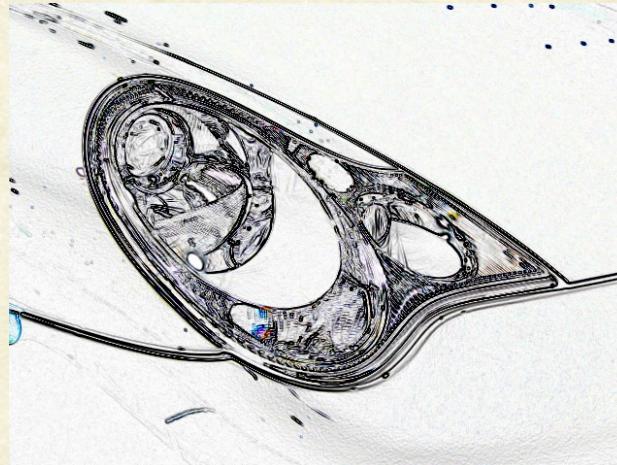




CS7.505: Computer Vision

Spring 2022: Face Detection



Anoop M. Namboodiri
Biometrics and Secure ID Lab, CVIT,
IIIT Hyderabad

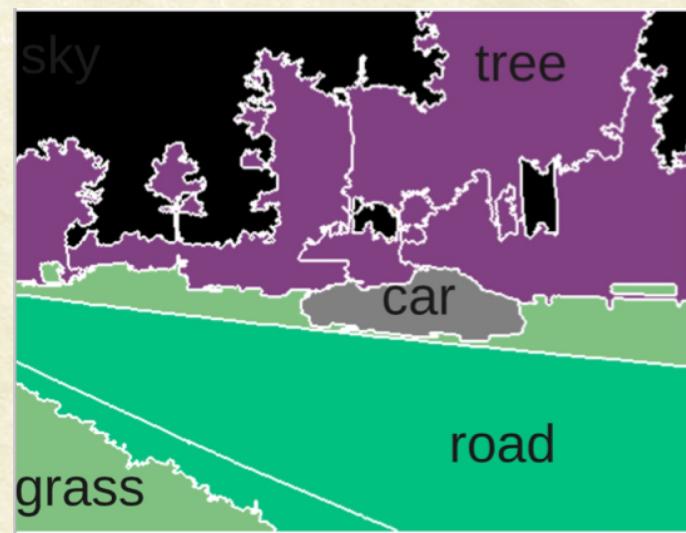
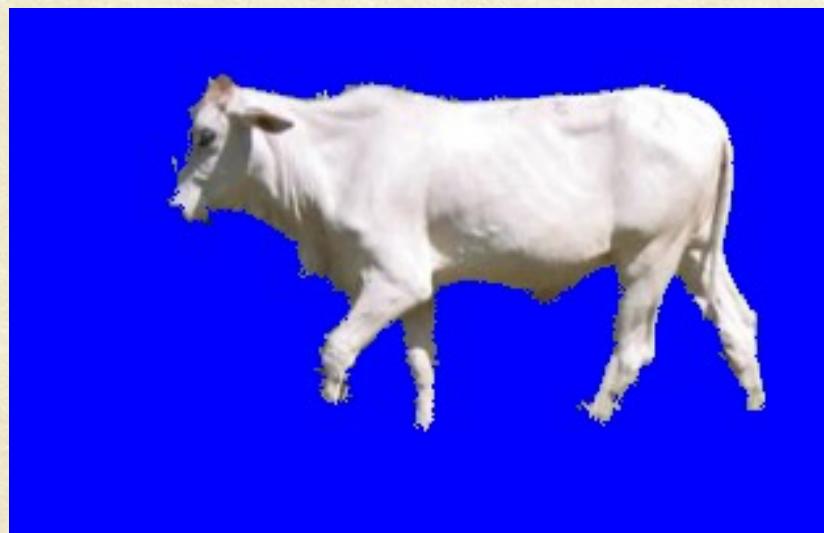


Labeling Problems in Vision

- Labeling Pixels
 - Segmentation (Semantic)
 - Stereo
 - Normal Estimation
 - Restoration/Enhancement
- Labeling Windows
 - Object Detection
 - Object Recognition
- Other Problems
 - Matching: Registration, Tracking
- Posed as Optimization or Energy Minimization
 - MLE/MAP
 - MRF
 - DTW
 - Convex
 - Linear Programming
 - MCMC
 - Simulated Annealing



