Object Detection using a Cascade of Classifiers

Mike Jones

Mitsubishi Electric Research Laboratories

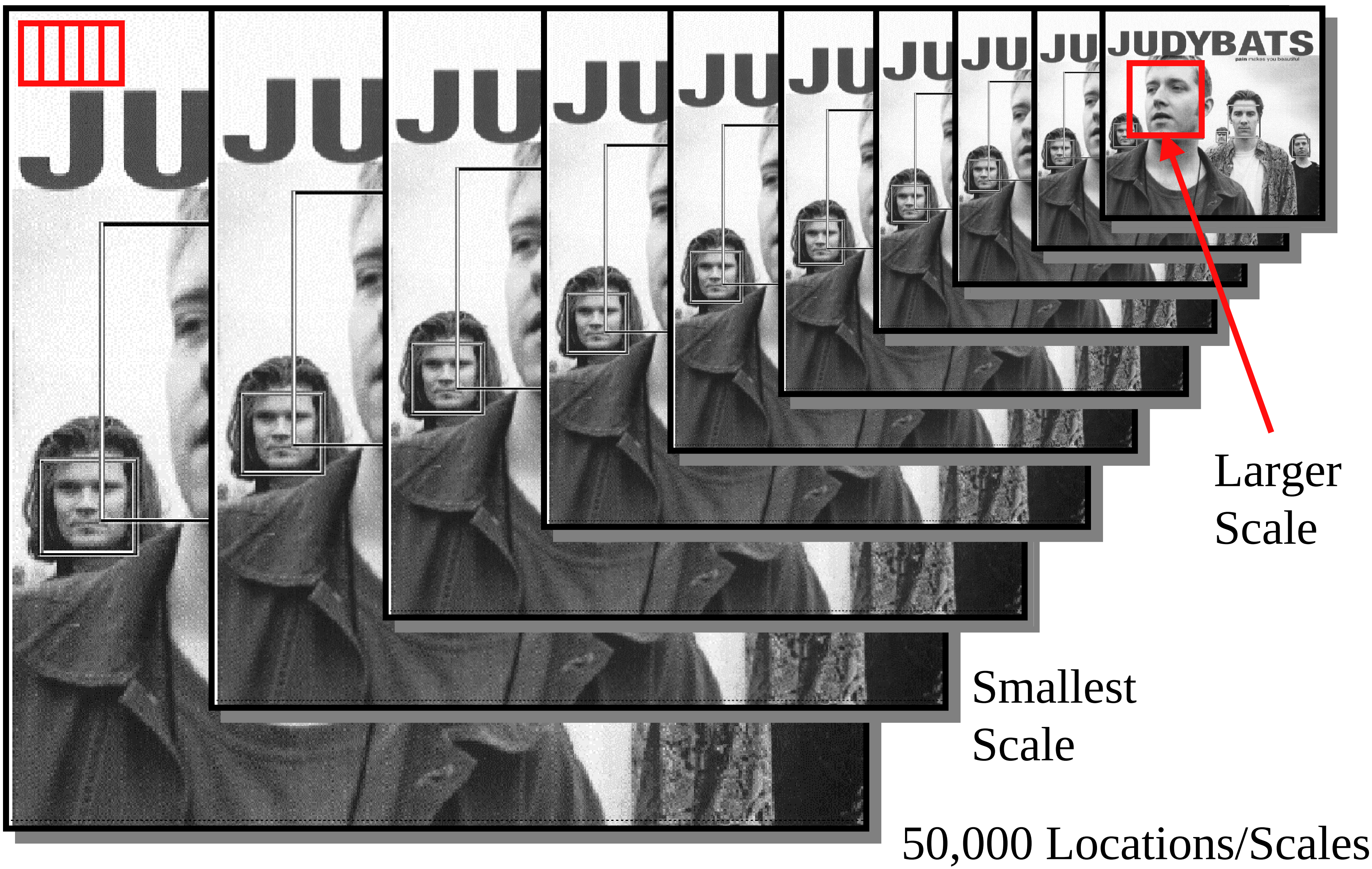
in collaboration with Paul Viola and Dan Snow

The Object Detection Problem

Previous Work

- Mixture of Gaussians - Sung and Poggio (MIT)
- Neural network - Rowley, Baluja and Kanade (CMU)
- Naive Bayes - Schneiderman and Kanade (CMU)
- SVM - Papageorgiou and Poggio (MIT)
- Partial eval of SVMs - Romdhani, Torr, Scholkopf, Blake (Microsoft)
- Color models – many authors

The Classical Face Detection Process



Classifier is Learned from Labeled Data

- Training Data
  - 5000 faces
    - All frontal
  - 10^8 non faces
  - Roughly normalized
    - Scale, translation
- Many variations
  - Across individuals
  - Illumination
  - Pose (rotation both in plane and out)

Cascade of classifiers approach

- Detector is a cascade of classifiers
- Classifier is a linear combination of simple features
- Integral image representation used for efficient evaluation of rectangle features
- AdaBoost used for feature selection during training

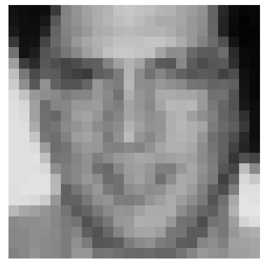
The result is a very efficient and robust detector

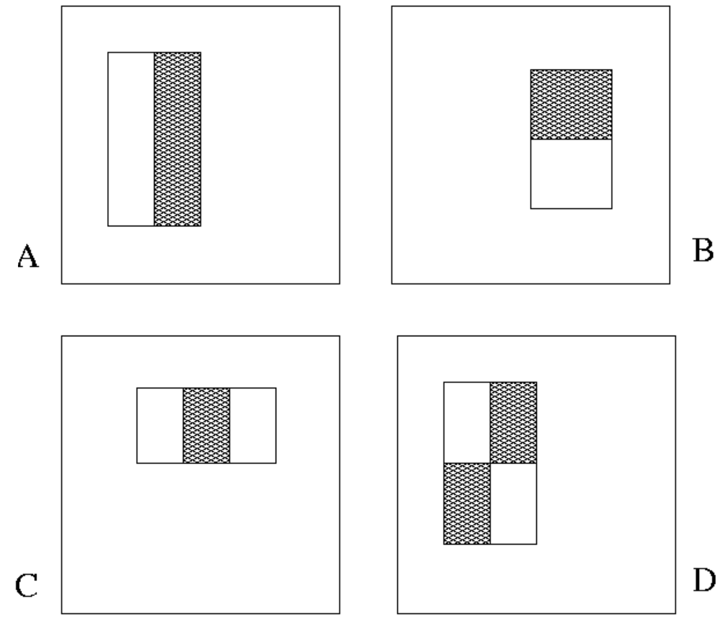
Image Features

“Rectangle filters”

Generalized Haar wavelets

Differences between sums of pixels in adjacent rectangles





A feature is defined as

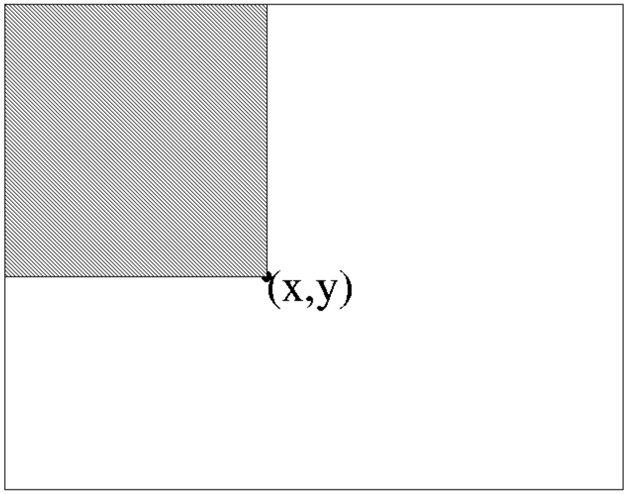
$$h_t(x) = \begin{cases} \alpha_t & \text{if } f_t(x) > \theta_t \\ \beta_t & \text{otherwise} \end{cases}$$

where $f_t(x)$ is a filter.

