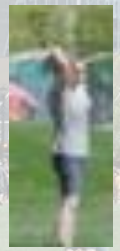
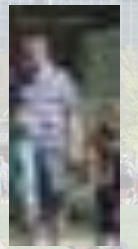
Face Detection

Hyun Soo Park

Challenges of Visual Recognition

- Appearance
  - DOF: texture, illumniation, material, shading, …
- **Shape**
  - DOF: object category, geometric pose, viewpoint, …



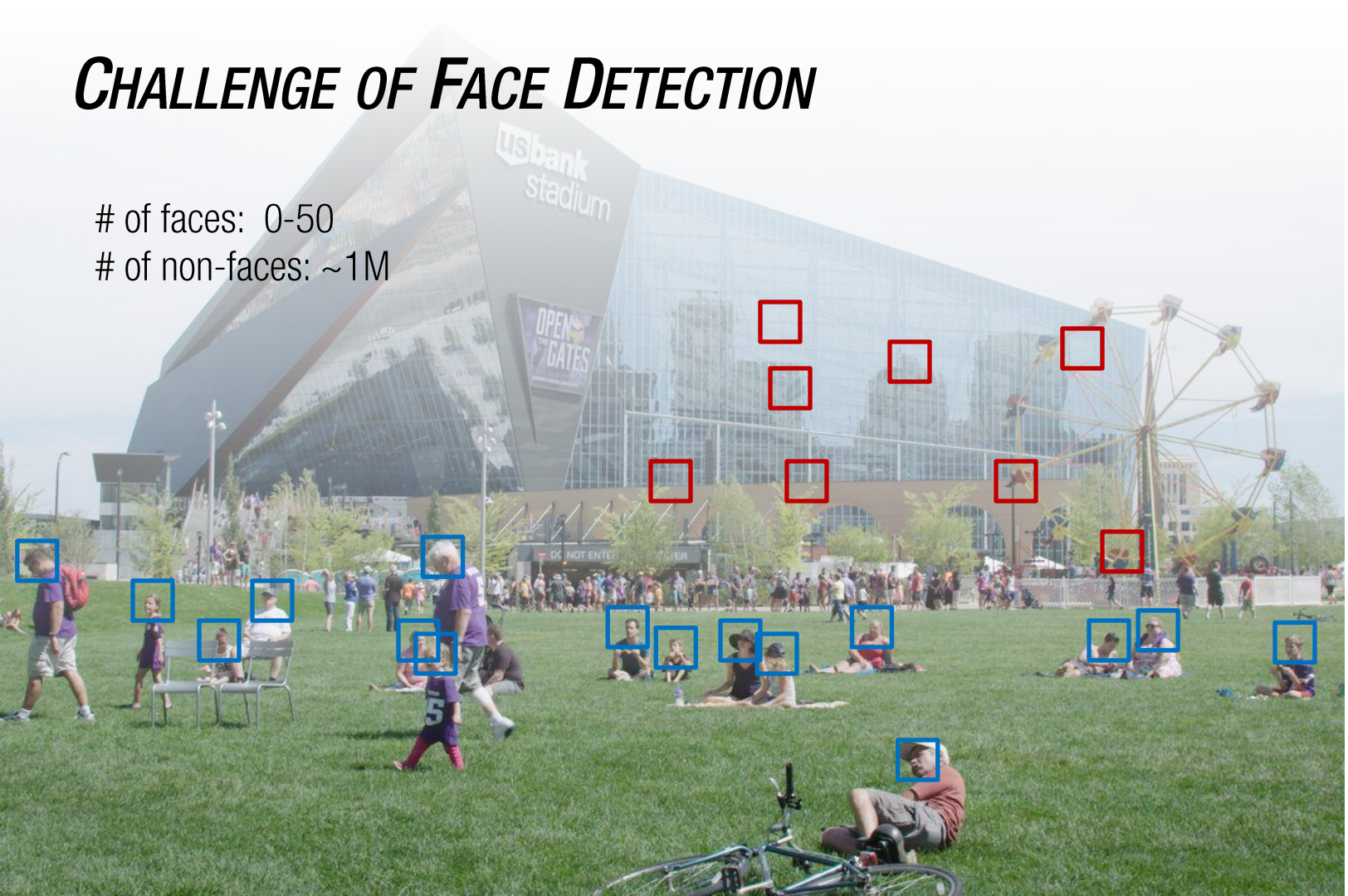




CHALLENGE OF FACE DETECTION

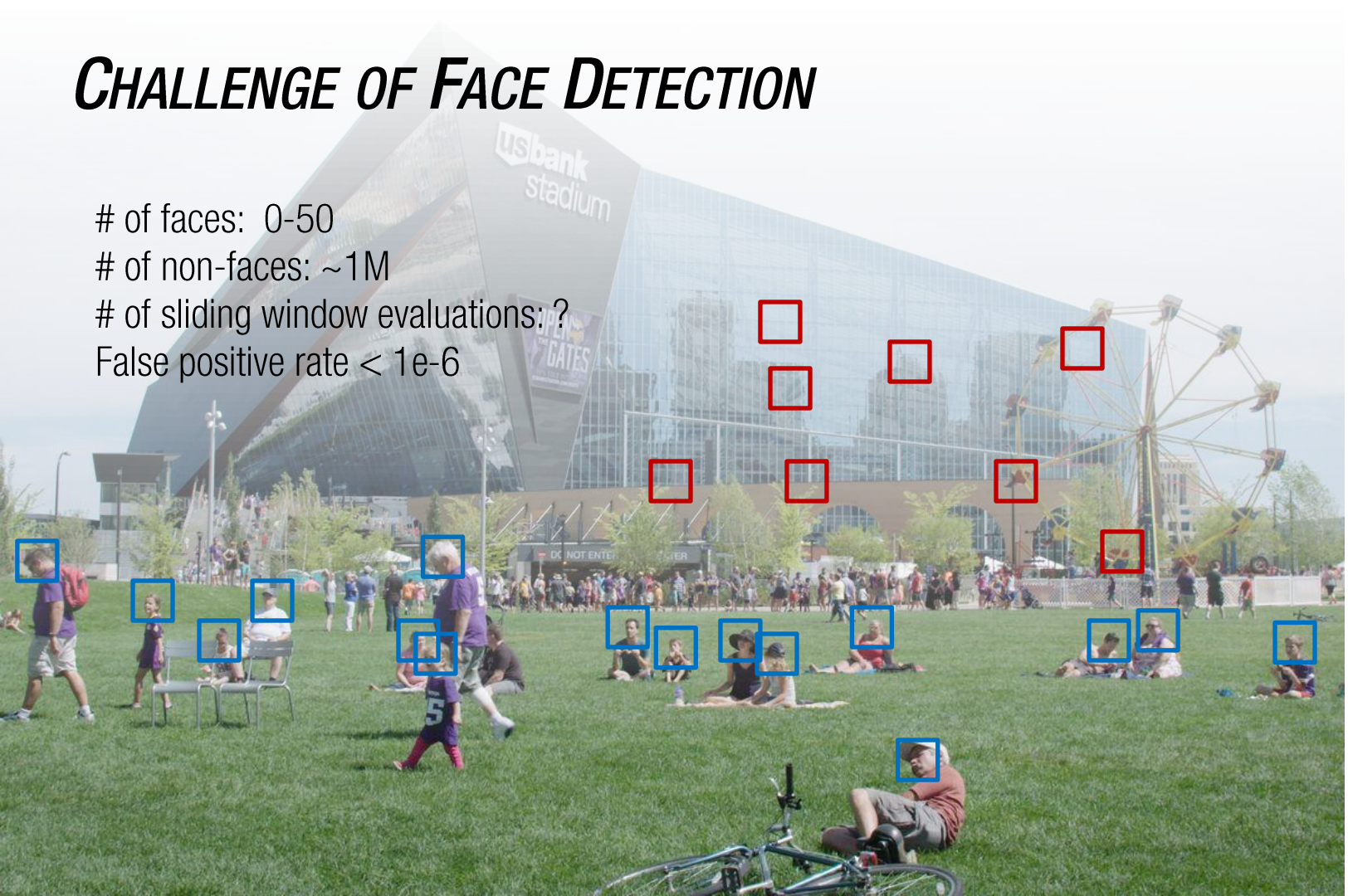
of faces: 0-50

Challenge of Face Detection

of faces: 0-50
# of non-faces: ~1M



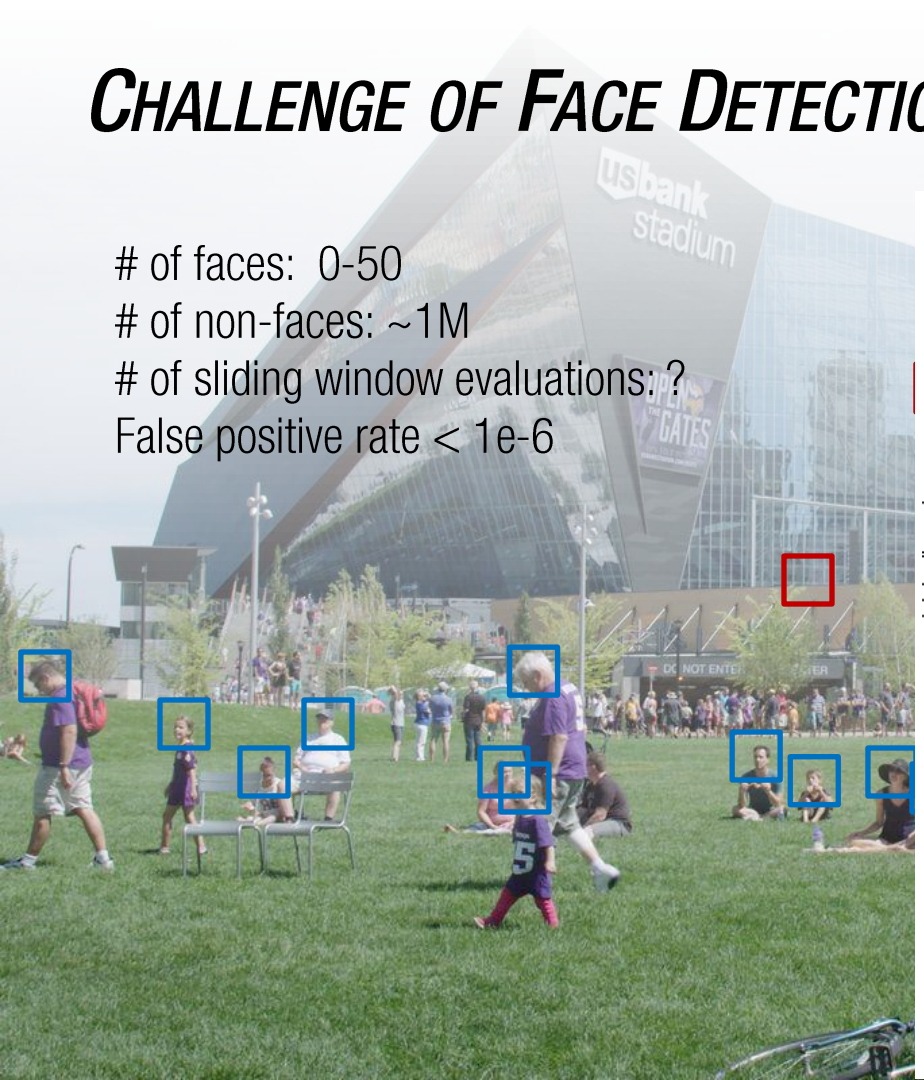
Challenge of Face Detection

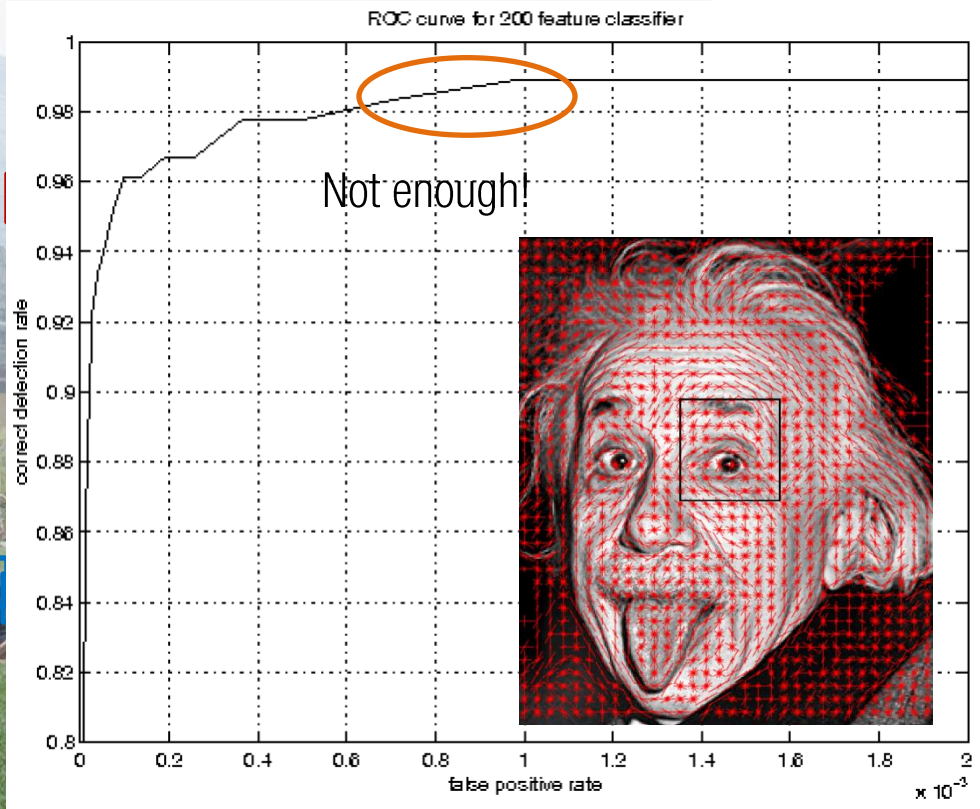


# of faces: 0-50
# of non-faces: ~1M
# of sliding window evaluations: ?
False positive rate < 1e-6

Challenge of Face Detection

of faces: 0-50
# of non-faces: ~1M
# of sliding window evaluations: ?
False positive rate < 1e-6