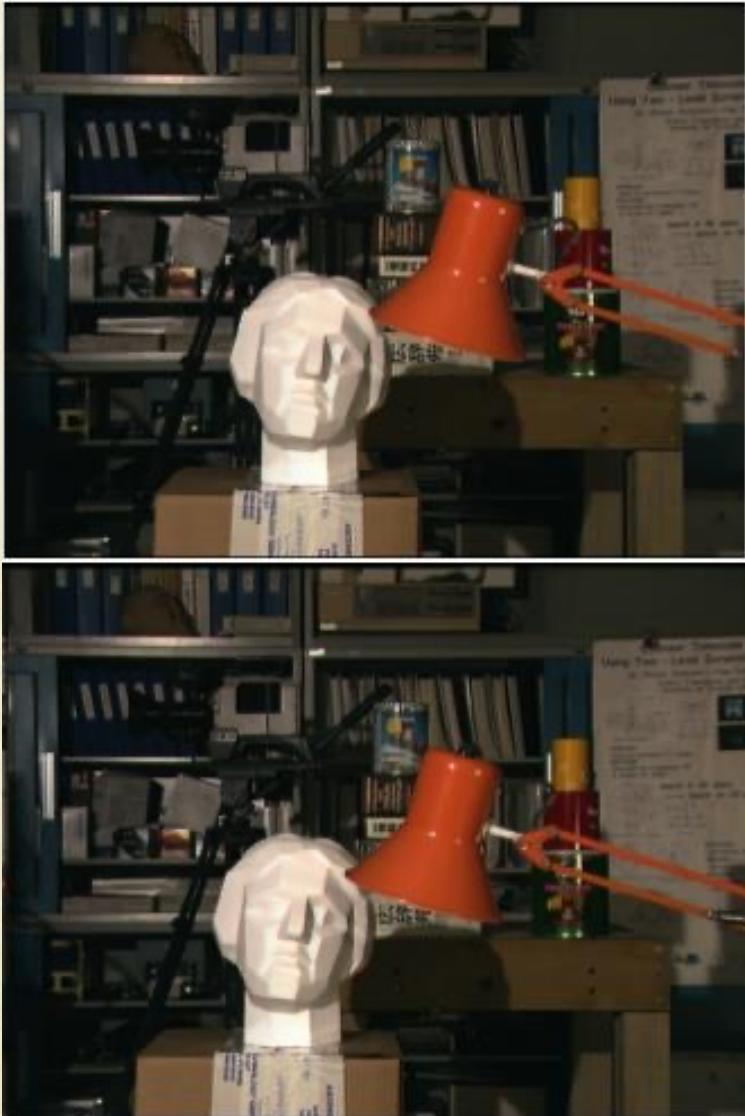
Segmentation as Labeling





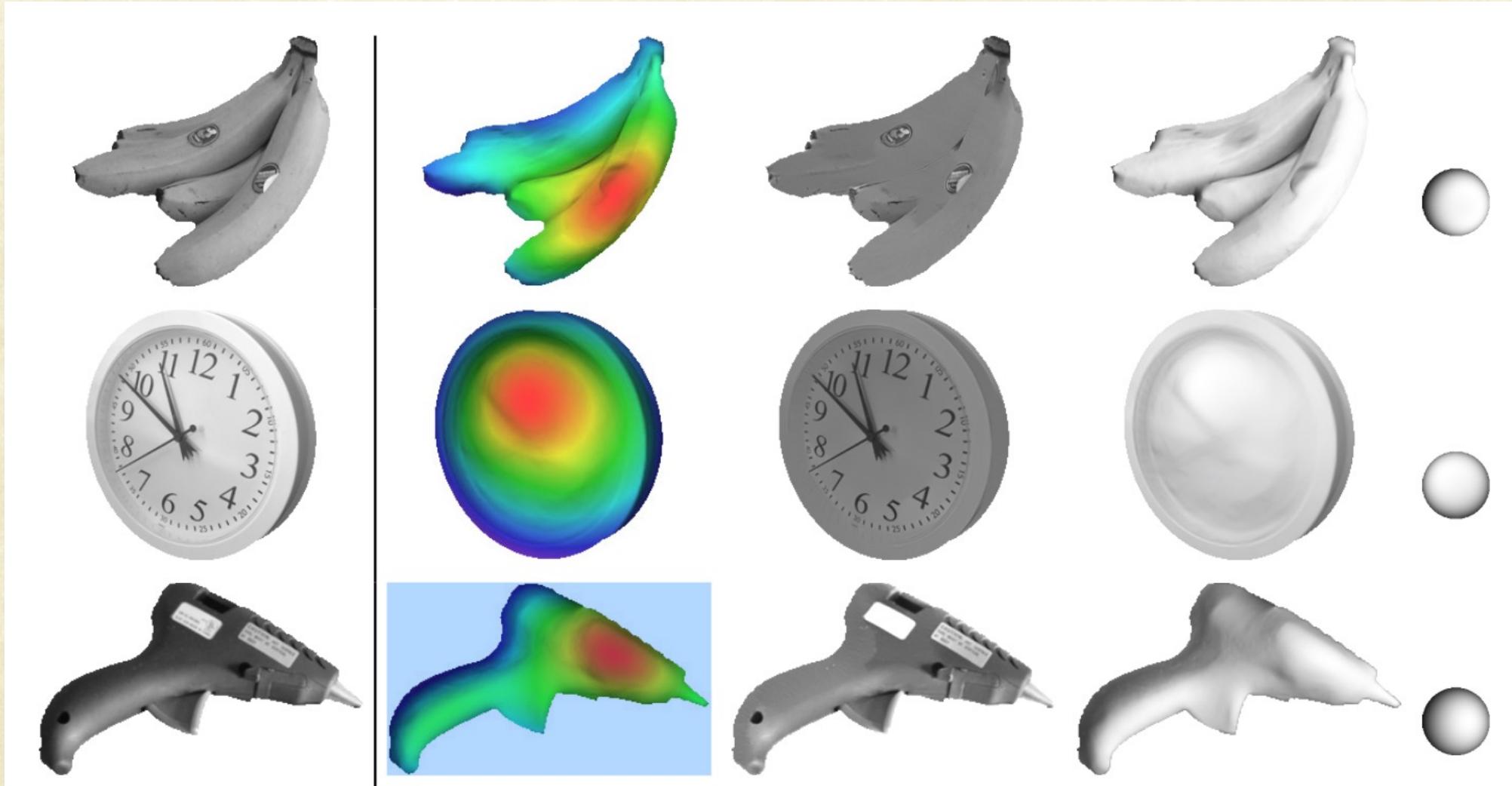
Stereo as Labeling

Each pixel is assigned a disparity (depth). DTW or MRF for optimization





Normal/Shape Estimation



Estimate normal and albedo at every pixel: What should we optimize?