Integral Image

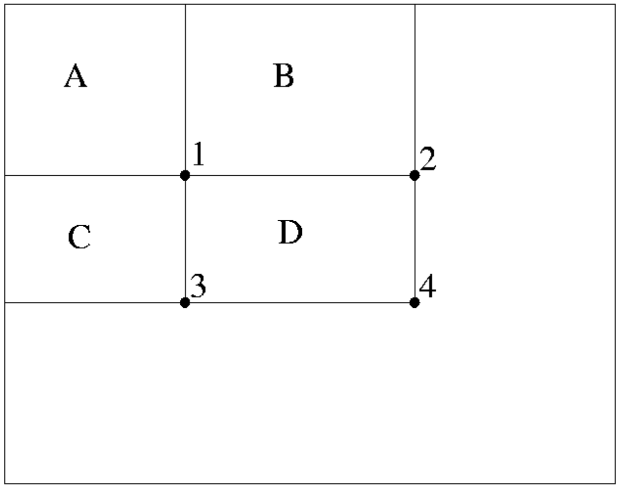
- Define the Integral Image

$$I'(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} I(x',y')$$



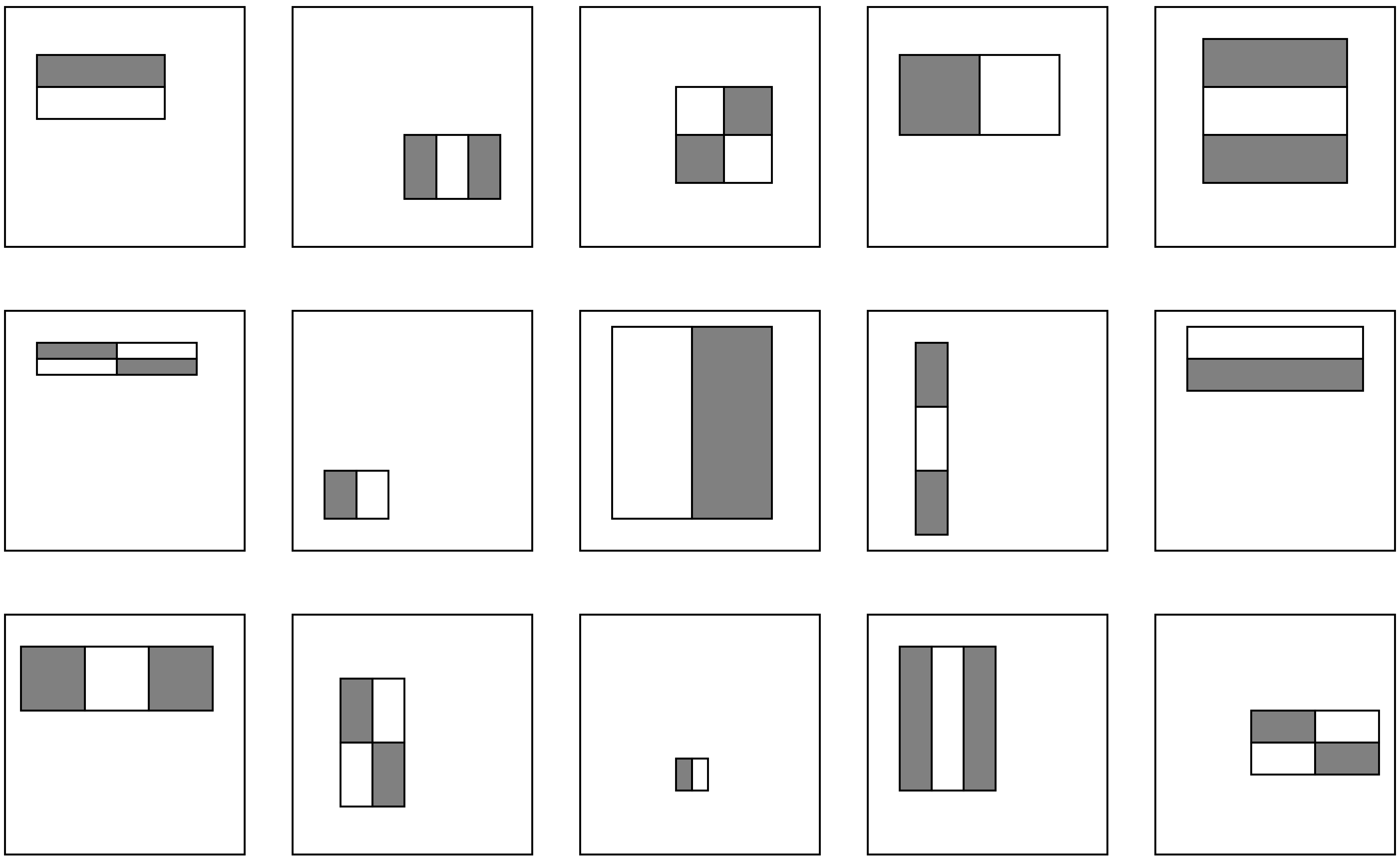
- Any rectangular sum can be computed in constant time:

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



- Rectangle filters can be computed as differences between rectangles

Huge "Library" of Filters



Classifier is linear combination of features

- Perceptron yields a sufficiently powerful classifier

$$C(x) = \begin{cases} 1 & if \ \sum_t h_t(x) > T \\ 0 & otherwise \end{cases}$$

- Use AdaBoost to efficiently choose best features

AdaBoost

(Freund & Schapire '95,
Schapire & Singer '99)

$$C(x) = \theta\left(\sum_t h_t(x)\right)$$

$$h_t(x) = \begin{cases} \alpha_t & \text{if } f_t(x) > \theta_t \\ \beta_t & \text{otherwise} \end{cases}$$

$$\alpha_t = 0.5\log\left(\frac{\sum_{\text{i is correct pos}} w^i}{\sum_{\text{i is false pos}} w^i}\right)$$

$$w_t^i = \frac{w_{t-1}^i e^{-y_i h_t(x_i)}}{\sum_i w_{t-1}^i e^{-y_i h_t(x_i)}}$$