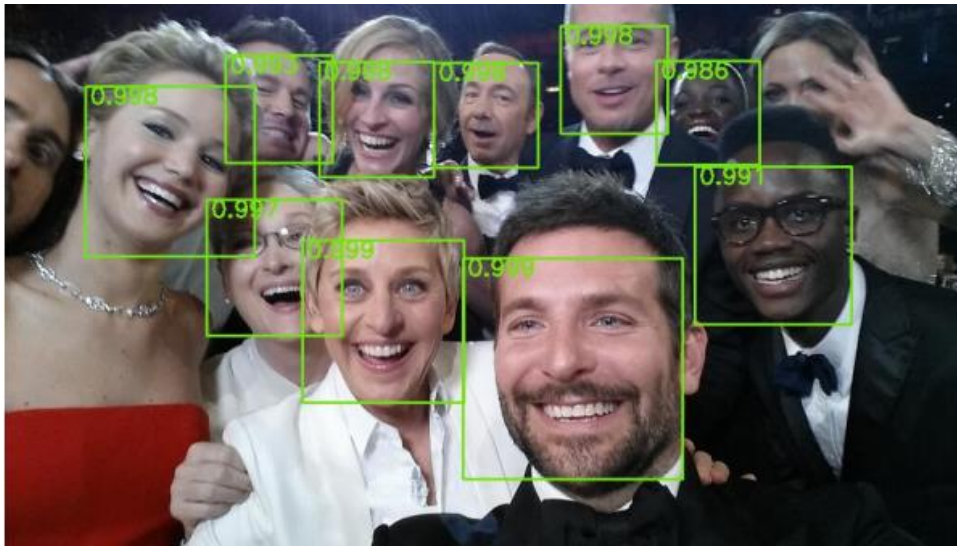


ROC curve for 200 feature classifier

VIOLA-JONES FACE DETECTION

Extremely fast and accurate face detection
- Running at real-time

https://www.youtube.com/watch?v=aTErTqOlkss

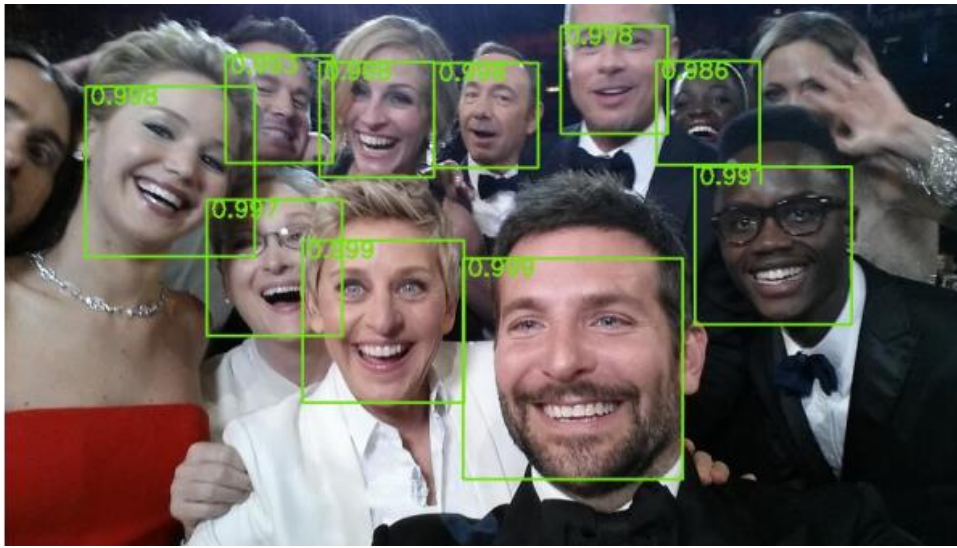


VIOLA-JONES FACE DETECTION

Extremely fast and accurate face detection
- Running at real-time

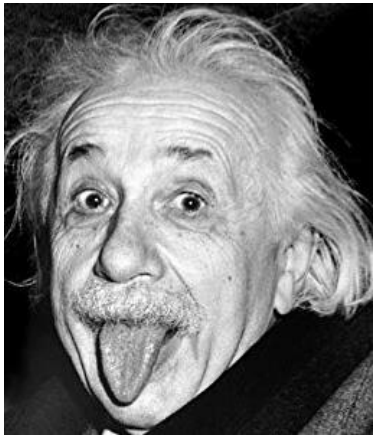
https://www.youtube.com/watch?v=aTErTqOlkss

Enabling factors:
- Efficient feature computation
- Efficient feature selection
- Efficient inference algorithm

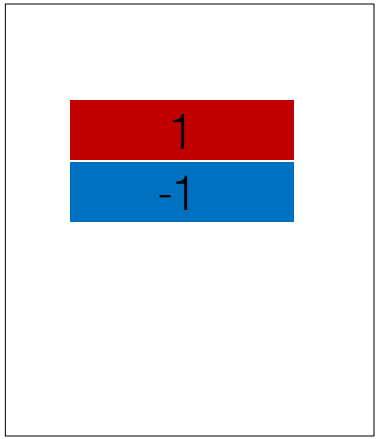


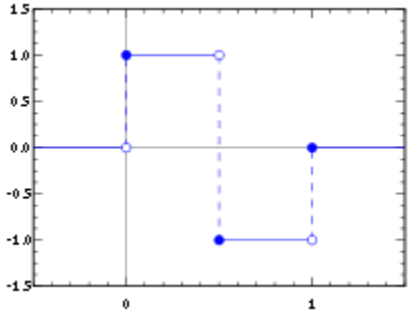
HAAR-LIKE FEATURE

A simple rectangular filter



⊗

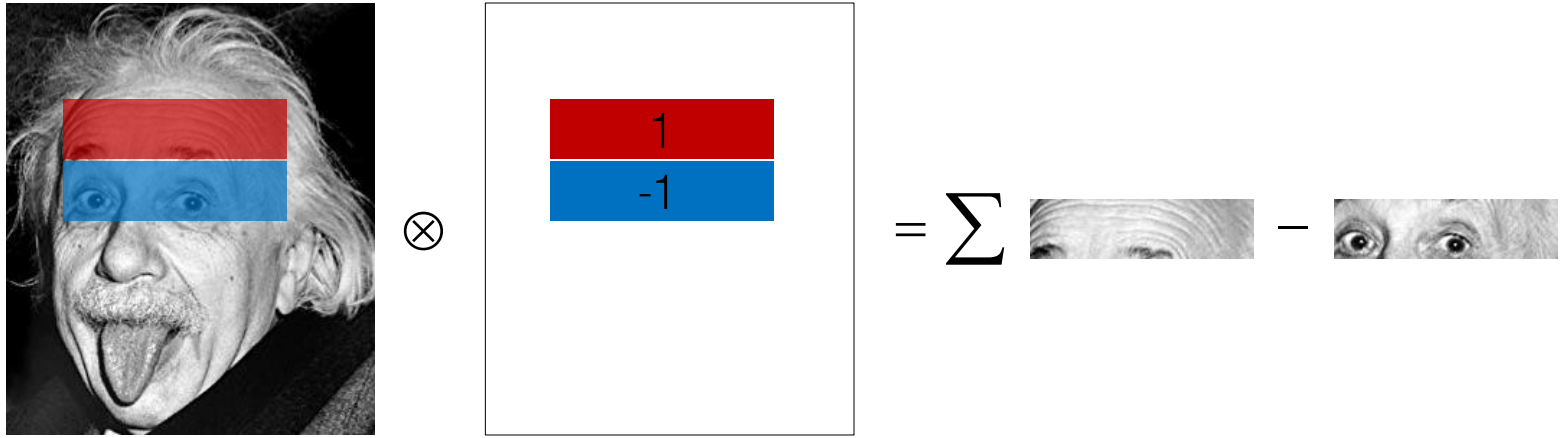




Wavelet filter

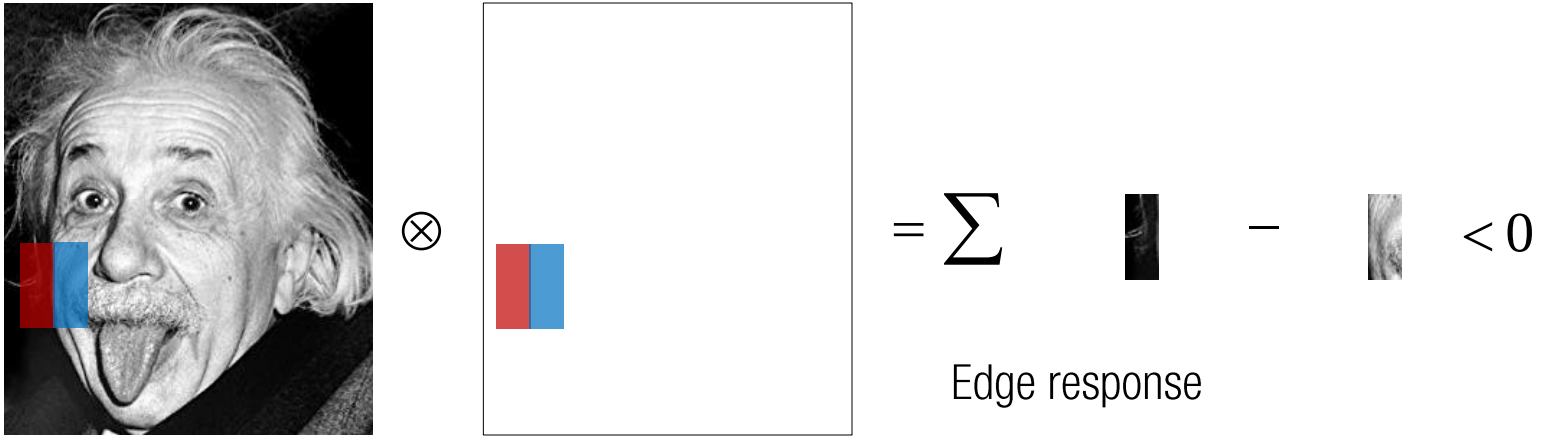
HAAR-LIKE FEATURE

A simple rectangular filter



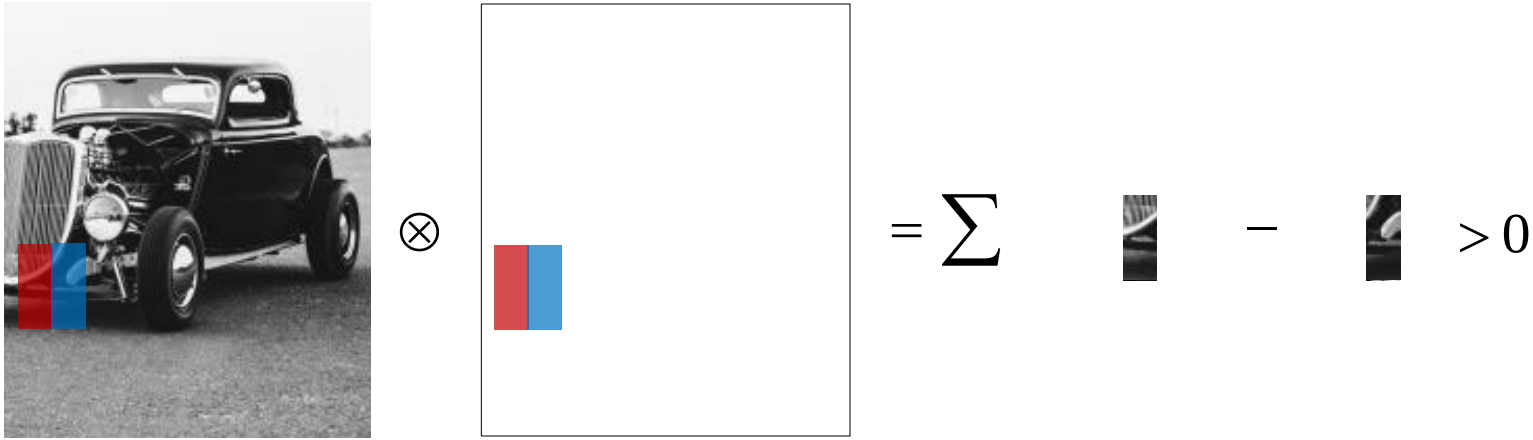
Haar-like Feature

A simple rectangular filter



HAAR-LIKE FEATURE

A simple rectangular filter

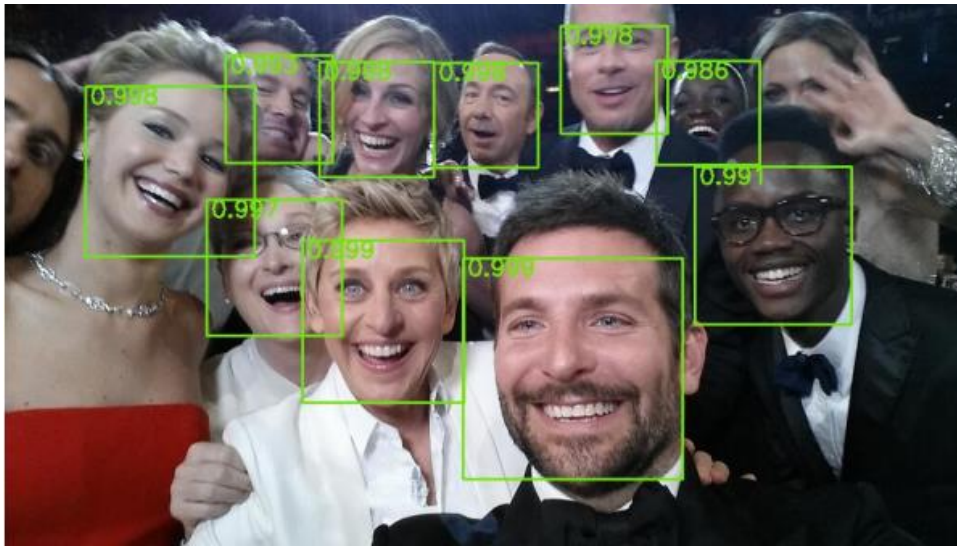


VIOLA-JONES FACE DETECTION

Extremely fast and accurate face detection
- Running at real-time

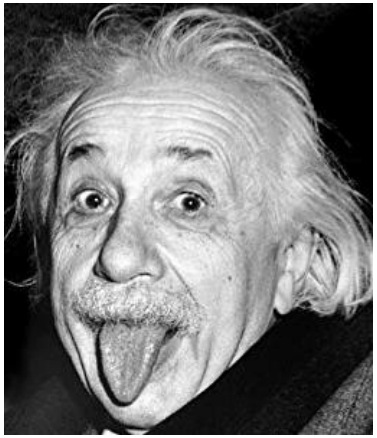
https://www.youtube.com/watch?v=aTErTqOlkss

Enabling factors:
- **Efficient feature computation**
- Efficient feature selection
- Efficient inference algorithm



Integral Image

Image with values at each pixel that is the sum of pixels above and left inclusive.

