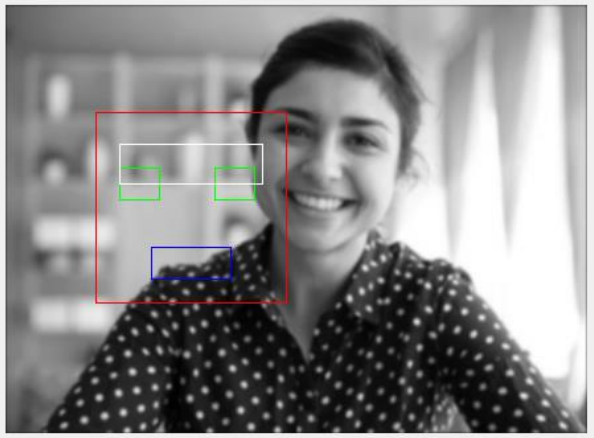
Final results

Final results

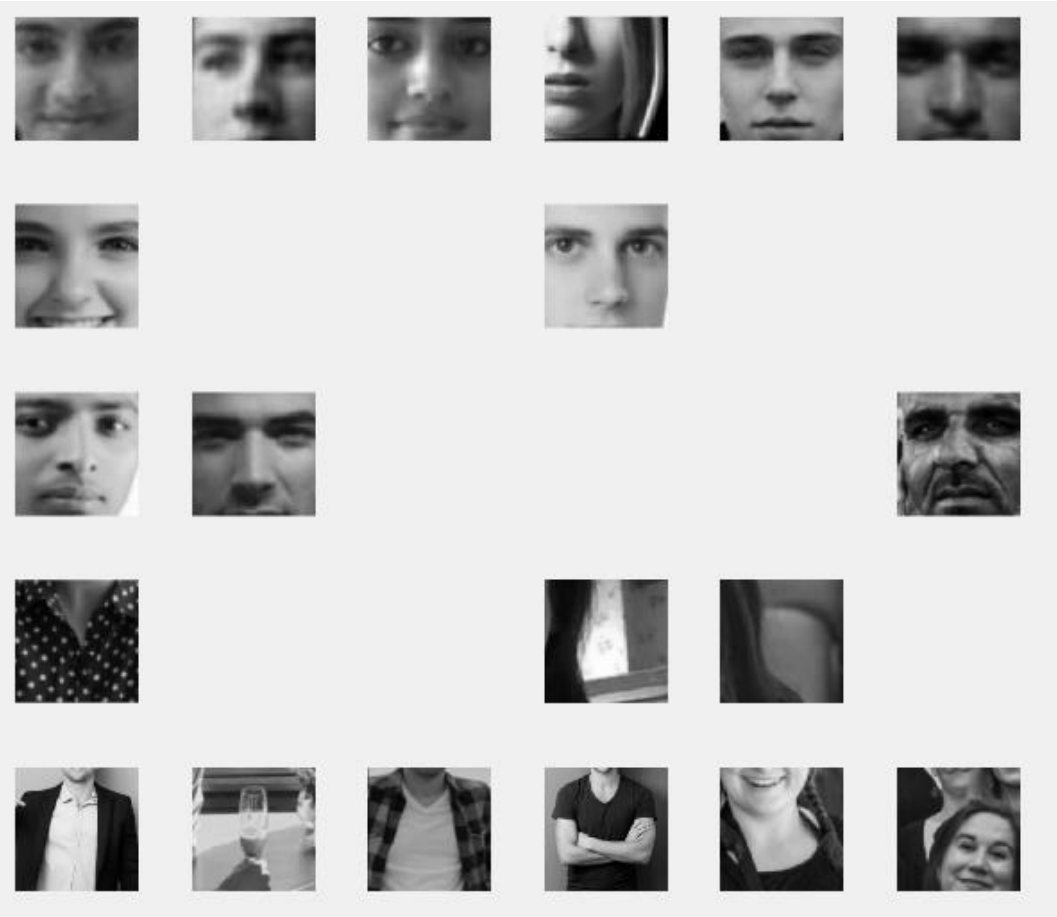








Comparison of results



Things we tried

- Optimizing the initial size of the Haar feature
- Different combinations of Haar features
- Changing threshold
- Cascade
- Calculating other types of Haar features using AdaBoost

We want to try:

- To apply exactly the features obtained from AdaBoost with their calculated threshold

Conclusions

- It’s **HARD** and it takes a lot of time to try different features manually.
- Threshold is very important and is different for each type and size of Haar feature.
- Almost impossible to get the perfect cascade of classifiers that works on complicated images manually.

- **STUDY MACHINE LEARNING**

```
%% Thank you for attention

% If you have any questions
% don’t hesitate to ask
```