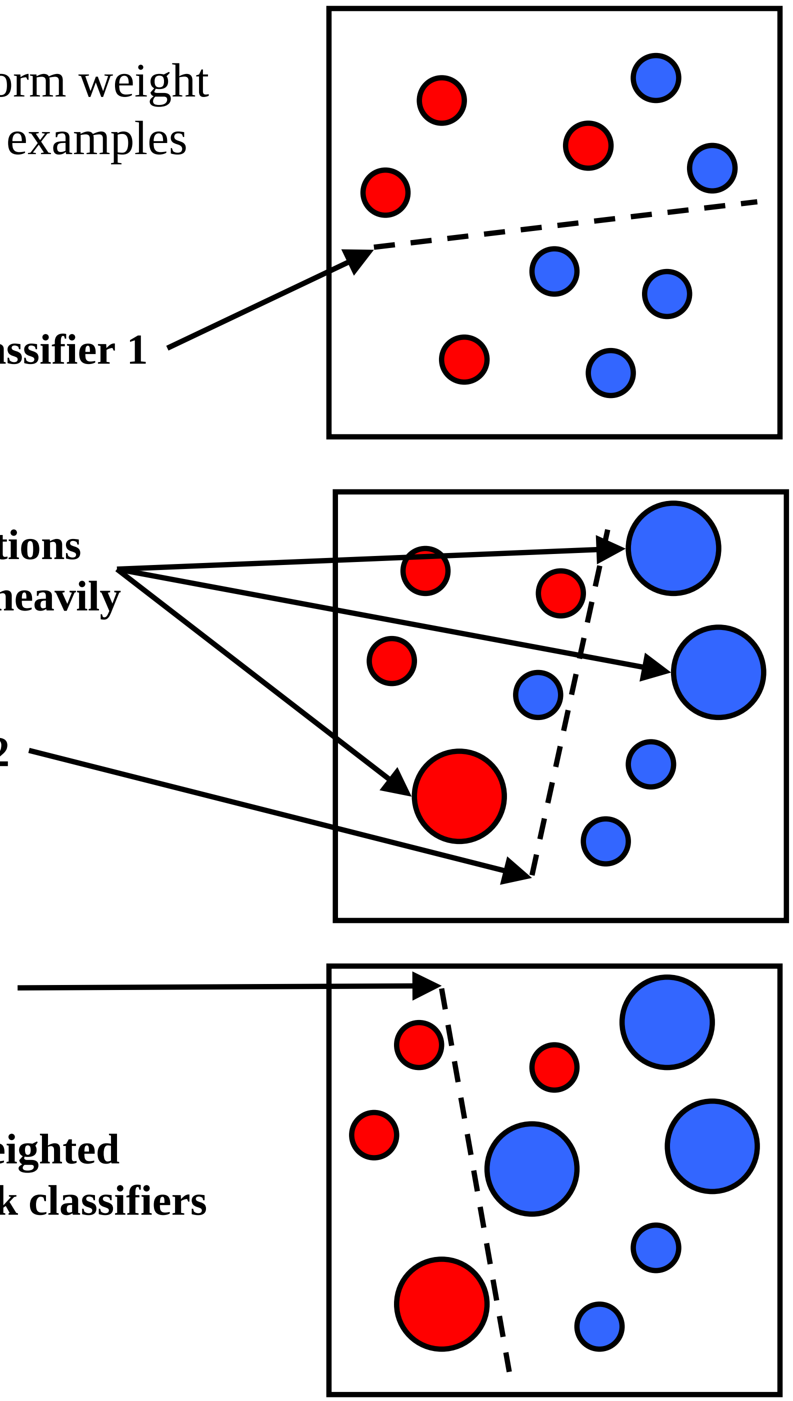
Initial uniform weight
on training examples

weak classifier 1

Incorrect classifications
re-weighted more heavily

weak classifier 2

weak classifier 3



Final classifier is weighted
combination of weak classifiers

AdaBoost for Efficient Feature Selection

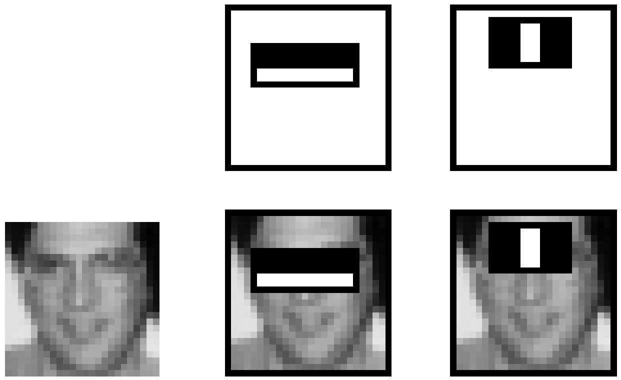
- Our Features = Weak Classifiers
- For each round of boosting:
  - Evaluate each rectangle filter on each example
  - Sort examples by filter values
  - Select best threshold for each filter (min error)
    - Sorted list can be quickly scanned for the optimal threshold
  - Select best filter/threshold combination
  - Weights for this feature are a simple function of weighted errors
  - Reweight examples

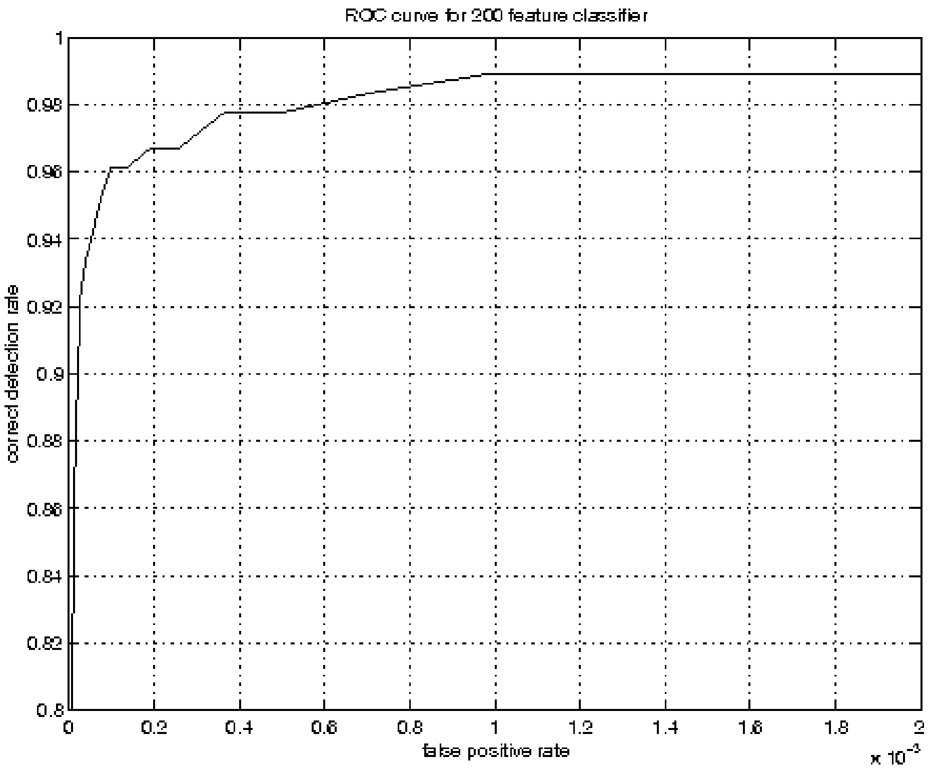
Example Classifier for Face Detection

A classifier with 200 rectangle features was learned using AdaBoost

95% correct detection on test set with 1 in 14084
false positives.

Not quite competitive...