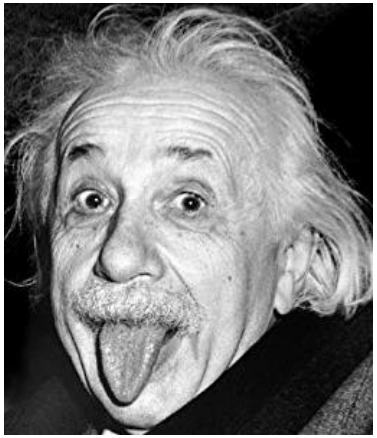


I

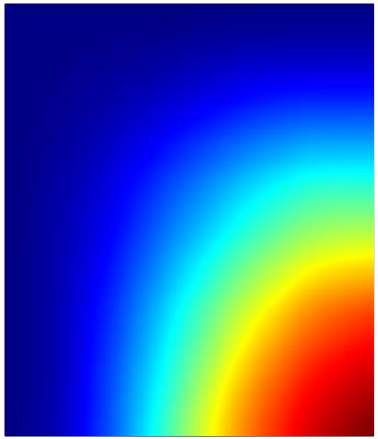
$$J(x,y) = \sum_{x<X,\,y<Y} \sum I(x,y)$$

Integral Image

Image with values at each pixel that is the sum of pixels above and left inclusive.



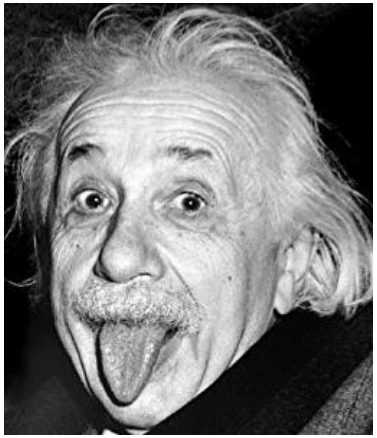
I



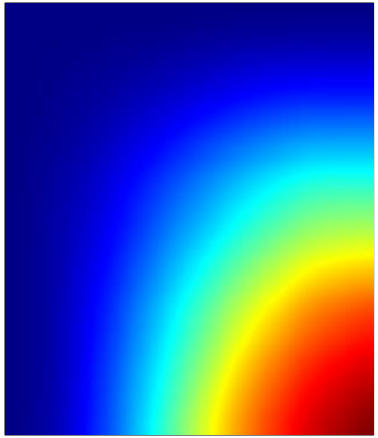
$$J(x,y) = \sum_{x<X,\,y<Y} \sum I(x,y)$$

INTEGRAL IMAGE

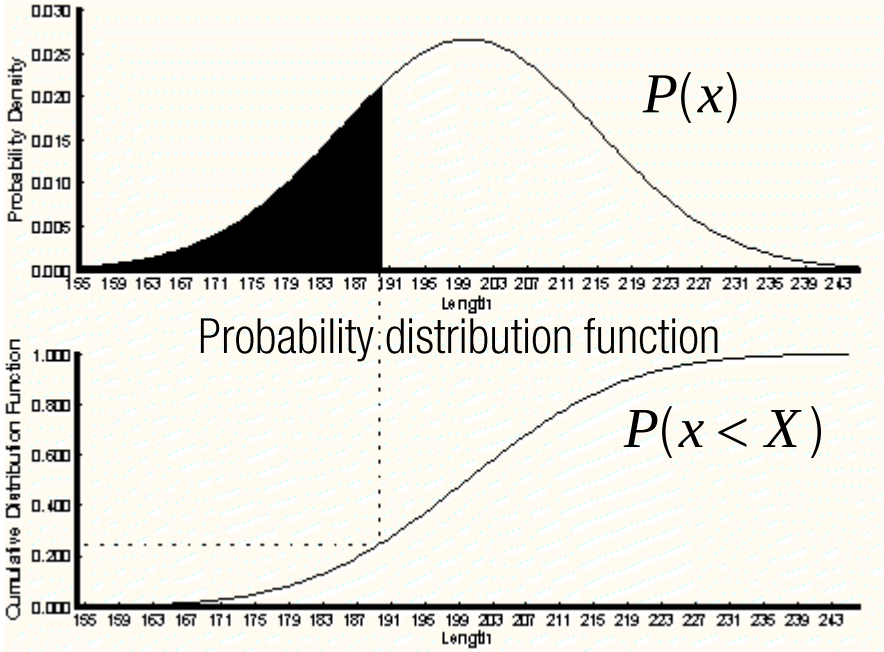
Image with values at each pixel that is the sum of pixels above and left inclusive.



I



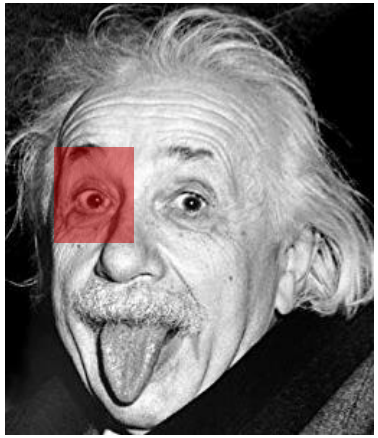
$$J(x,y) = \sum_{x<X,\,y<Y} \sum I(x,y)$$



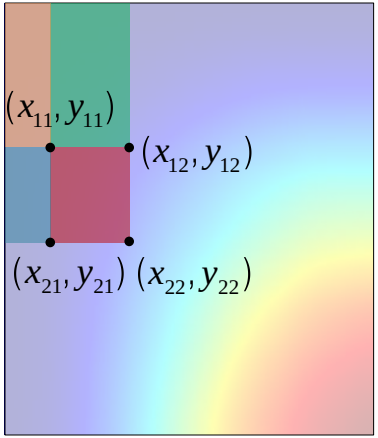
Probability distribution function

Cummulative distribution function

Feature Computation with Integral Image



I



$$J(x,y) = \sum_{x<X,\, y<Y} \sum I(x,y)$$

Sum of pixels in the area:

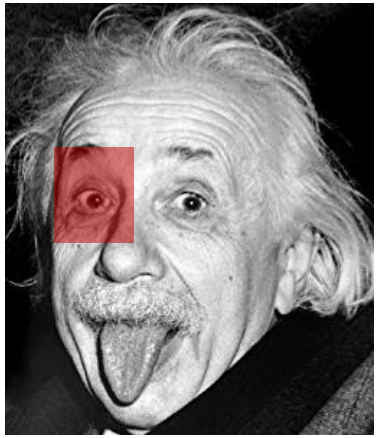
$J(x_{22}, y_{22}) =$

$J(x_{12}, y_{12}) =$

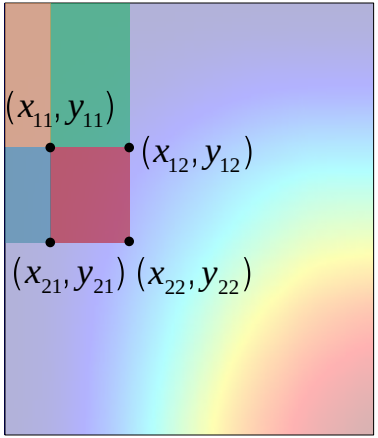
$J(x_{21}, y_{21}) =$

$J(x_{11}, y_{11}) =$

Feature Computation with Integral Image



I



$$J(x,y) = \sum_{x<X,\, y<Y} \sum I(x,y)$$

Sum of pixels in the area:

$J(x_{22}, y_{22}) =$

$J(x_{12}, y_{12}) =$

$J(x_{21}, y_{21}) =$

$J(x_{11}, y_{11}) =$

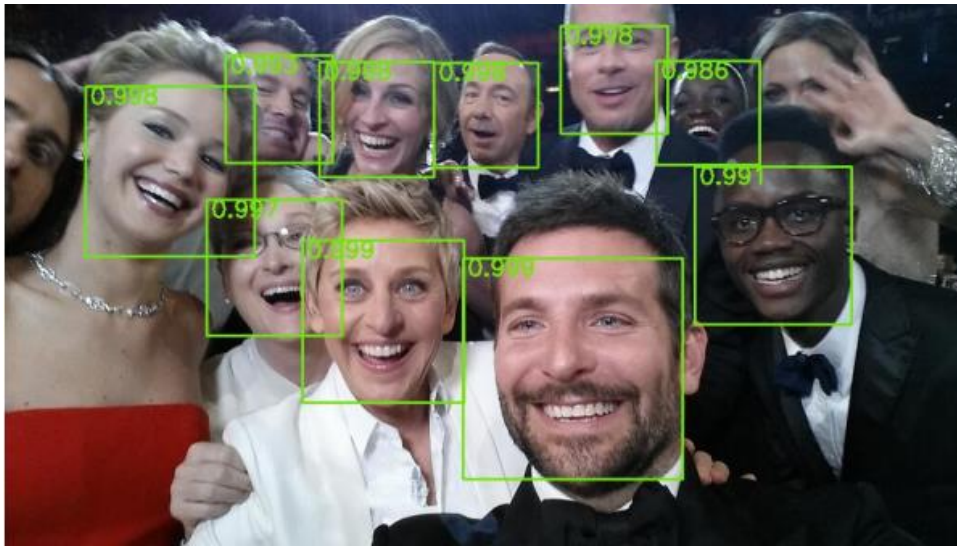
$$= J(x_{22}, y_{22}) - J(x_{12}, y_{12}) - J(x_{21}, y_{21}) + J(x_{11}, y_{11})$$

VIOLA-JONES FACE DETECTION

Extremely fast and accurate face detection
- Running at real-time

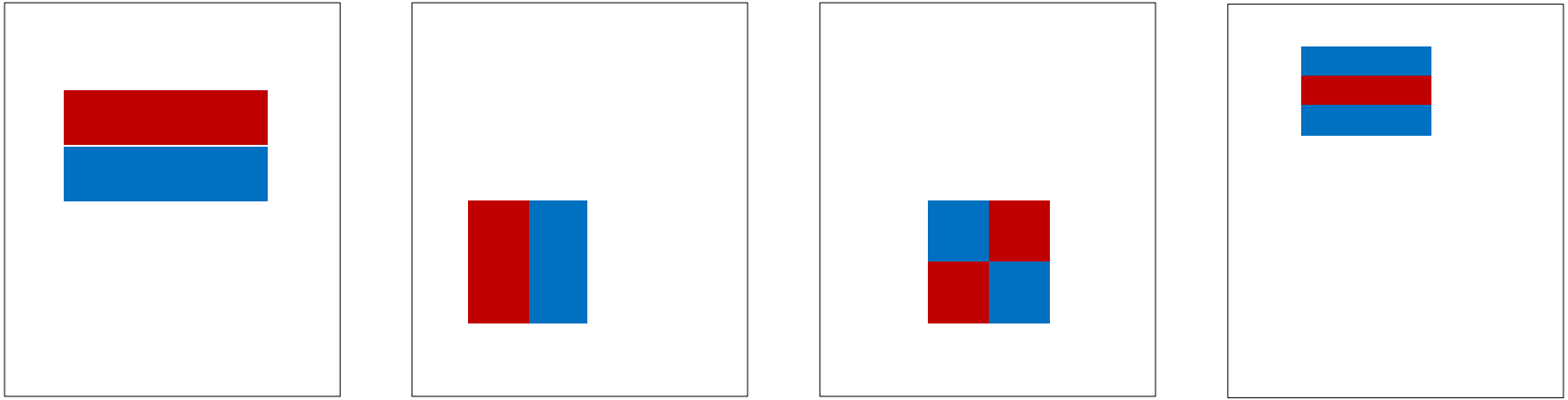
https://www.youtube.com/watch?v=aTErTqOlkss

Enabling factors:
- Efficient feature computation
    - Integral image
- **Efficient feature selection**
    - **Minimal filtering operations**
- Efficient inference algorithm



Too Many Features

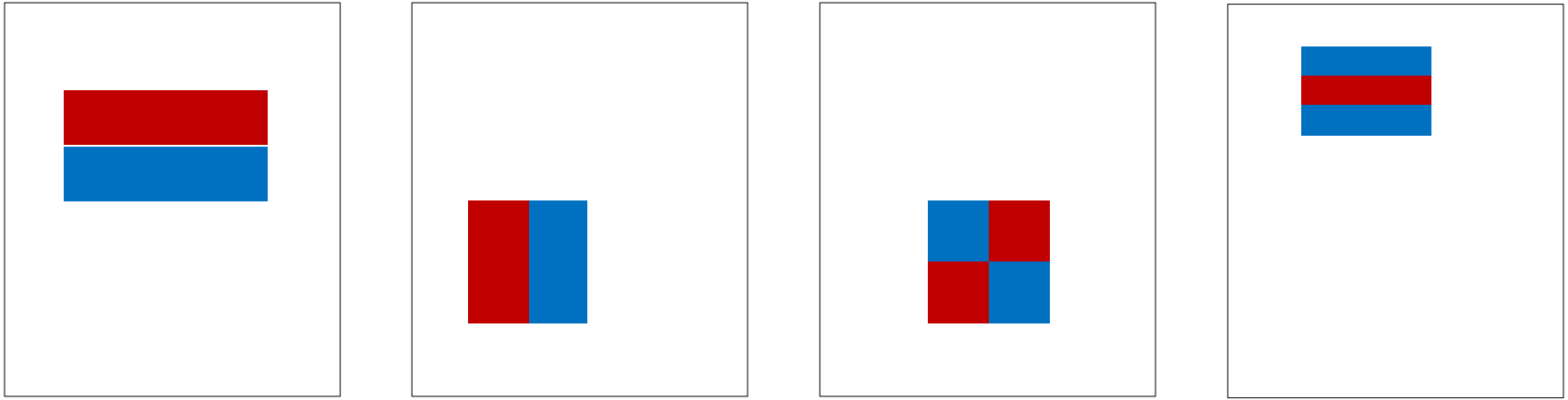
Various derivative filters varying location, size, and combinations



# of possible filters for 24x24 patch: ~160k

Feature Selection

Various derivative filters varying location, size, and combinations

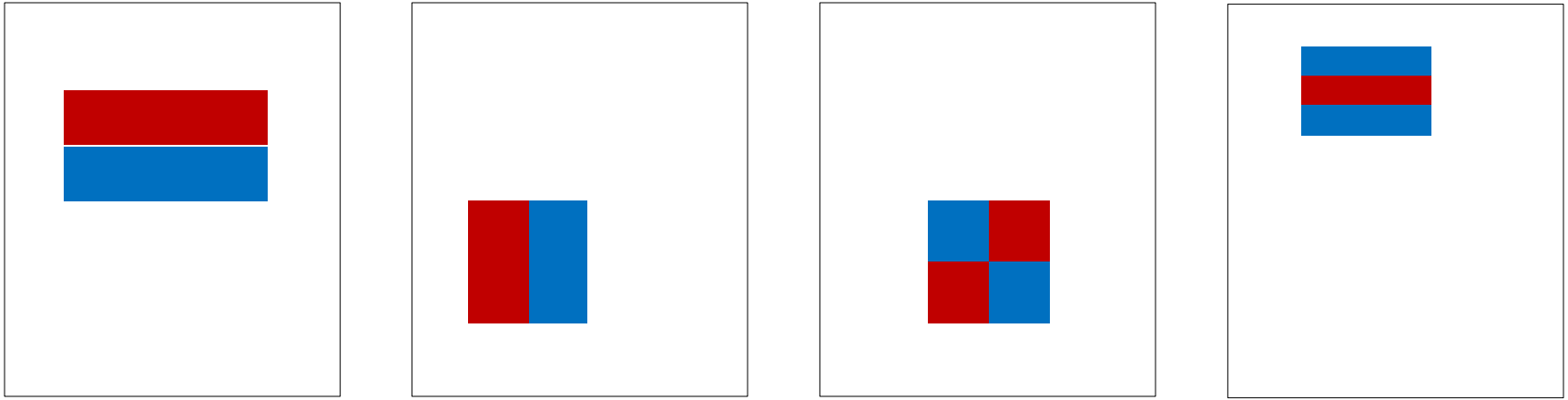


# of possible filters for 24x24 patch: ~160k

Can we choose a set of *good* filters?