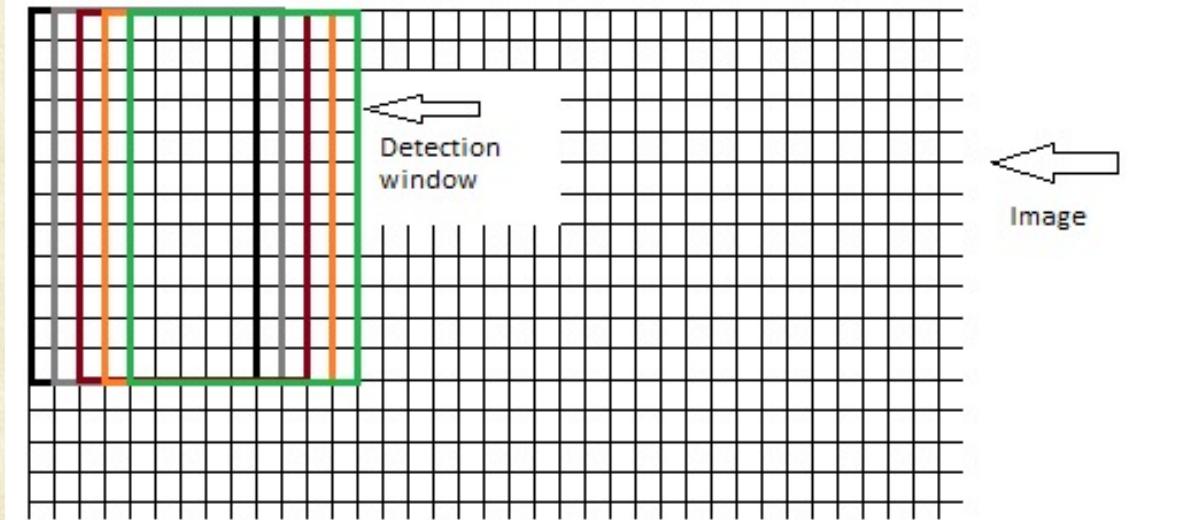


Image Restoration/Enhancement

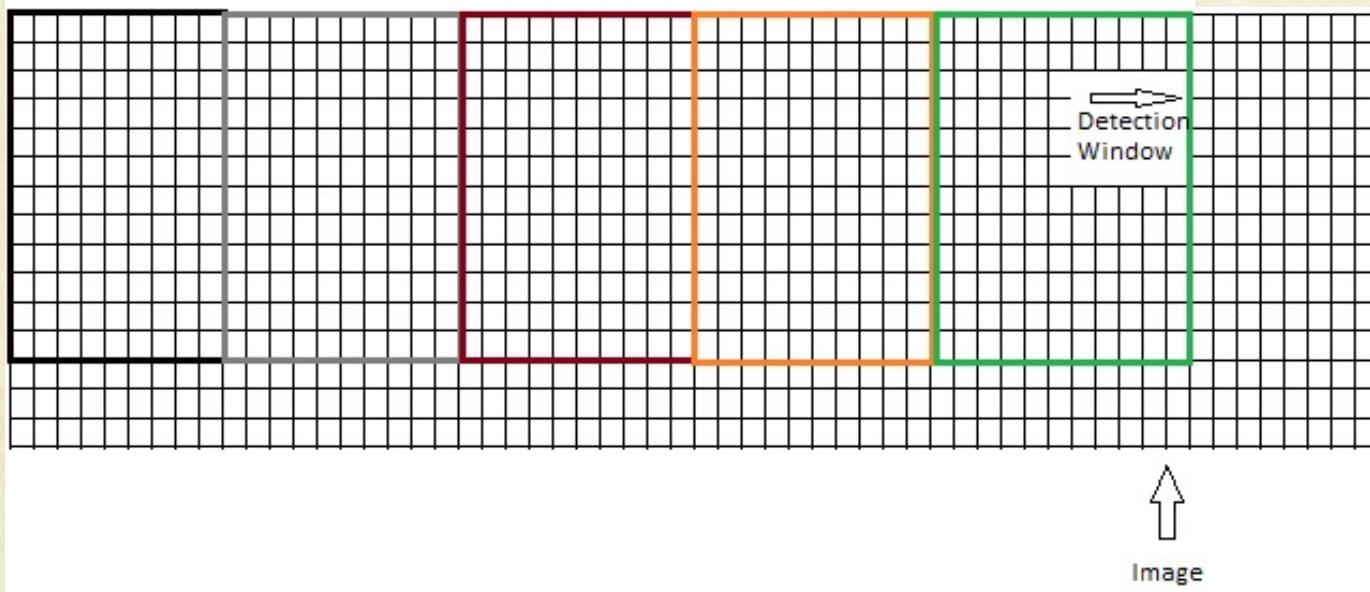




Detection and Recognition



Classify each
window



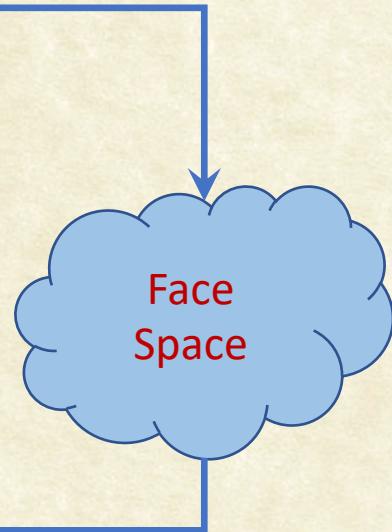


Face Detection using Eigenfaces

Φ



$\hat{\Phi}$



- Given an unknown image Γ

Step 1: compute $\Phi = \Gamma - \Psi$

Step 2: compute $\hat{\Phi} = \sum_{i=1}^K w_i u_i$ ($w_i = u_i^T \Phi$)
(where $\|u_i\|=1$)

Step 3: compute $e_d = \|\Phi - \hat{\Phi}\|$

Step 4: if $e_d < T_d$, then Γ is a face.