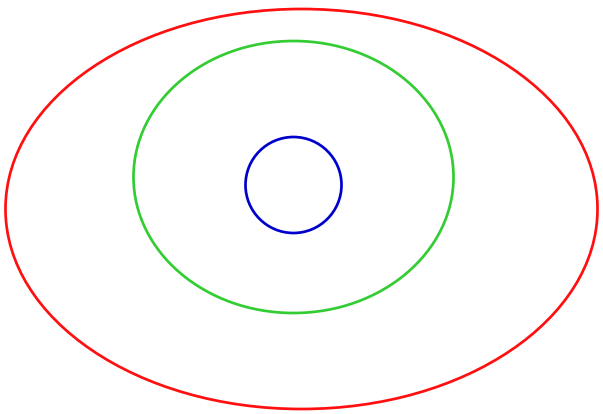


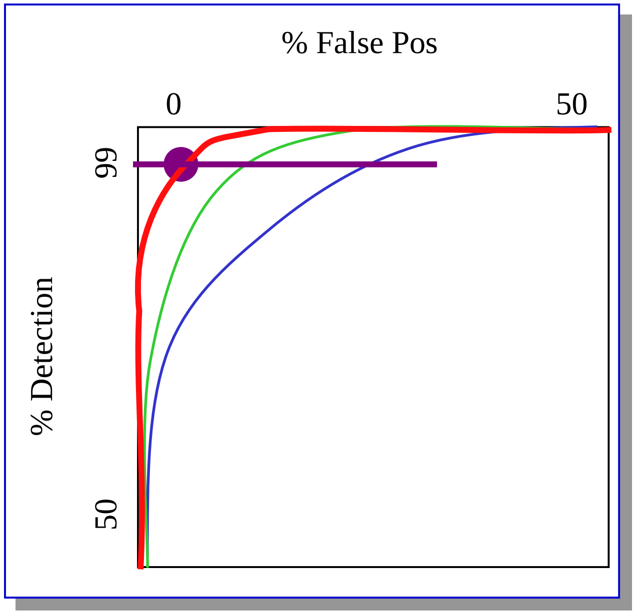


ROC curve for 200 feature classifier

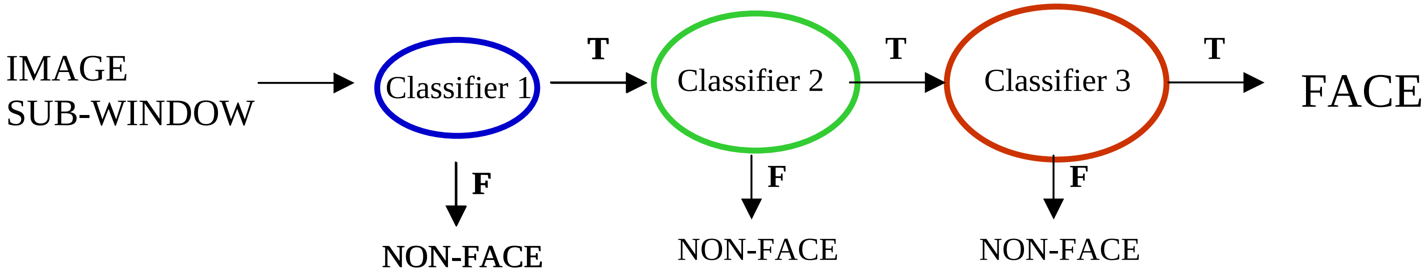
Trading Speed for Accuracy

- Given a nested set of classifier hypothesis classes





- Computational Risk Minimization



Cascade Training

Given initial training set of positive and negative examples

1. Train a strong classifier with N features using AdaBoost

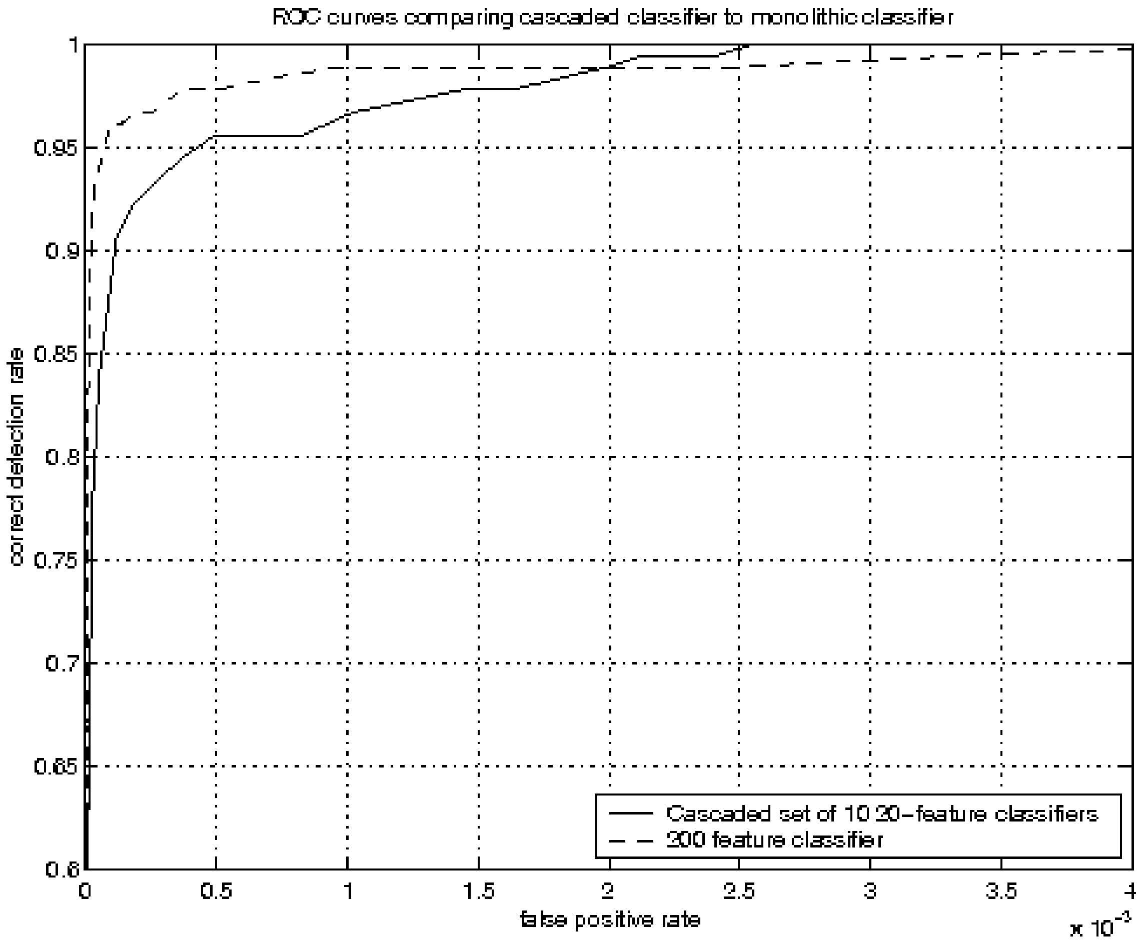
2. Use validation set to find strong classifier threshold that meets detection rate and false positive rate criteria. If not possible, go back to step 1 to add N more features.

3. Use current cascade to find false positives in large training set of negative examples.

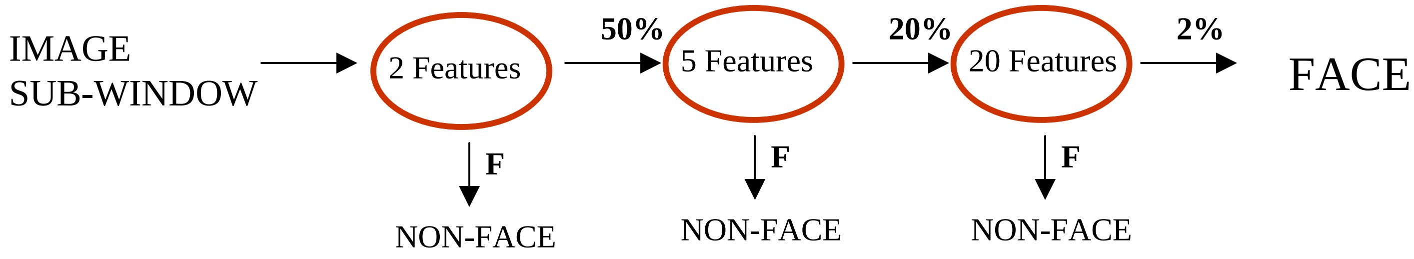
4. Create new set of negative examples from the false positives. Positive examples can stay the same as before.