Feature Selection

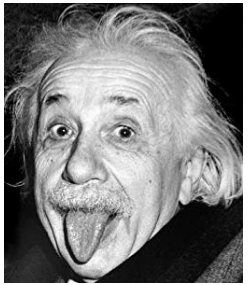
Various derivative filters varying location, size, and combinations



# of possible filters for 24x24 patch: ~160k

Can we choose a set of *good* filters?

What defines the good filters?



FEATURE SELECTION: BOOSTING

Boosting is a classifier that combines a set of weak classifiers to build a strong classifier.

I

$$= \sum \quad - \quad = f(I)$$

threshold

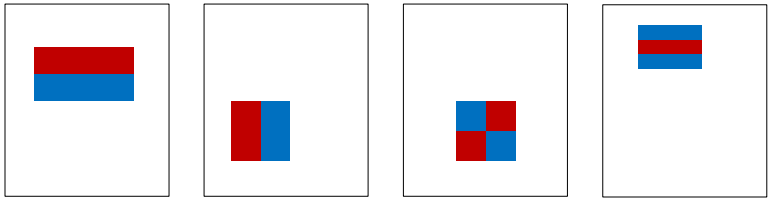
$$h_t(I) = \begin{cases} +1 & p_t f(I) > p_t \theta \\ -1 & \text{otherwise} \end{cases}$$

Each filter can be a weak classifier.

$p_t \in \{-1, 1\}$: to change the direction of threshold

FEATURE SELECTION: BOOSTING

Boosting is a classifier that combines a set of weak classifiers to build a strong classifier.



$$h(I) = \sigma\left(\sum_t \alpha_t h_t(I)\right)$$

Ensemble classifier

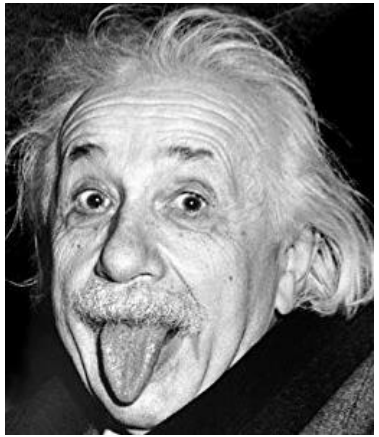
threshold

$$h_t(I) = \begin{cases} +1 & p_t f(I) > p_t \theta \\ -1 & \text{otherwise} \end{cases}$$

$p_t \in \{-1, 1\}$: to change the direction of threshold

FEATURE SELECTION: BOOSTING

Boosting is a classifier that combines a set of weak classifiers to build a strong classifier.



$$h_1 = 1$$
$$h_2 = -1$$
$$h_3 = 1$$
$$h_4 = 1$$

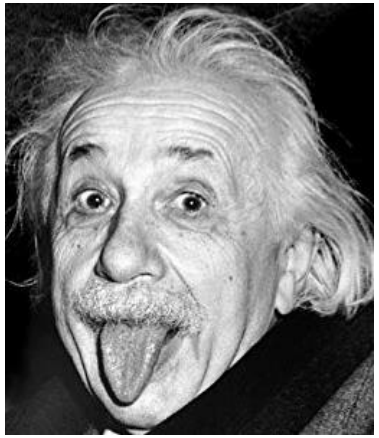
$$h_1 = -1$$
$$h_2 = -1$$
$$h_3 = 1$$
$$h_4 = 1$$

$$h(I) = \sigma\left(\sum_t \alpha_t h_t(I)\right)$$

$$h(I) = \sigma\left(\sum_t \alpha_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Boosting is a classifier that combines a set of weak classifiers to build a strong classifier.



$$h_1 = 1$$
$$h_2 = -1$$
$$h_3 = 1$$
$$h_4 = 1$$

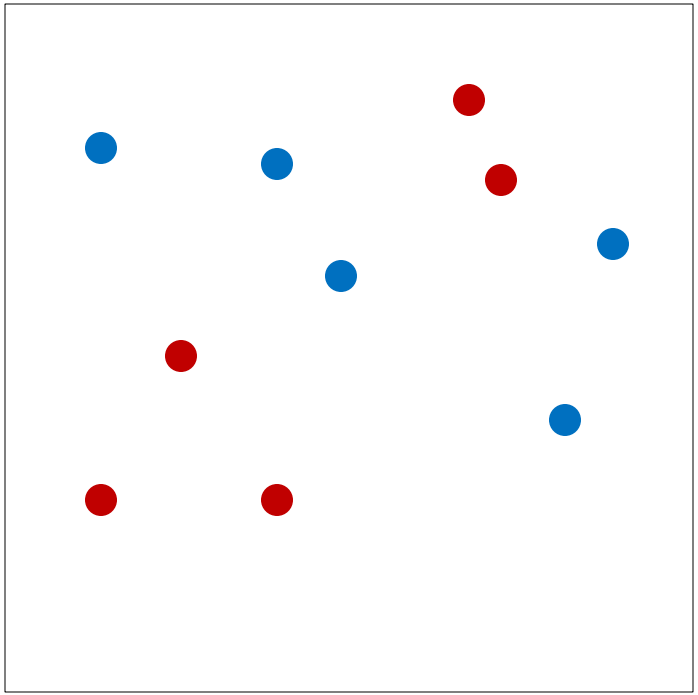
$$h_1 = -1$$
$$h_2 = -1$$
$$h_3 = 1$$
$$h_4 = 1$$

$$h(I) = \sigma\left(\sum_t \alpha_t h_t(I)\right)$$

$$h(I) = \sigma\left(\sum_t \alpha_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

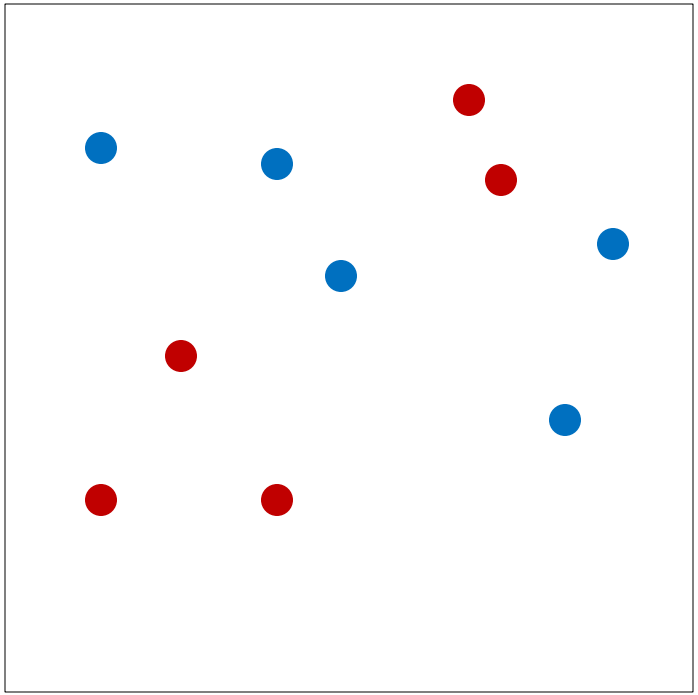
Round 1



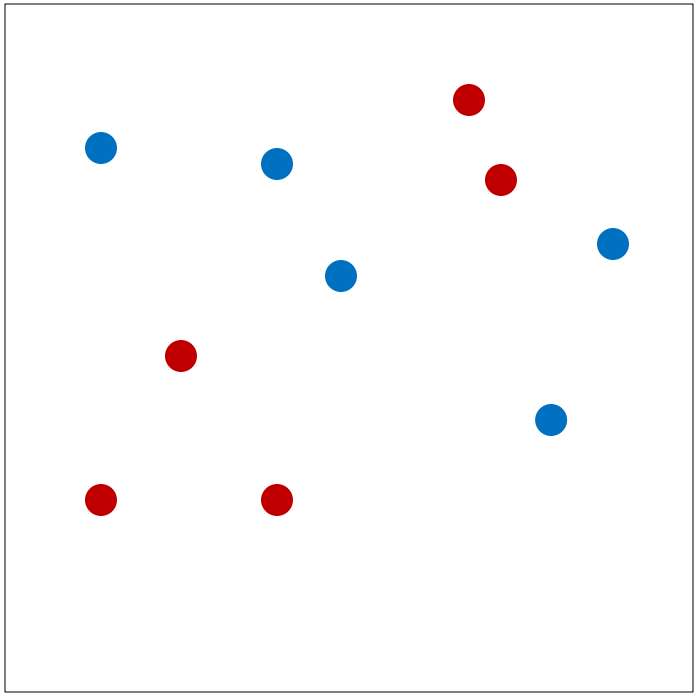
$h_1(I)$

FEATURE SELECTION: BOOSTING

Round 1



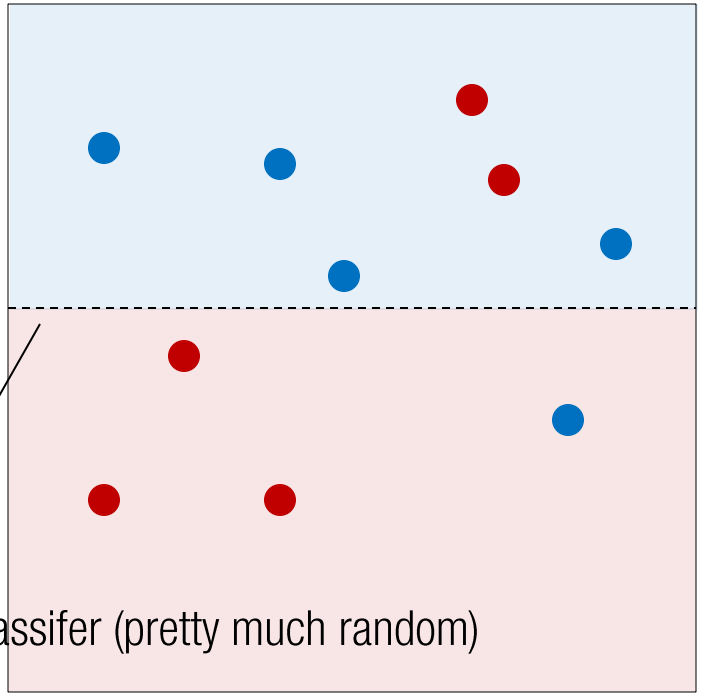
$h_1(I)$



$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

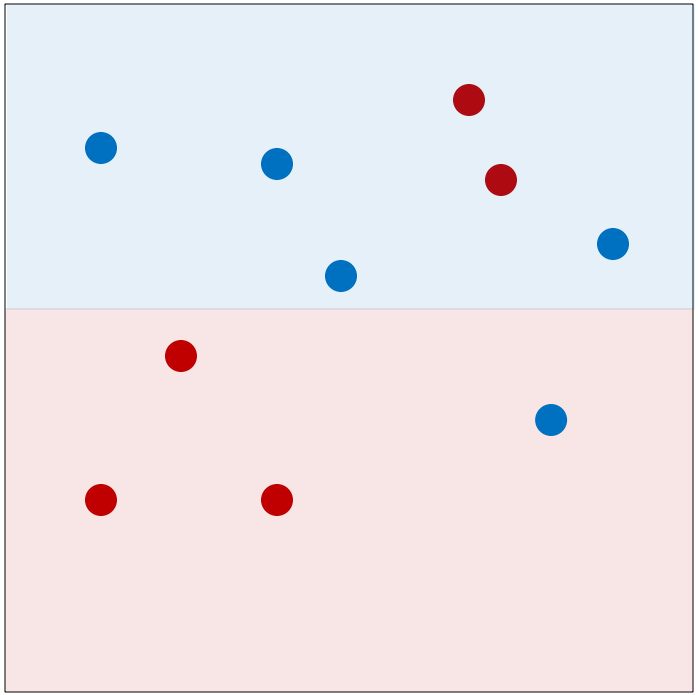
FEATURE SELECTION: BOOSTING

Round 1



Weak classifer (pretty much random)

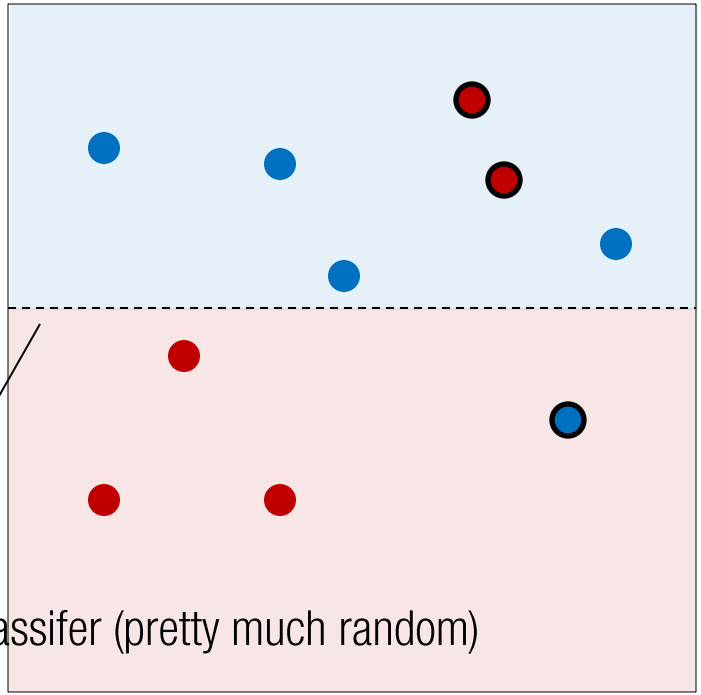
$h_1(I)$



$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

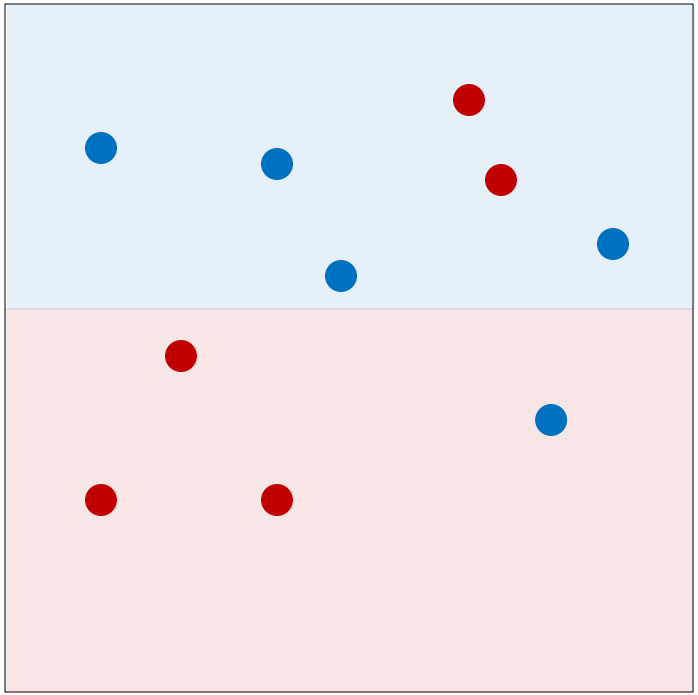
FEATURE SELECTION: BOOSTING

Round 1



Weak classifer (pretty much random)