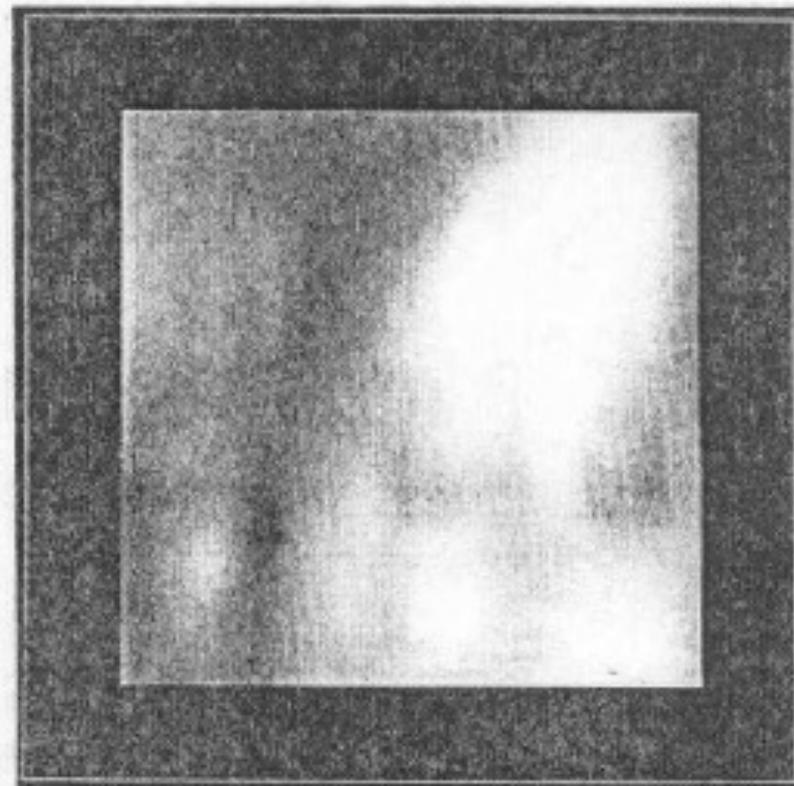
$e_d = \|\Phi - \hat{\Phi}\|$ is called **distance from face space (dffs)**



Distance from Face Space

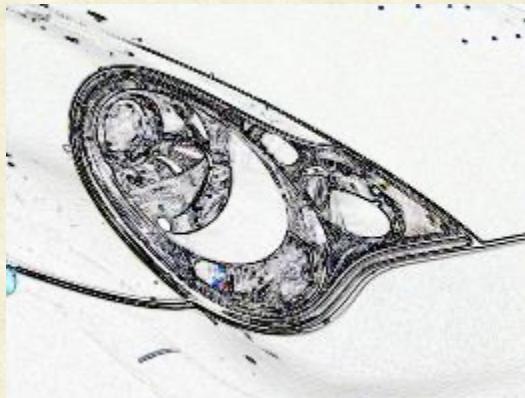
Visualize dffs:

$$e_d = \|\Phi - \hat{\Phi}\|$$



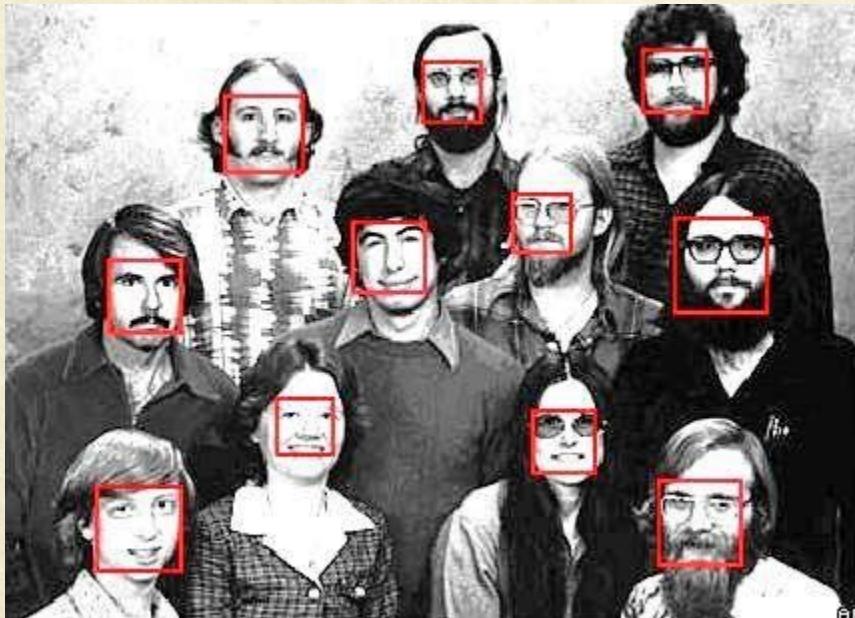


Real-time Face Detection: Viola and Jones





The Task of Face Detection





Basic Idea

Slide a window across image and evaluate a face model at every location





How Many Windows (Speed)

- 1280×1024 image;
24×24 to
1024×1024
windows
- # of Windows?
 - 1.3 million locations
 - 18 scales/window sizes at 1.25 multiples
 - 14.5 million potential face candidates
 - Features to be extracted for each candidate window





Accuracy (FPR)

- Classify each as face or non-face
 - What should be the accuracy (False Positive Rate)?
 - Faces are rare: 0-10 per image
 - For computational efficiency, we should try to **spend as little time as possible on the non-face windows**
 - To avoid having a false positive in every image, the **false positive rate has to be less than 10^{-6}**