5. Go to step 1 to train a new classifier in the cascade.

Experiment: Simple Cascaded Classifier



Cascaded Classifier



- A 2 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

A Real-time Face Detection System

Training faces: 4916 face images (24 x 24 pixels) plus vertical flips for a total of 9832 faces

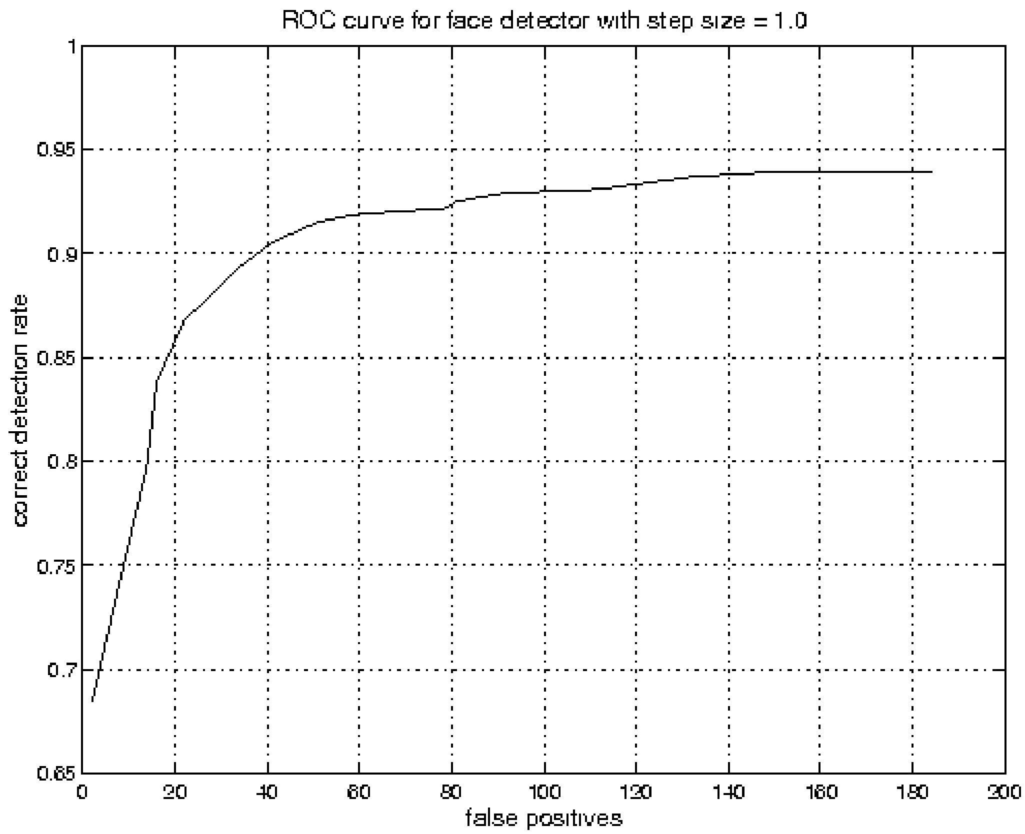
Training non-faces: 350 million sub-windows from 9500 non-face images

Final detector: 38 layer cascaded classifier
The number of features per layer was 1, 10, 25, 25, 50, 50, 50, 75, 100, …, 200, …

Final classifier contains 6061 features.

Accuracy of Face Detector

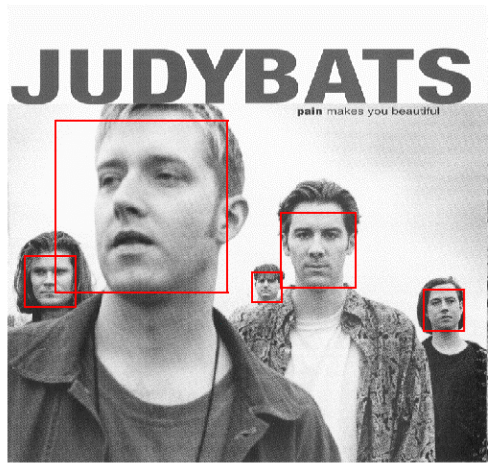
Performance on MIT+CMU test set containing 130 images with 507 faces and about 75 million sub-windows.

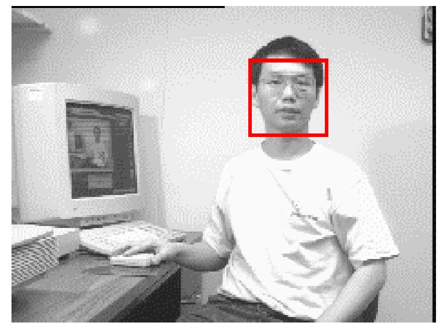


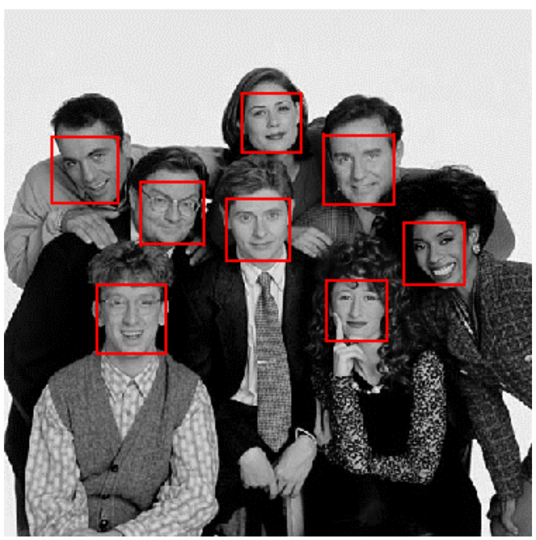
Comparison to Other Systems

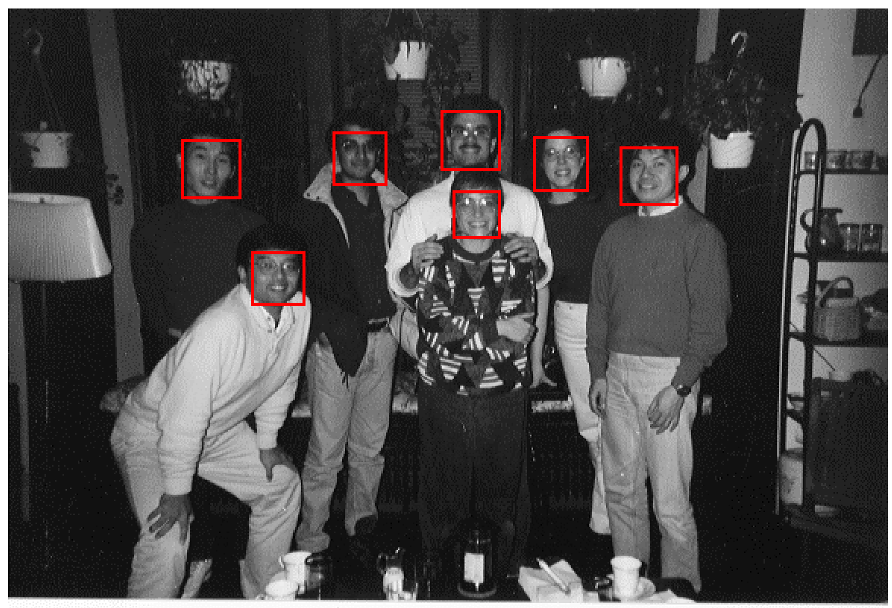
False Detections / Detector	10	31	50	65	78	95	110	167
Viola-Jones	76.1	88.4	91.4	92.0	92.1	92.9	93.1	93.9
Viola-Jones (voting)	81.1	89.7	92.1	93.1	93.1	93.2	93.7	93.7
Rowley-Baluja-Kanade	83.2	86.0				89.2		90.1
Schneiderman-Kanade				94.4				