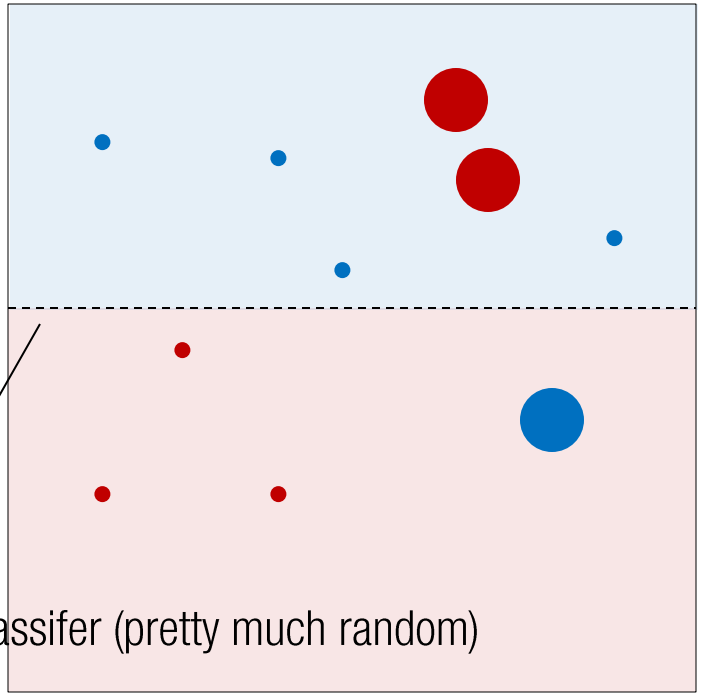
$h_1(I)$



$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

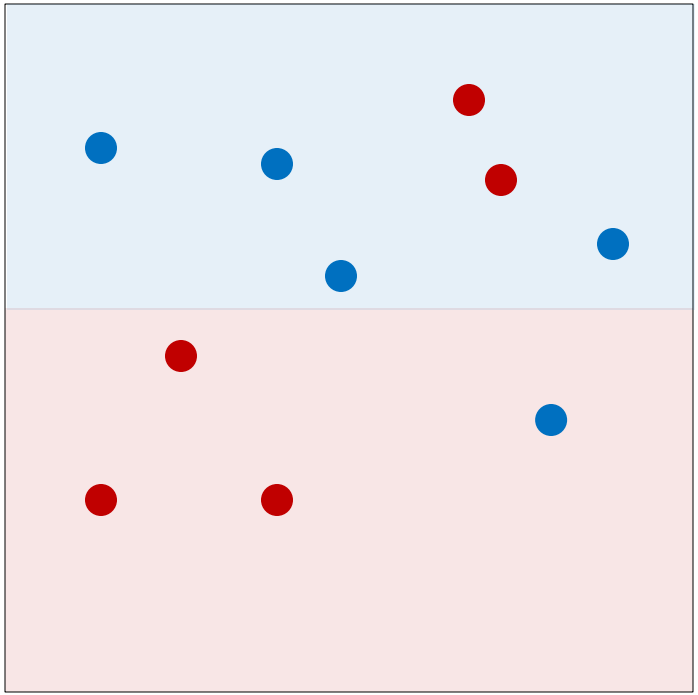
FEATURE SELECTION: BOOSTING

Round 1



Weak classifer (pretty much random)

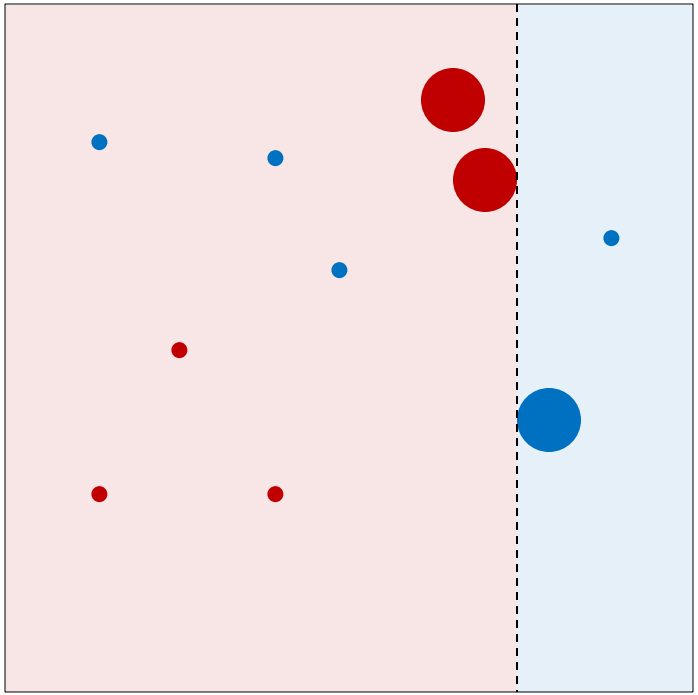
$h_1(I)$



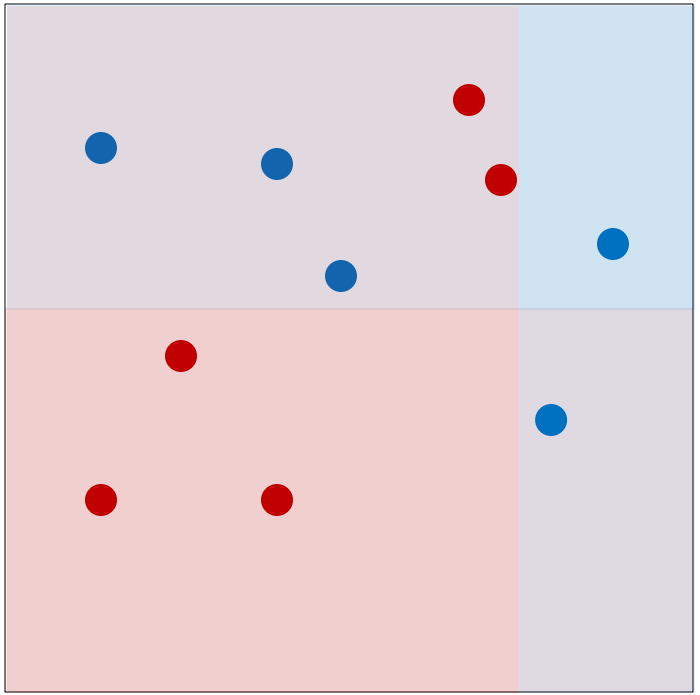
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 2



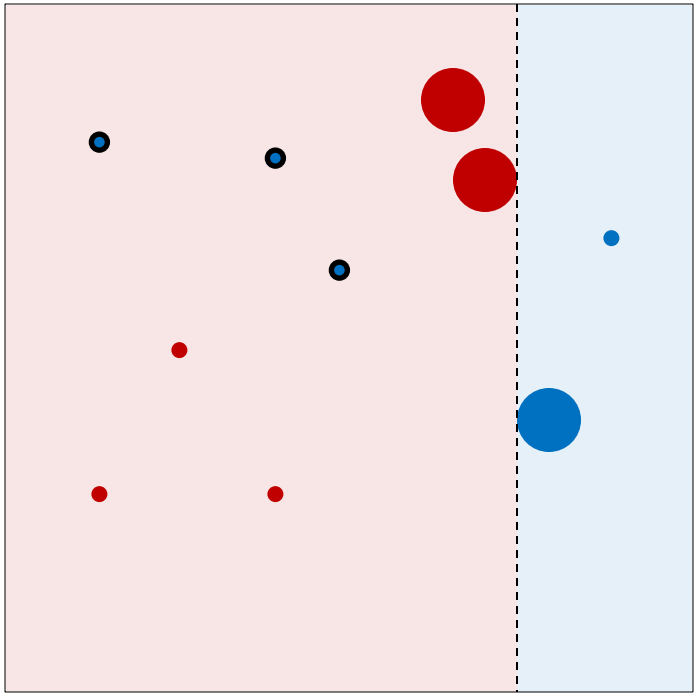
$h_2(I)$



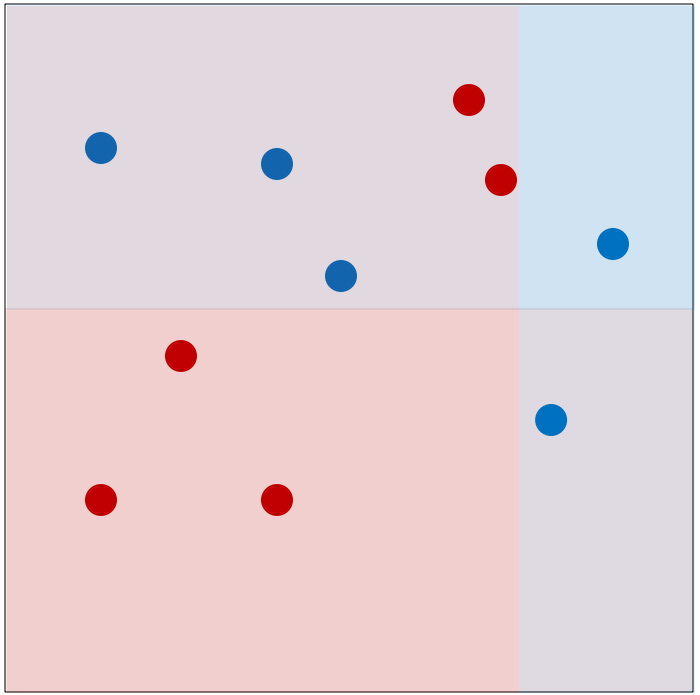
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 2



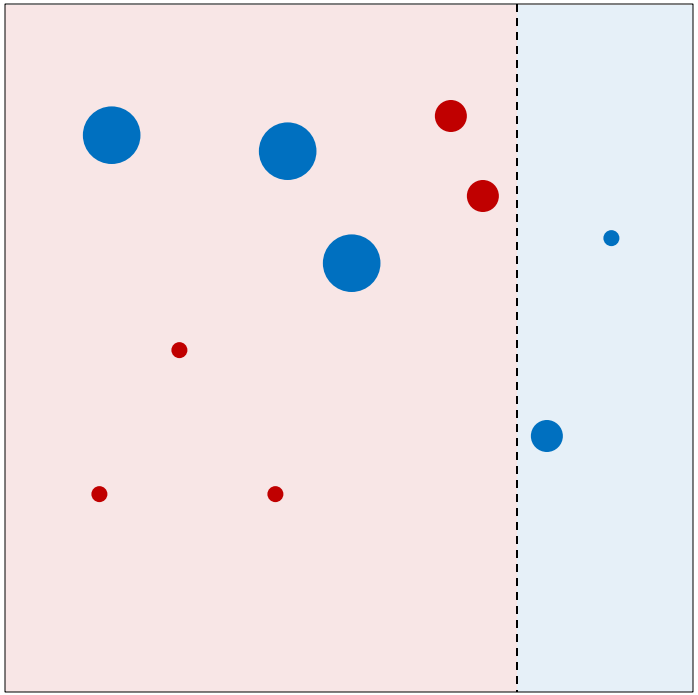
$h_2(I)$



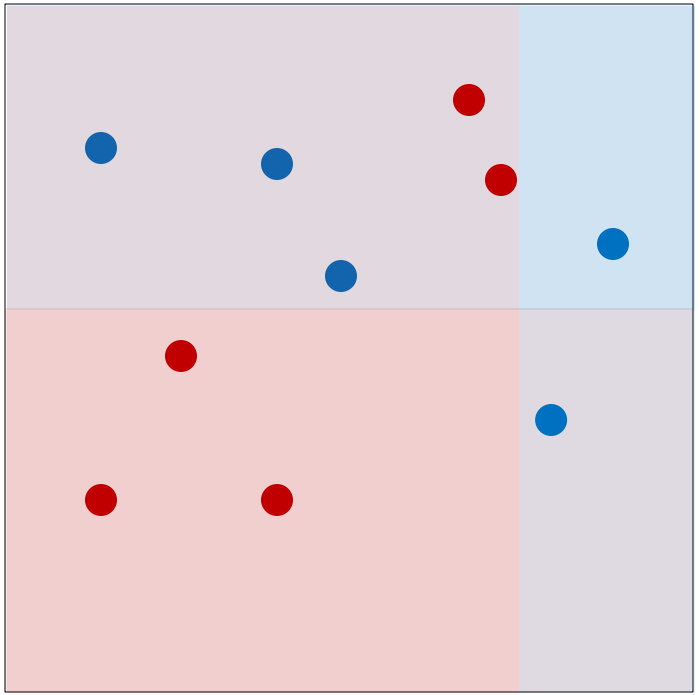
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 2