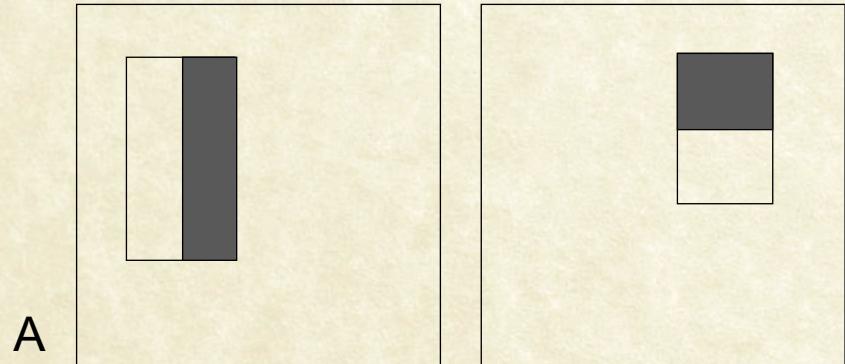


The Viola/Jones Face Detector

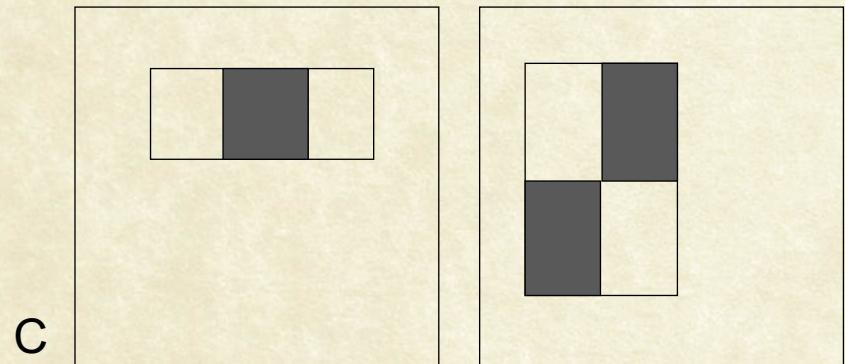
- A seminal approach to **real-time object detection**
- Key ideas
 - *Integral images* for **fast feature evaluation**
 - *Boosting* for **feature selection**
 - *Attentional cascade* for **fast rejection of non-face windows**
- P. Viola and M. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features*. CVPR 2001.
- P. Viola and M. Jones. *Robust Real-Time Face Detection*. IJCV 57(2), 2004.



Image Features

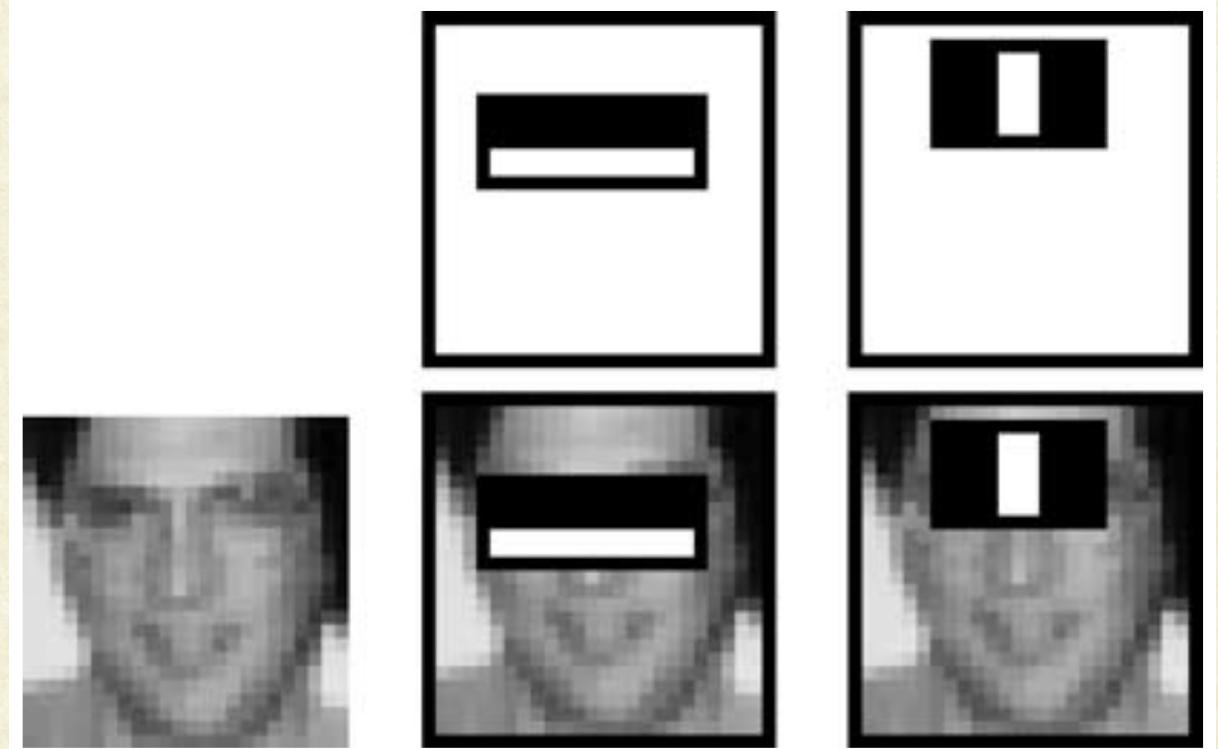


B



D

Rectangular filters



$$f(r, u) = \sum_{i \in r} p_r(i) - \sum_{i \in u} p_w(i)$$

Local feature
2-rectangle

Too many features

In a 24×24 patch with 4×4 detector, there are over 160,000 locations for rectangles

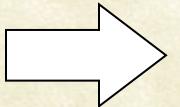
in black area;
feature (D)



Integral Image

IMAGE

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0



INTEGRAL IMAGE

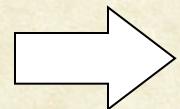
0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21



Computing Integral Image

IMAGE

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0



INTEGRAL IMAGE

0	1	2	3
1	4	7	11
2	7	?	