Output of Face Detector on Test Images





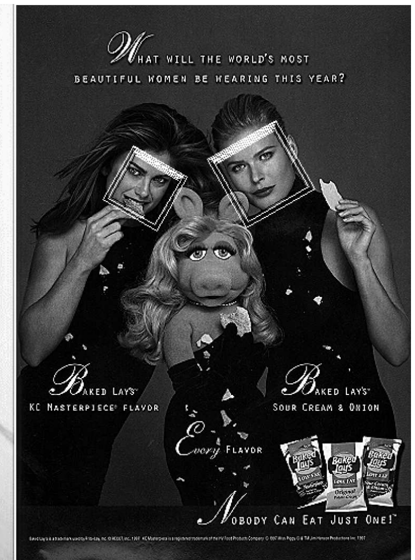


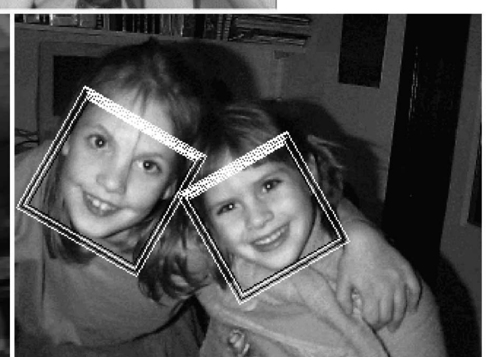


Live Demo

Profile Detection

Rotated Face Detection



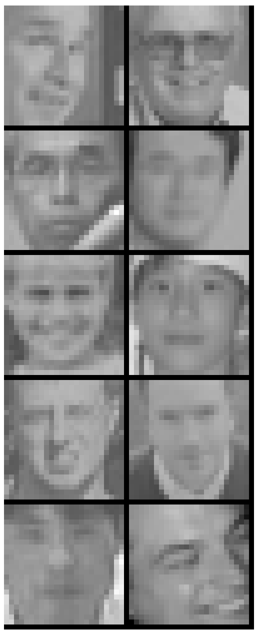


Facial Feature Detection

Gender Classification

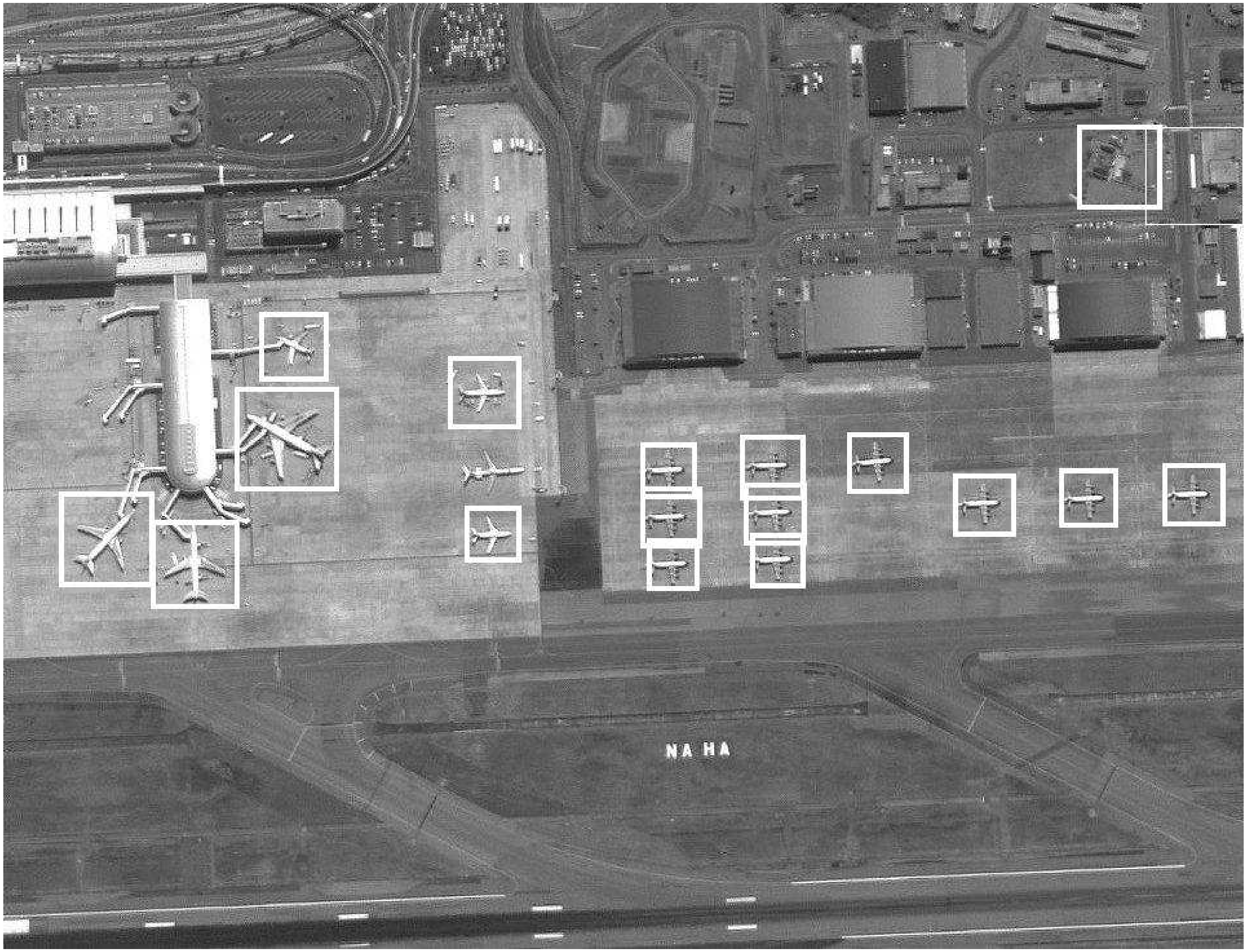
- Gender classifier is trained from face images divided into male and female classes
- Achieves about 80% accuracy