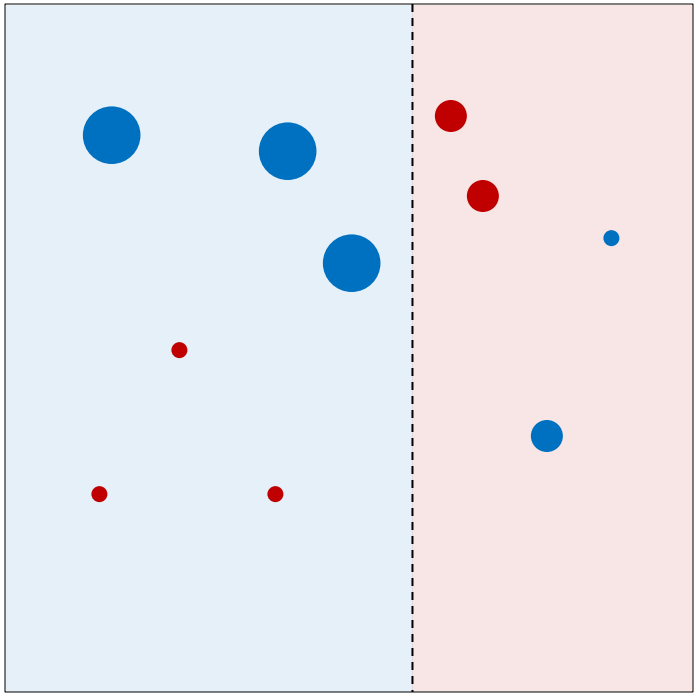
$h_2(I)$



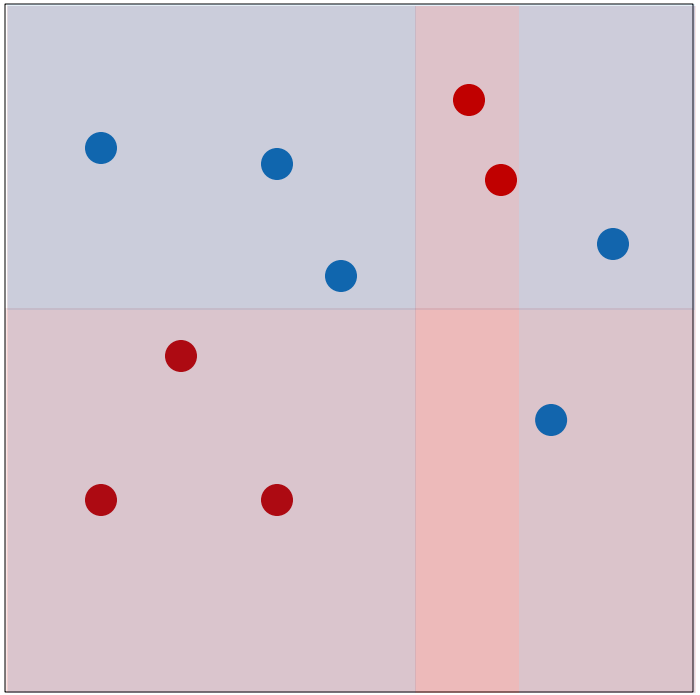
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 3



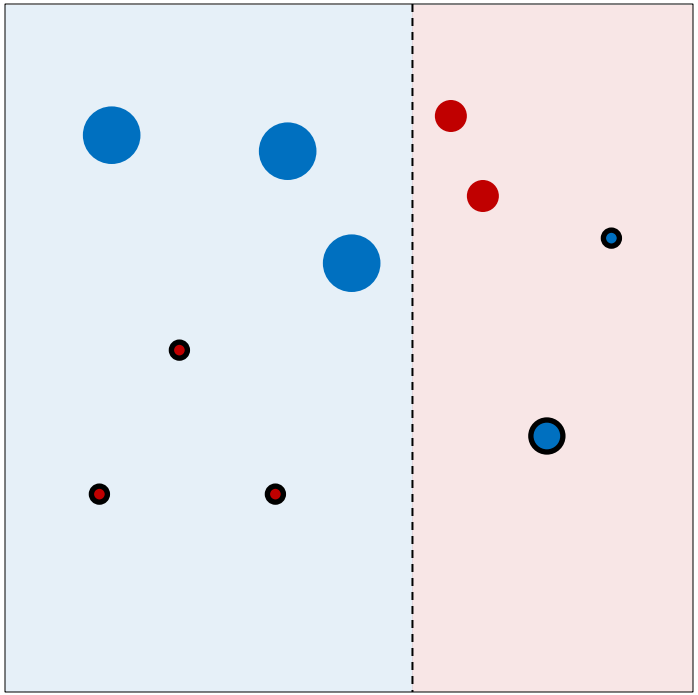
$h_3(I)$



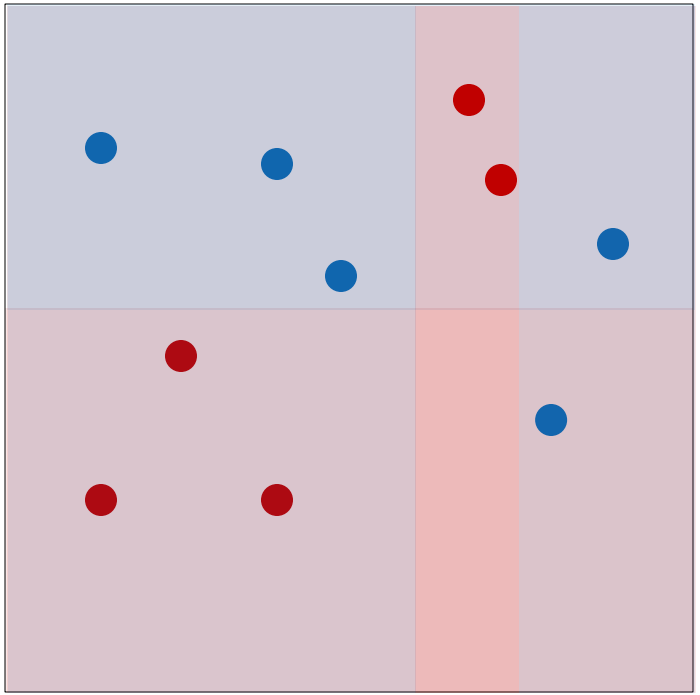
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 3



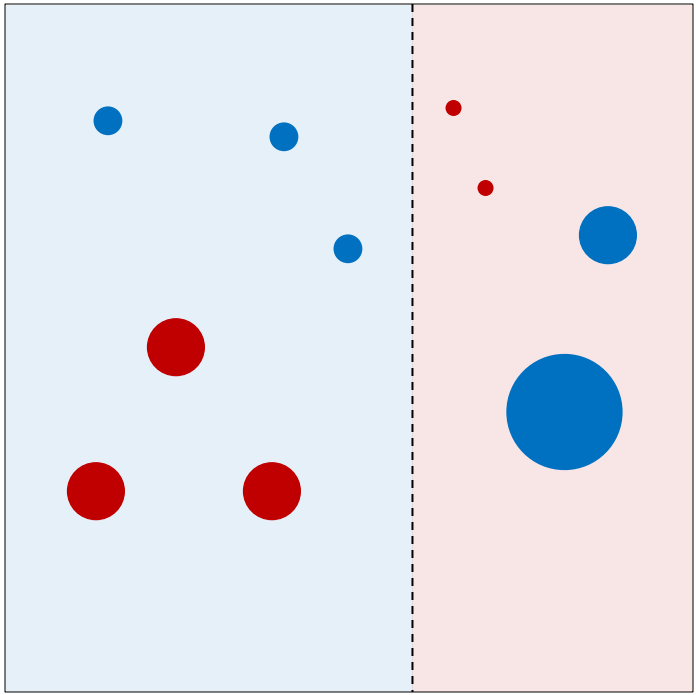
$h_3(I)$



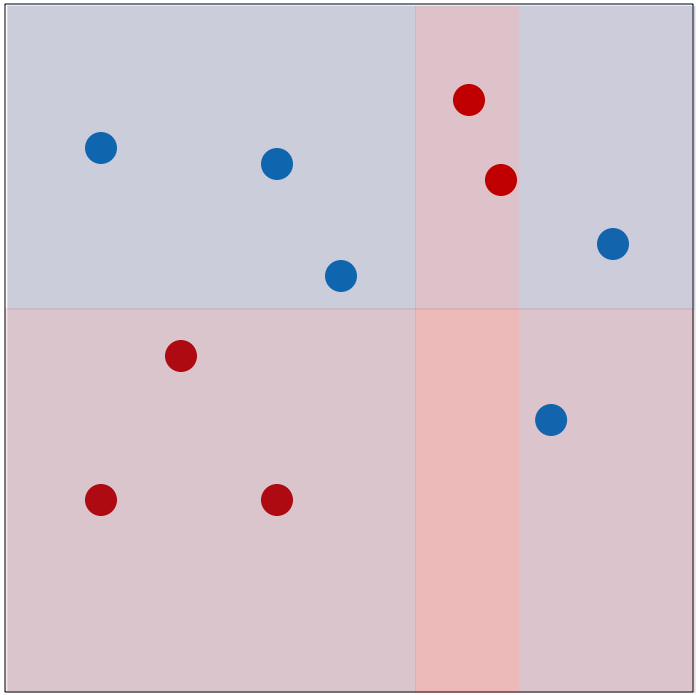
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 3



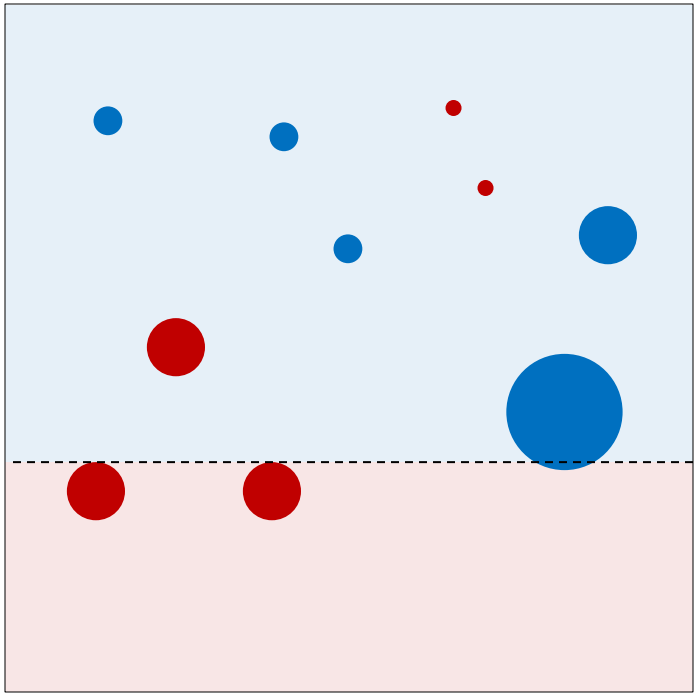
$h_3(I)$



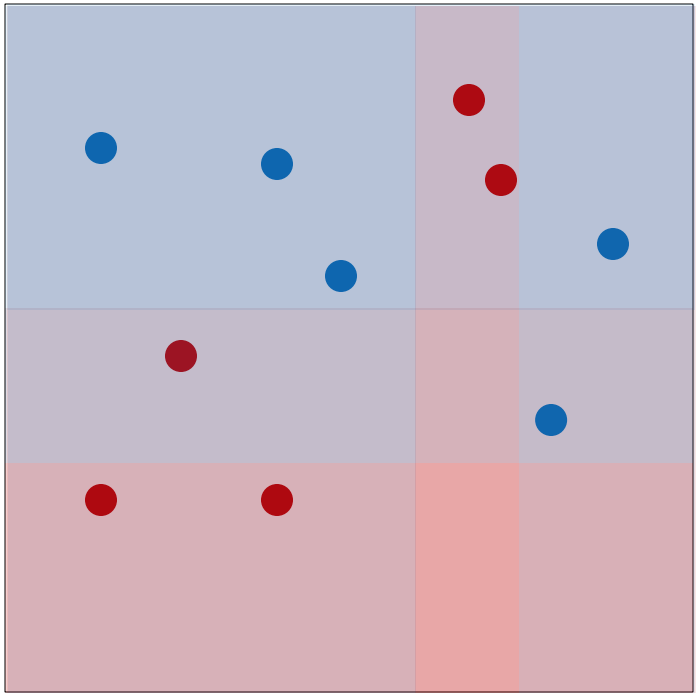
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 4



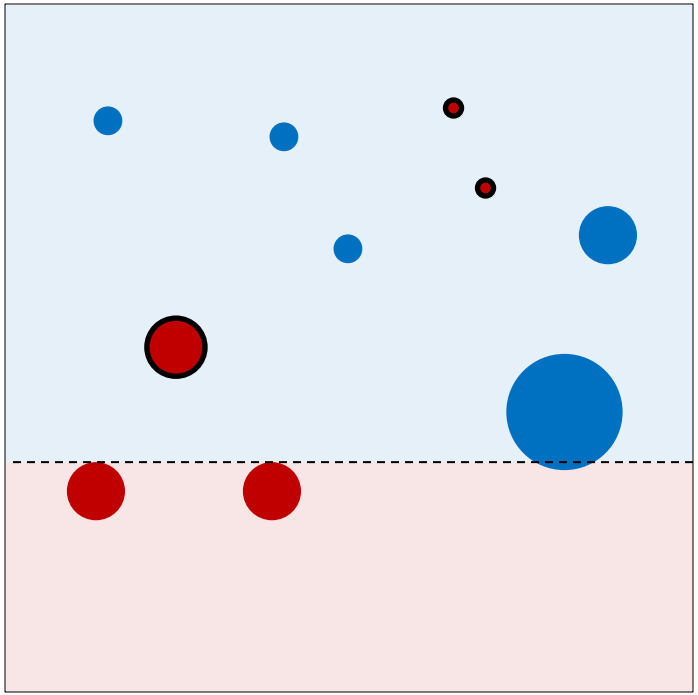
$h_4(I)$



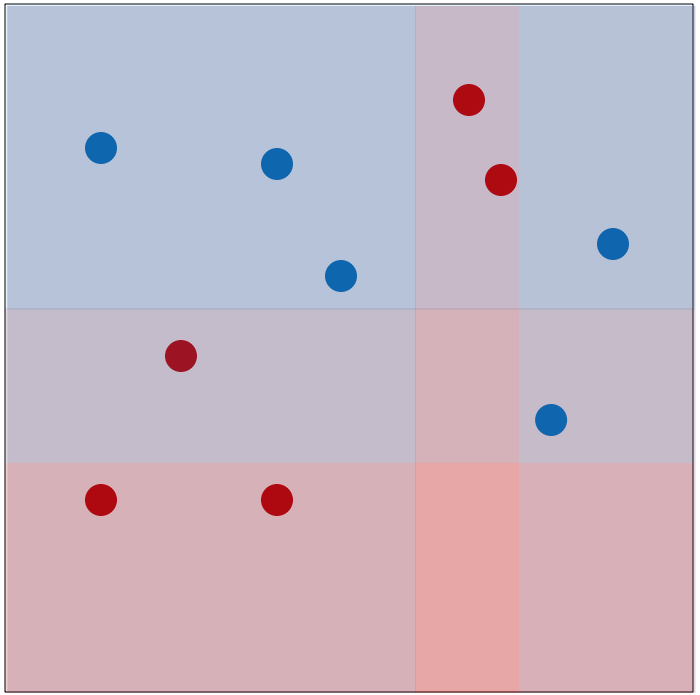
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 4