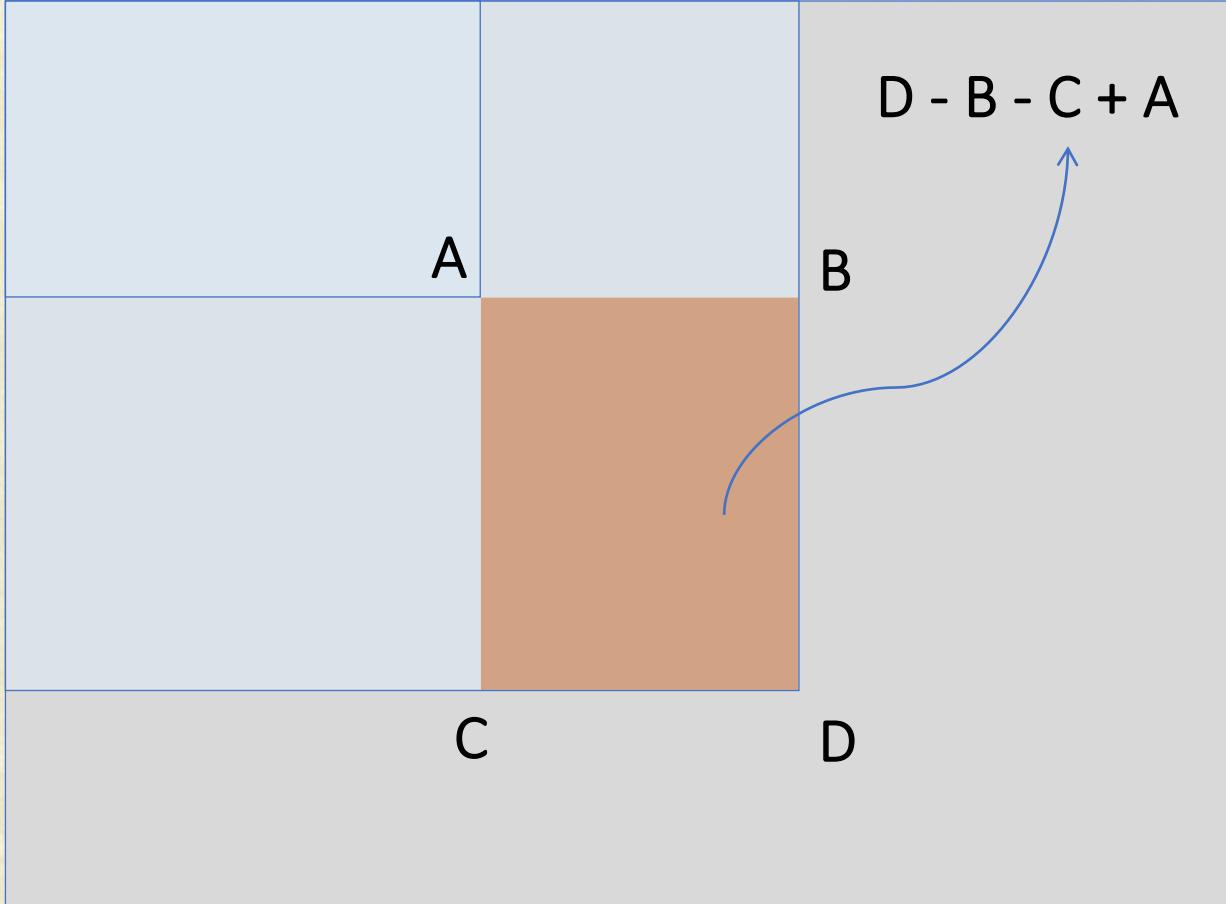
$s(x,y)$ is the
cumulative row
sum

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

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



Computing Rectangular Features



- Rectangle features are somewhat primitive and coarse, they are sensitive to presence of edges, bars, and other simple structure
- Generating a large no. of these features and their computational efficiency compensates for this.



Face Detector

- Scan the input at many scales
- Starting at the base scale in which faces are detected at 24x24 pixels, a 384x288 pixel image is scanned at 12 scales each a factor 1.25 larger than the last
- Any rectangle feature can be evaluated at any scale and location in a few operations
- Face detection @15 fps for the entire image



Weak Learners for Face Detection

- Hypothesis: A very small no. of 160K ($\times 4$) features for each image sub-window can be formed to find an effective classifier (face vs. non-face)
- AdaBoost is used both to select the features and train the classifier
- Weak learner: a single rectangle feature that best separates positive and negative examples; so weak classifier is a thresholded single feature (can be viewed as a single node decision tree)

$$h_t(x) = \begin{cases} 1 & \text{if } p_t f_t(x) > p_t \theta_t \\ 0 & \text{otherwise} \end{cases}$$