Males



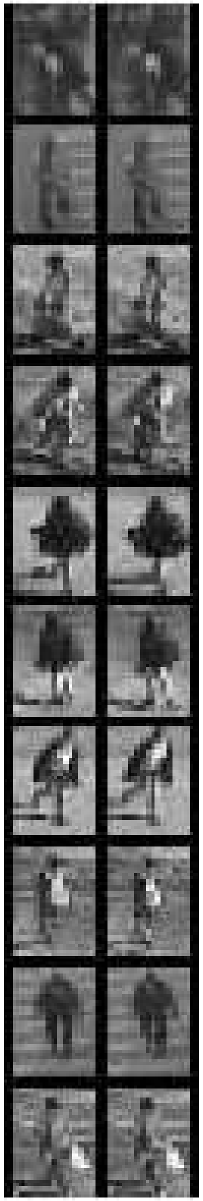
Females

Airplane Detection



16 out of 17 detections, 2 false alarms

Pedestrian Detection



Problems
- Low Resolution ~ 20 pixels high
- Varying Lighting / Clothing / Pose

Input Representation

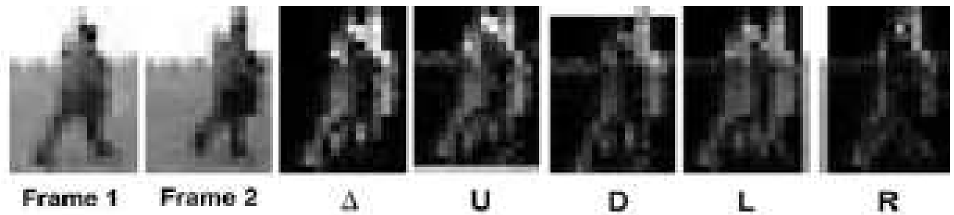


Image Diff
Up Shifted
Down Shifted
Left Shifted
Right Shifted

$$\Delta = abs(I_2 - I_1)$$

$$U = abs\left(I_2 - \left(\uparrow I_1\right)\right)$$

$$D = abs\left(I_2 - \left(\downarrow I_1\right)\right)$$

$$L = abs\left(I_2 - \left(\leftarrow I_1\right)\right)$$

$$R = abs\left(I_2 - \left(\rightarrow I_1\right)\right)$$

Pedestrian Detection Features

Intensity

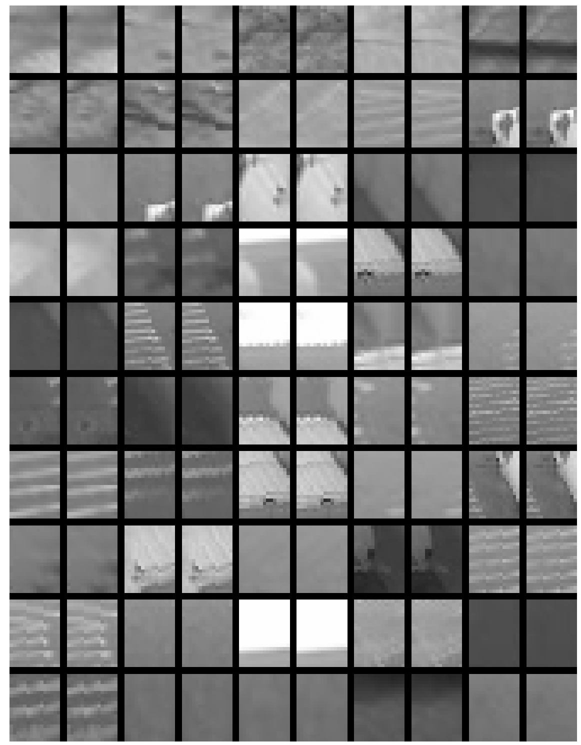
Motion-Direction

Motion-shear

Abs-Difference

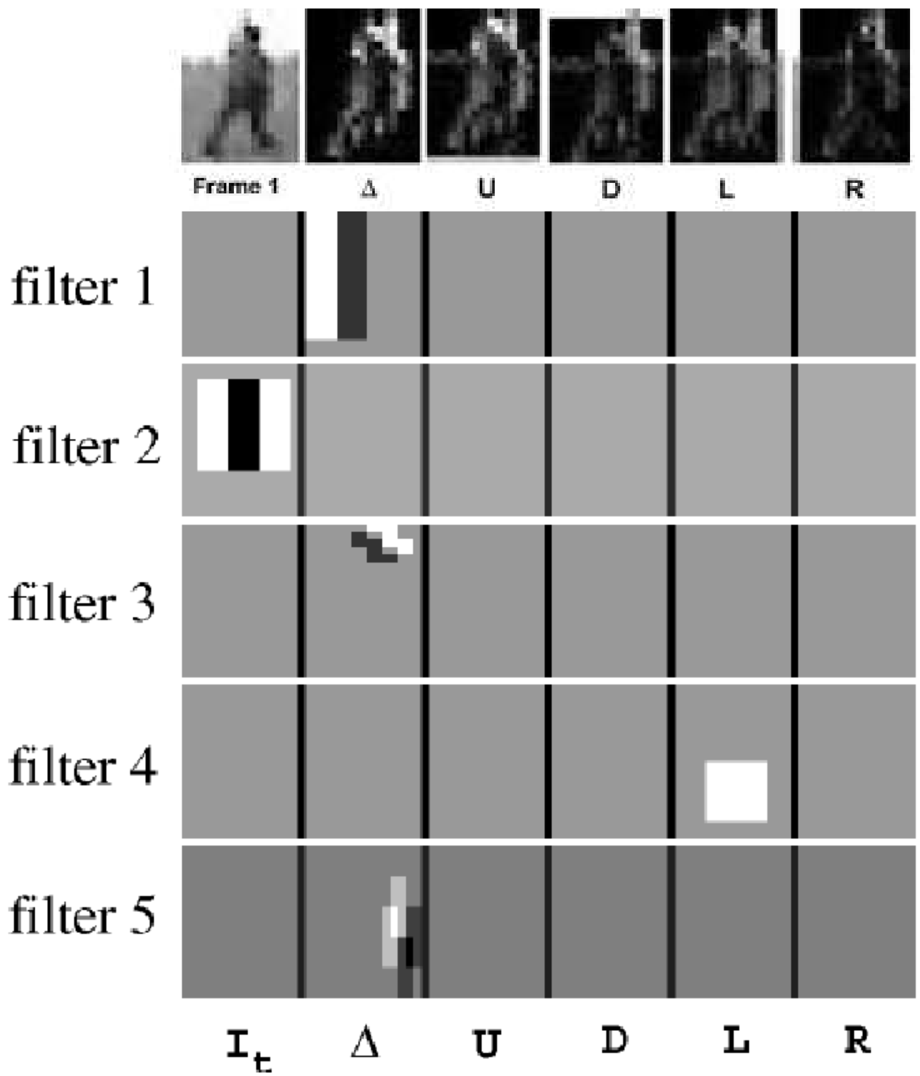
Some example patterns used for training

Positive example pairs



Negative example pairs

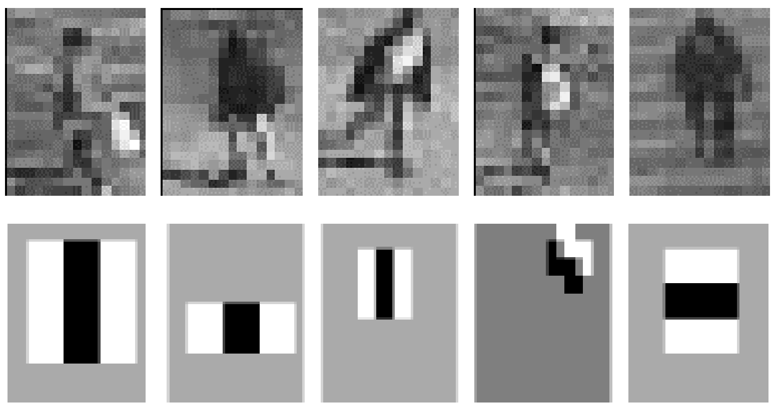
Motion and Appearance Filters Learned



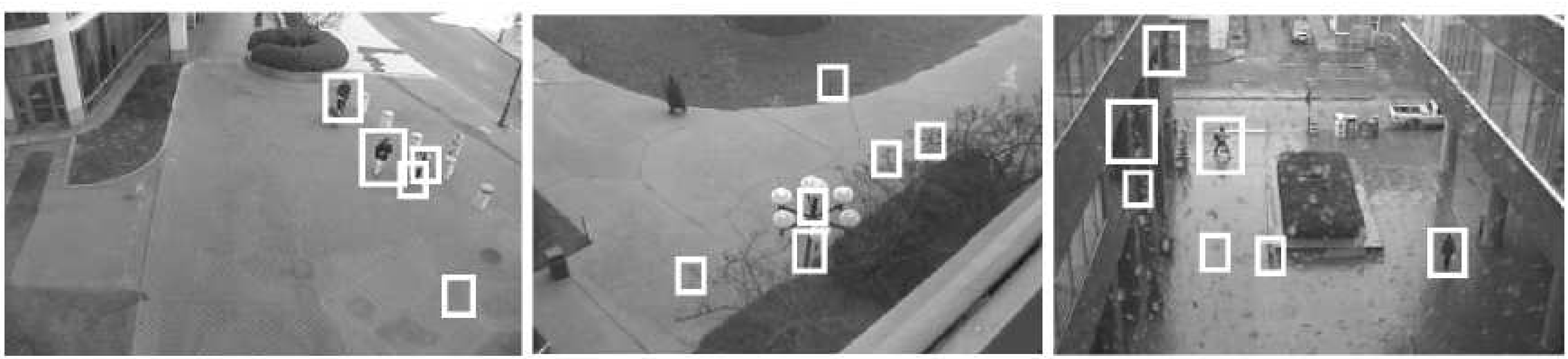
Detection rate: 90%
False pos rate: 1 in 100,000