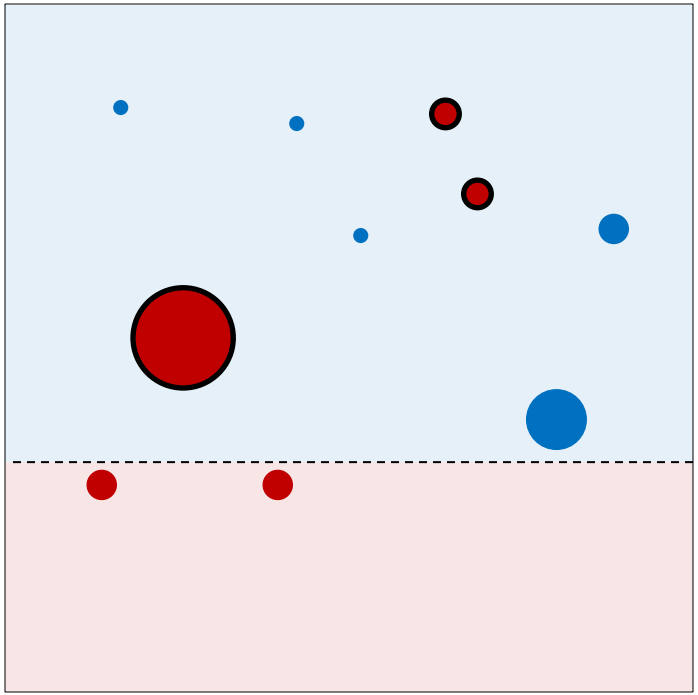
$h_4(I)$



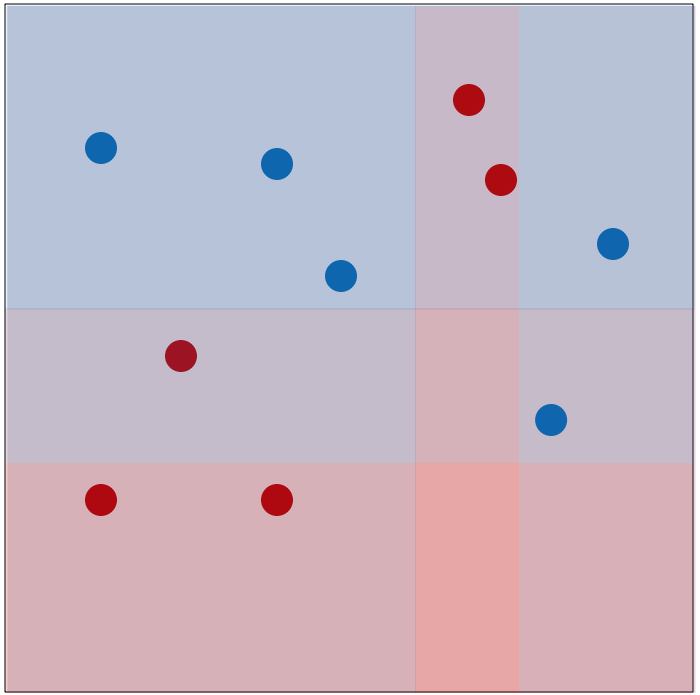
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 4



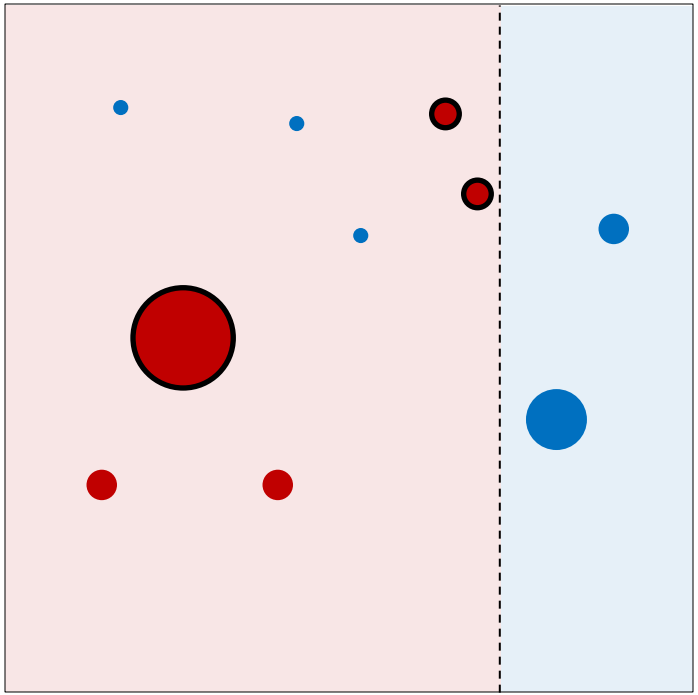
$h_4(I)$



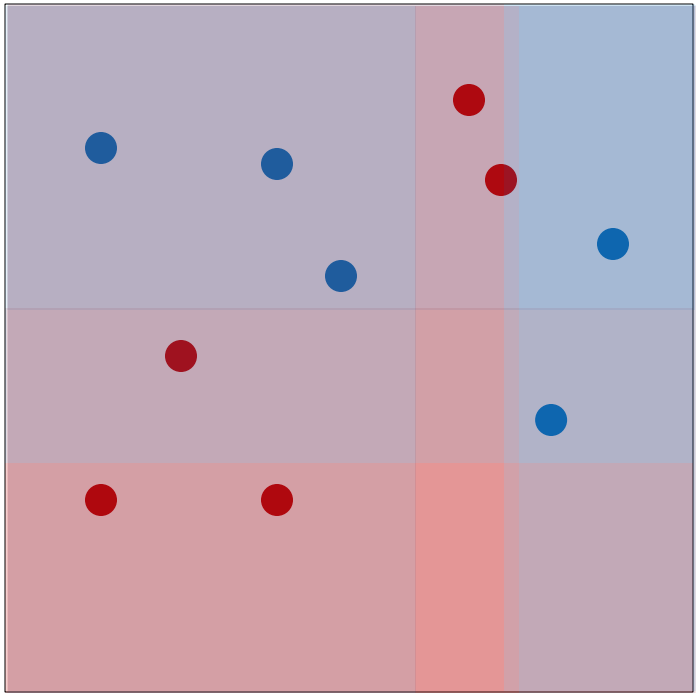
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 5



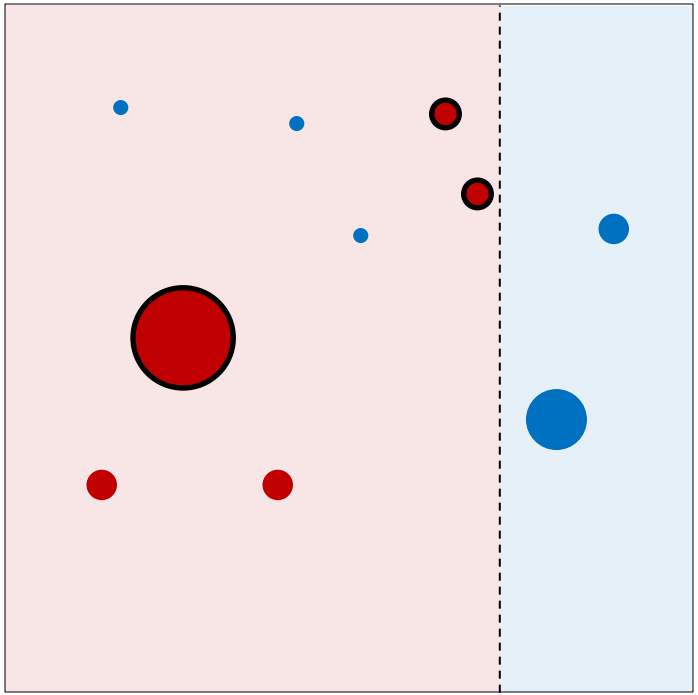
$h_5(I)$



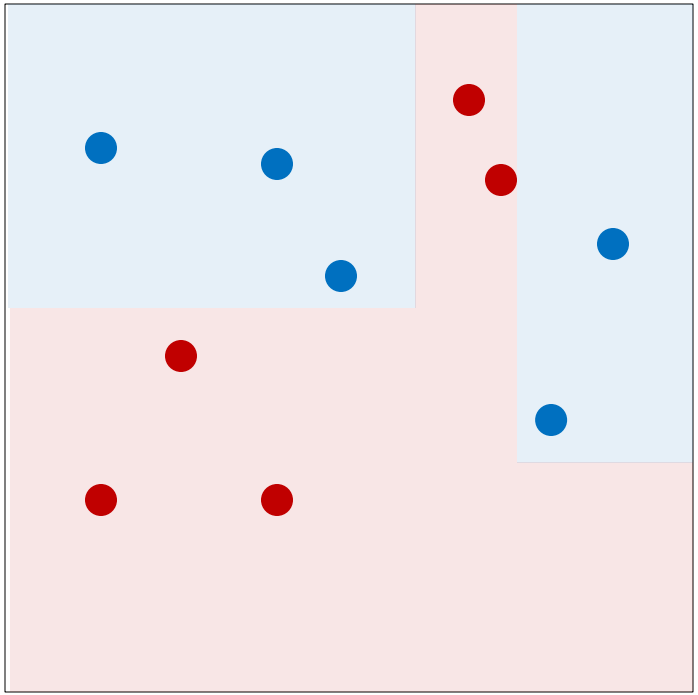
$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

FEATURE SELECTION: BOOSTING

Round 5



$h_5(I)$



$$h(I) = \sigma\left(\sum_t h_t(I)\right)$$

https://www.youtube.com/watch?v=k4G2VCuOMMg

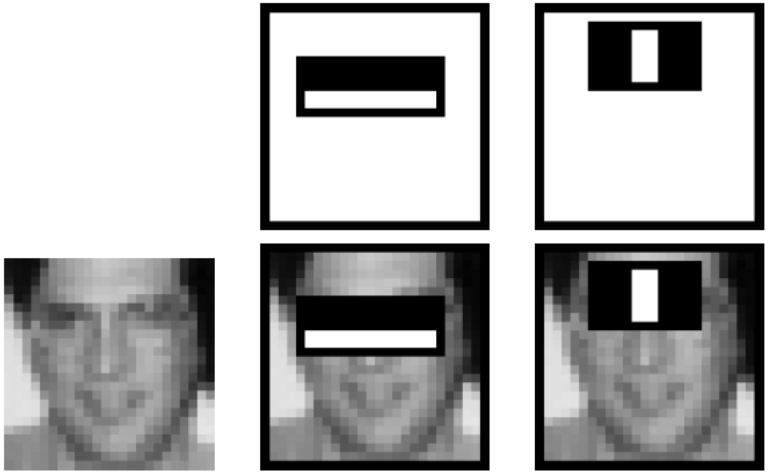
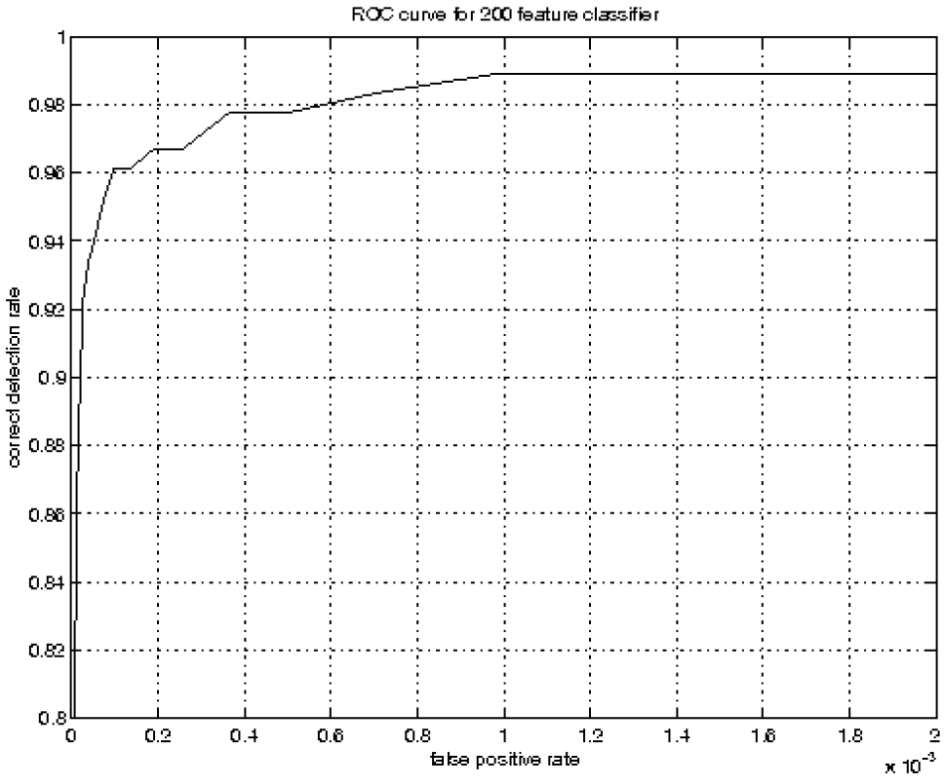


Figure 3: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.