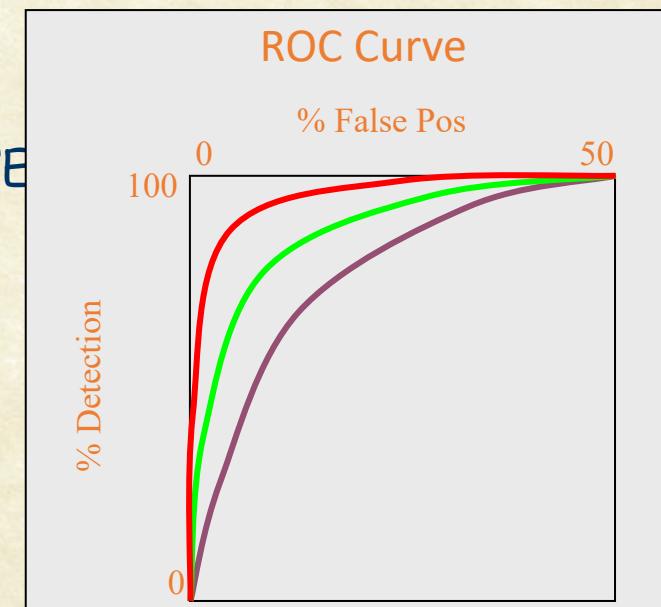
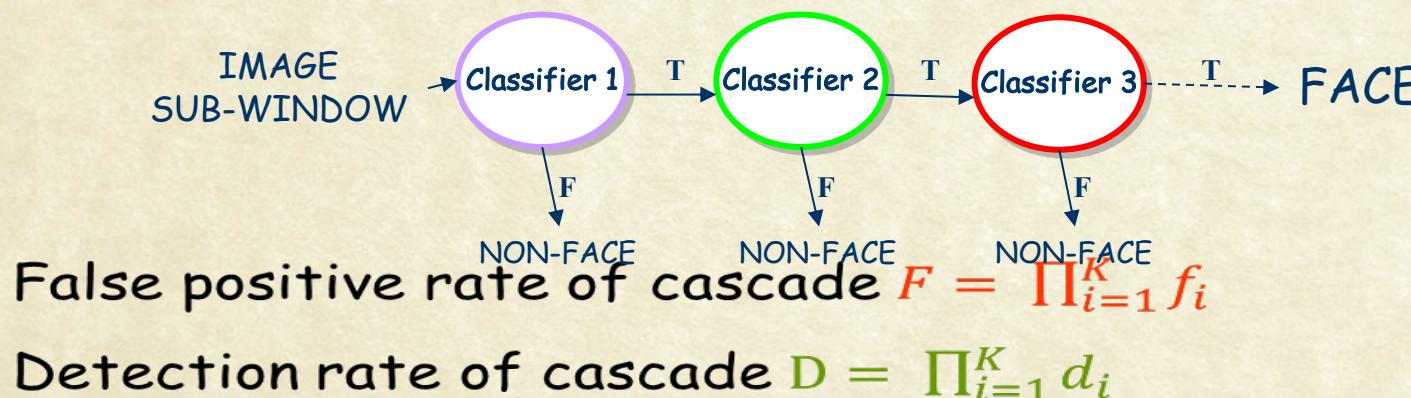
Diagram illustrating the components of the weak learner:

- window**: Points to the input window x .
- value of rectangle feature**: Points to the value of the rectangle feature $p_t f_t(x)$.
- parity**: Points to the condition p_t .
- threshold**: Points to the threshold θ_t .



Cascade of classifiers

- Train each classifier with **increasing** number of features until the **target** false positive and detection rates are achieved
- Use false positives from current stage as the **negative training examples** for the next stage
- Classifiers are progressively more complex and have lower false positive rates