Experiments

Constructed a baseline Viola-Jones static detector



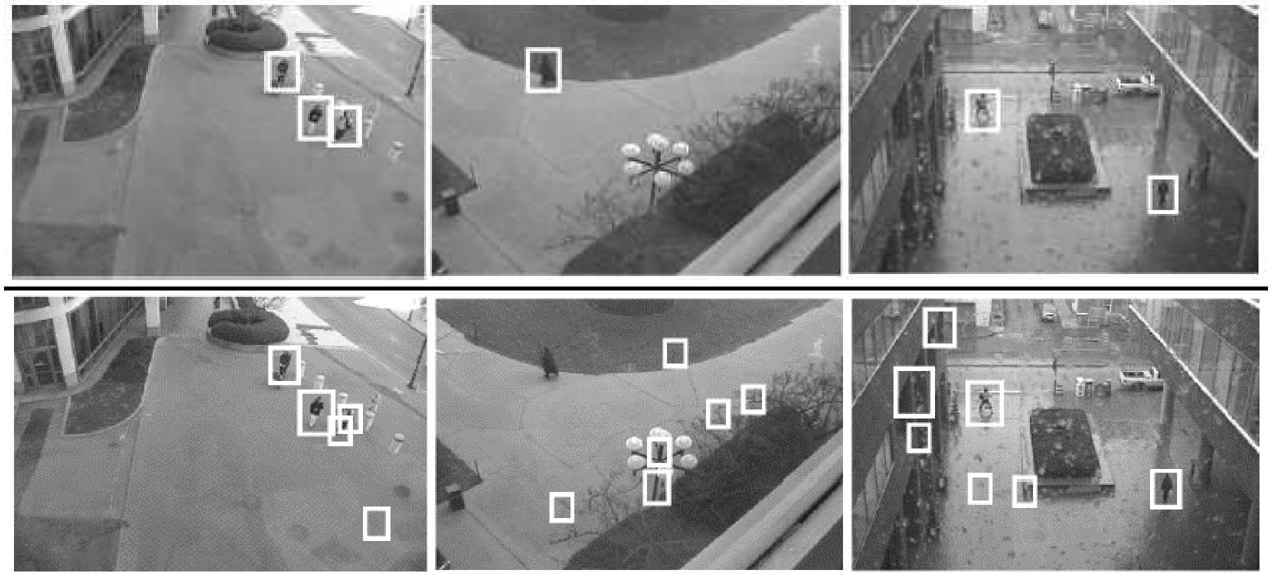
Detection rate: 80%
False pos rate: 1 in 100,000

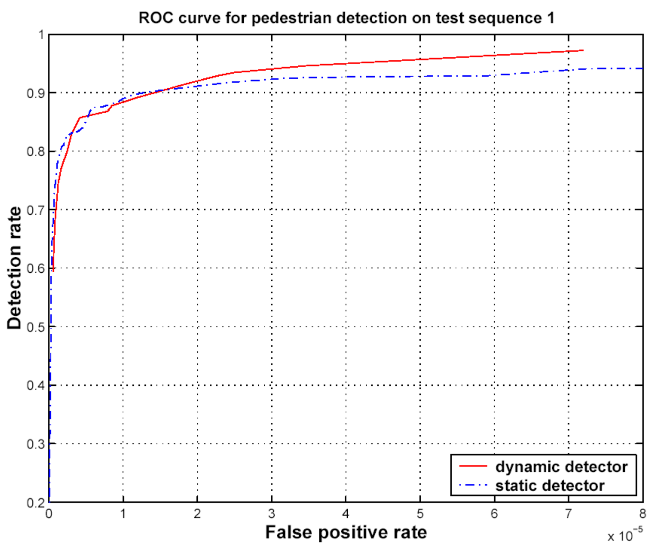


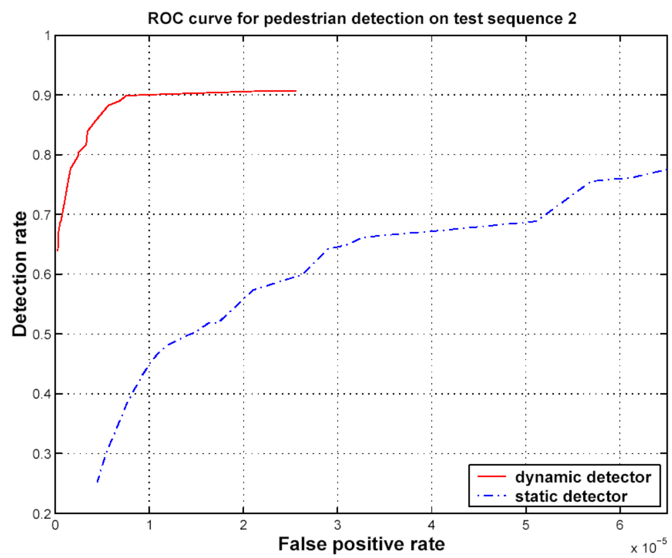
Before and After

Dynamic detector

Static detector







Results Videos

The Face Recognition Problem

System is given a *gallery* of known faces:

. . .

Input is a *probe* face:

Problem is to determine if the probe face matches any of the gallery faces

Face recognition as binary classification

Build a face verifier, F( )

Input: two face images

Output: {same, different}

F

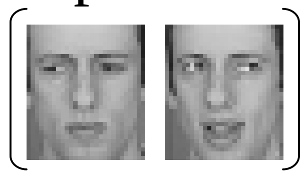
$\longrightarrow$ {same, different}

Compare probe image to every gallery image.

Learn Verifier from Examples

Training set consists of “same” pairs:

→ same



→ same

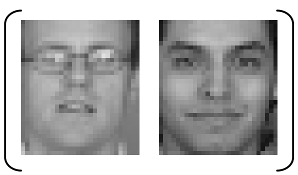
→ same

→ same

And “different” pairs:

→ different

→ different



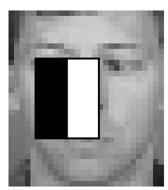
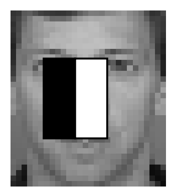
→ different

→ different

Details of Face Verifier

Our verifier is a linear combination of simple features.

A feature is evaluated on each image in the input pair, the results are subtracted, the absolute value is taken and then thresholded.

$f_i = \Big\{$ α if $\Big|$  $-$  $\Big| > T$

β otherwise

$$F = \theta\left(\sum_i f_i + b\right)$$