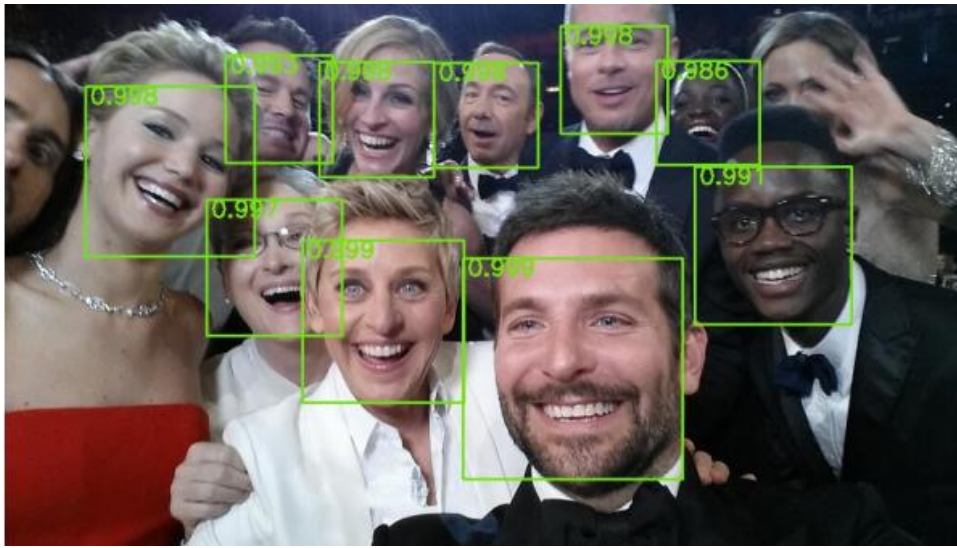
ROC curve for 200 feature classifier

VIOLA-JONES FACE DETECTION

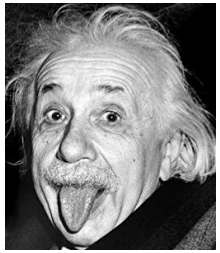
Extremely fast and accurate face detection
- Running at real-time

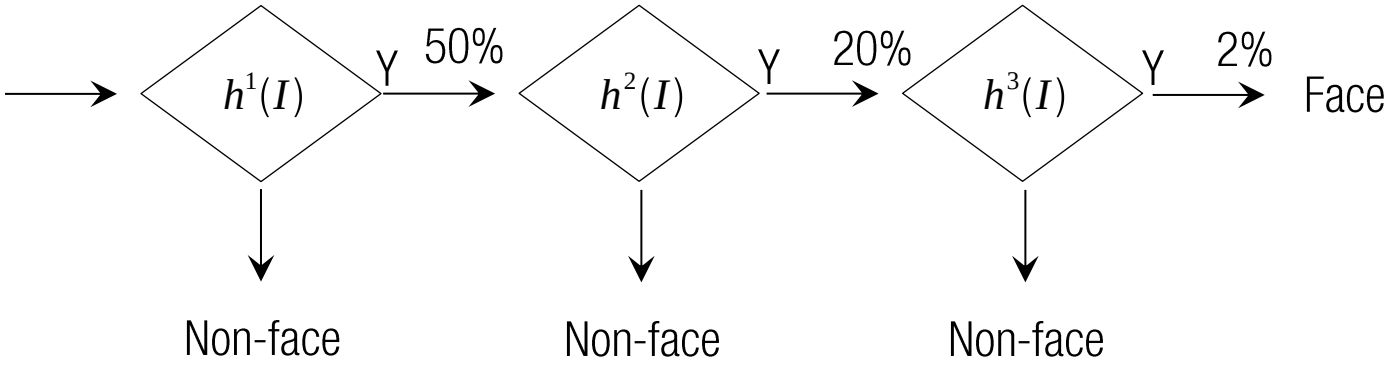
https://www.youtube.com/watch?v=aTErTqOlkss

Enabling factors:
- Efficient feature computation
  - Simple filtering operations
- Efficient feature selection
  - Minimal filtering operations
- **Efficient inference algorithm**
  - **Early rejection of non-face patches**



CASCADE CLASSIFIER





$h^1(I)$: acheives 100% detection rate and 50% false positive rate

$h^2(I)$: acheives 100% detection rate and 40% false positive rate (cummulative 20%)

$h^3(I)$: acheives 100% detection rate and 10% false positive rate (cummulative 2%)

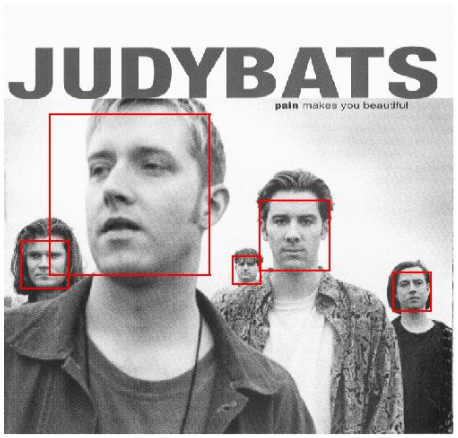
90% detection rate and 1E-6 false positive rate can be achieved by 10 cascade classifiers.

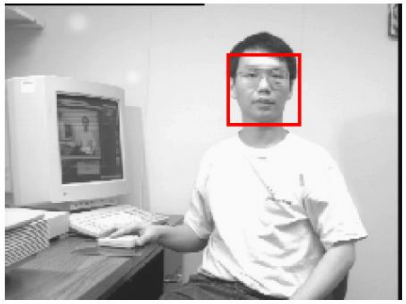
Training Data

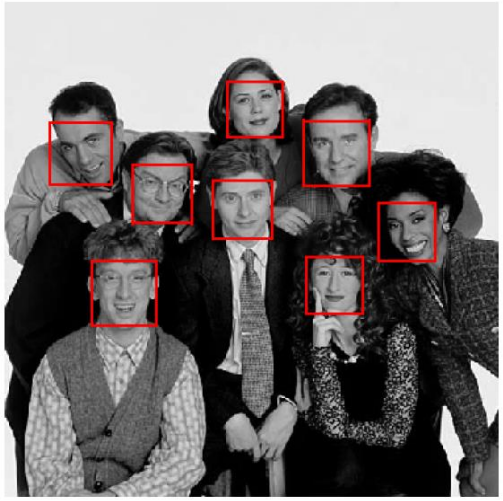
5000 faces (frontal and normalized)
300 million non-faces

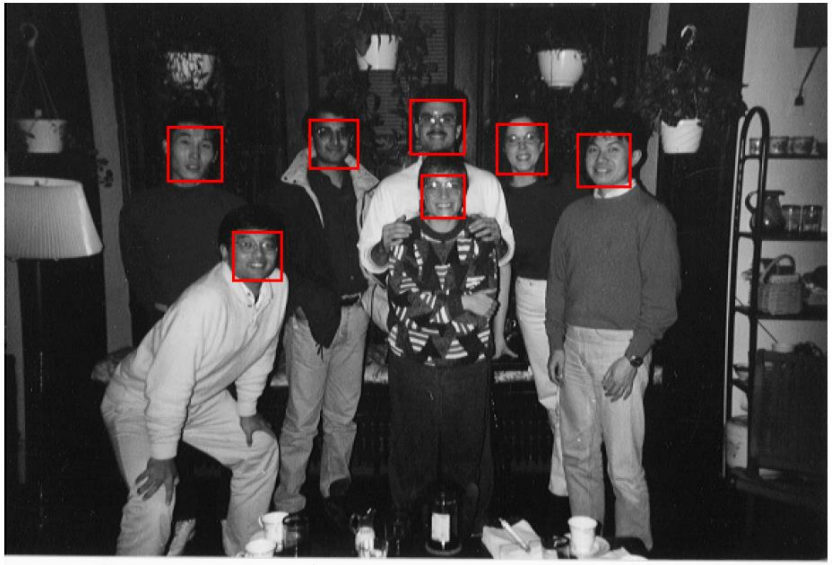
Resulting Detection Algorithm

- Training time: “weeks” on 466 MHz Sun workstation
- 38 layers, total of 6061 features
- Average of 10 features evaluated per window on test set
- “On a 700 Mhz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds”
  - 15 Hz
  - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)

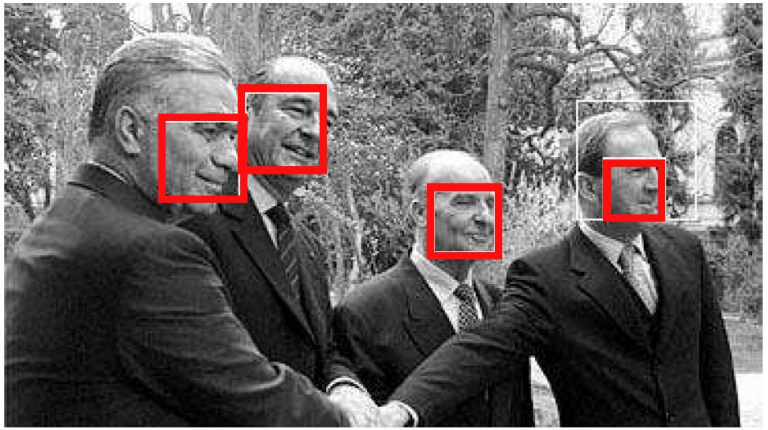

JUDYBATS
pain makes you beautiful





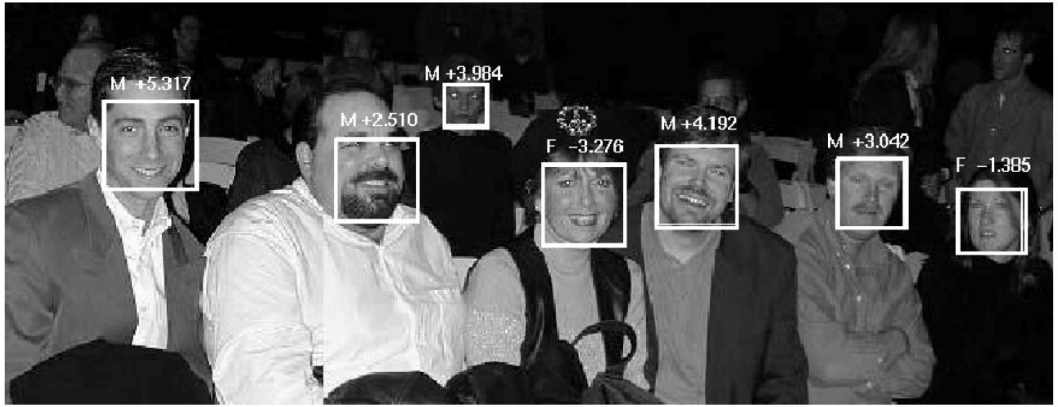


Facial Feature Localization

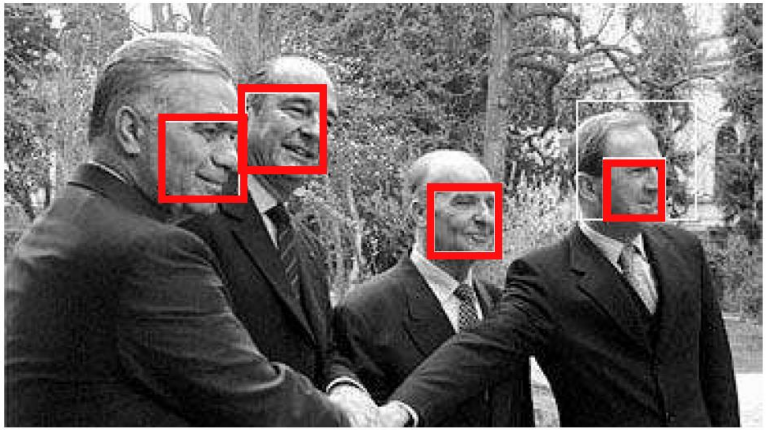


Profile Detection

Demographic Analysis

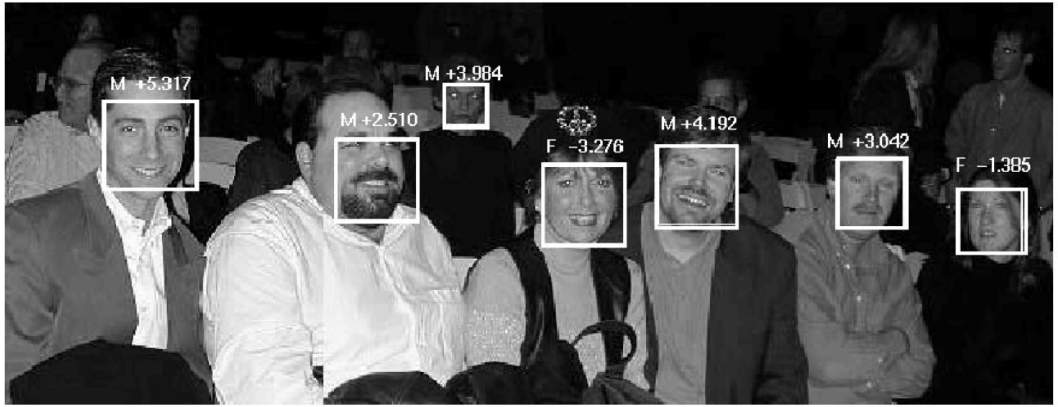


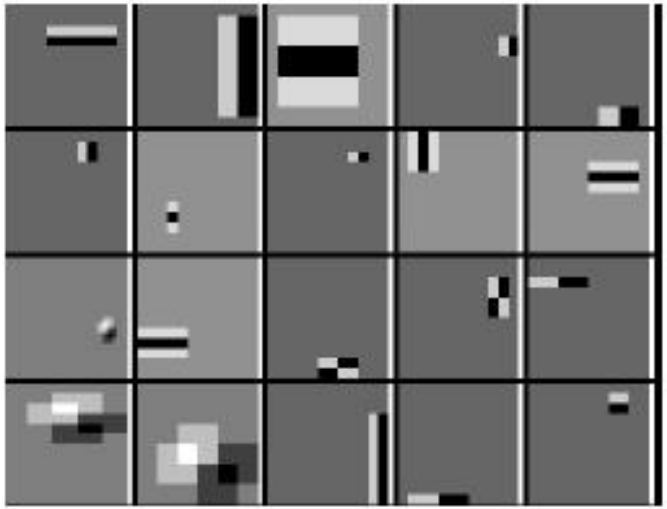
Facial Feature Localization

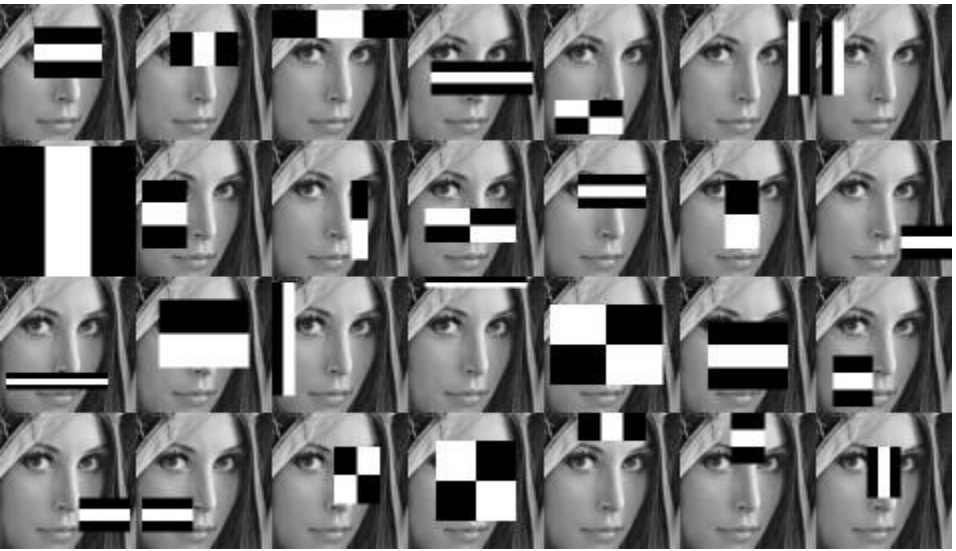


Profile Detection

Demographic Analysis







https://www.youtube.com/watch?v=hPCTwxF0qf4