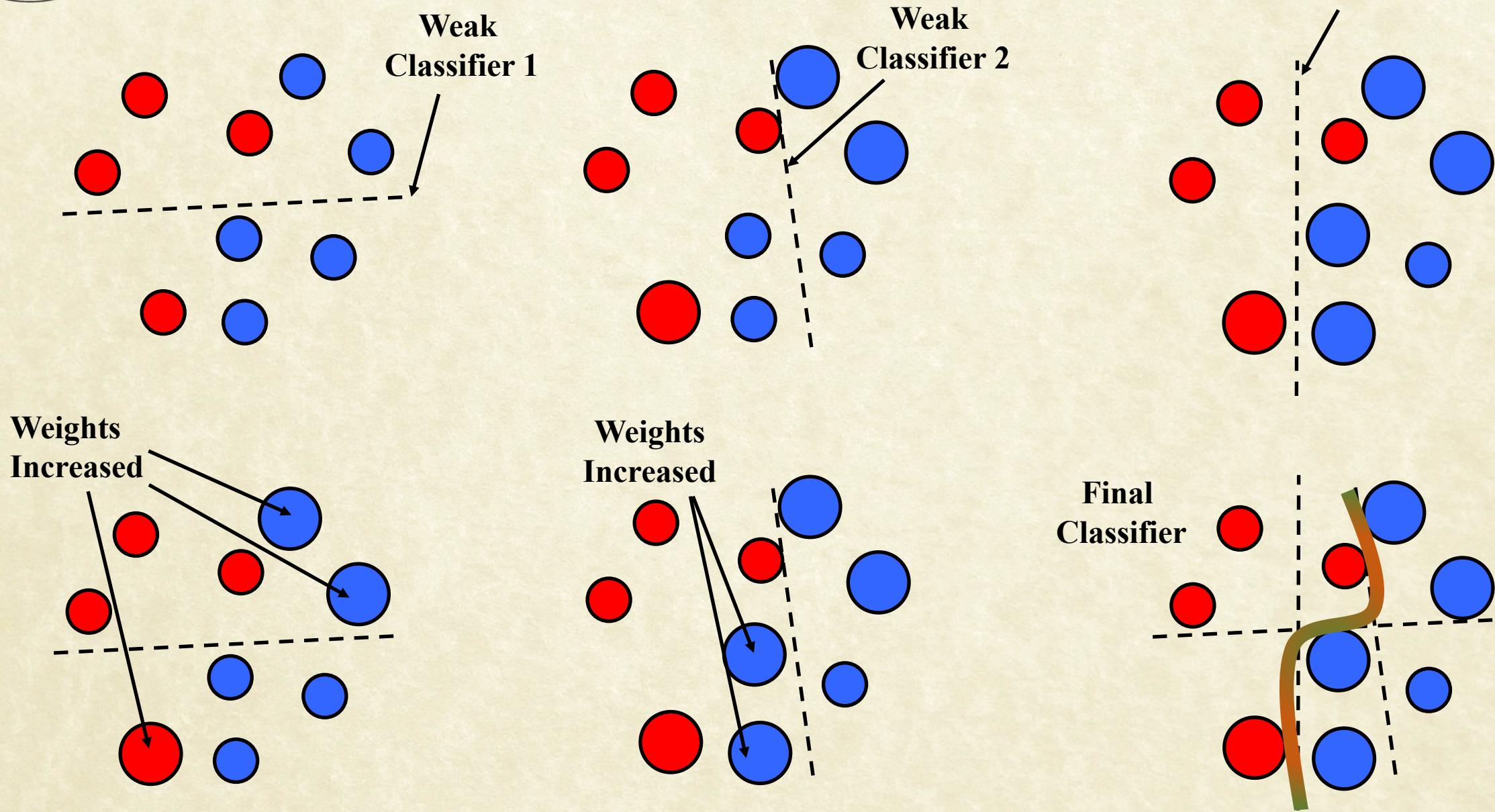


Boosting

- Training set contains **face** and **nonface** examples
 - Initially, with equal weight
- For each round of boosting:
 - Evaluate each **rectangle filter** on each example
 - Select best **threshold** for each filter
 - Select best **filter/threshold combination**
 - **Reweight** examples
- Computational complexity of learning: $O(MNK)$
 - M rounds, N examples, K features



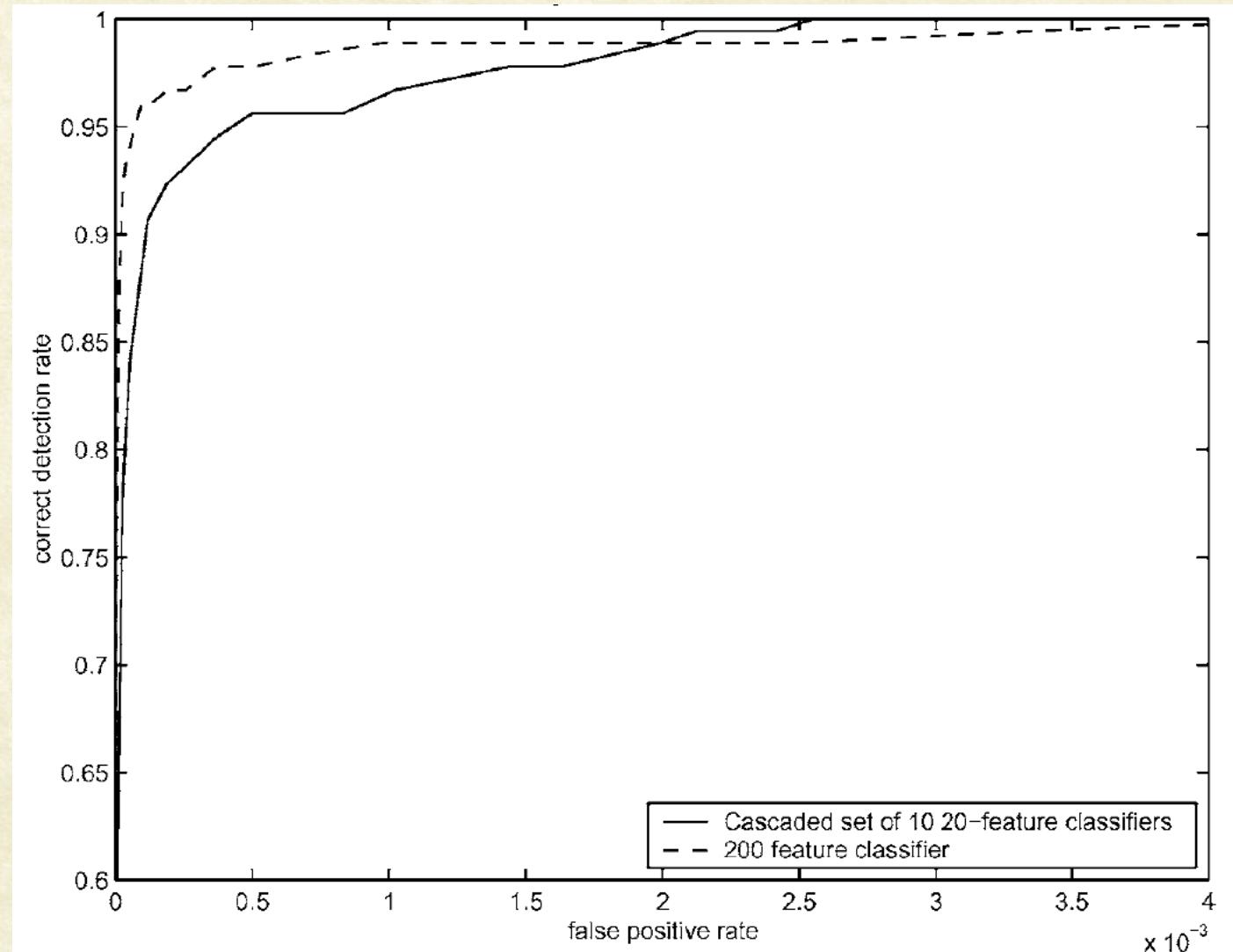
Boosting Illustration





ROC Curves Cascaded Classifier to Monolithic Classifier

- Little difference between the two in accuracy.
- Big difference in terms of speed: ~10 times faster
 - The first stage throws out most non-faces; never evaluated by subsequent stages.





Face Detection System

- Training Data
 - 5000 faces: all frontal, rescaled to 24x24 pixels
 - 9500 million non-faces
- Many variations
 - Across individuals
 - Illumination
 - Pose

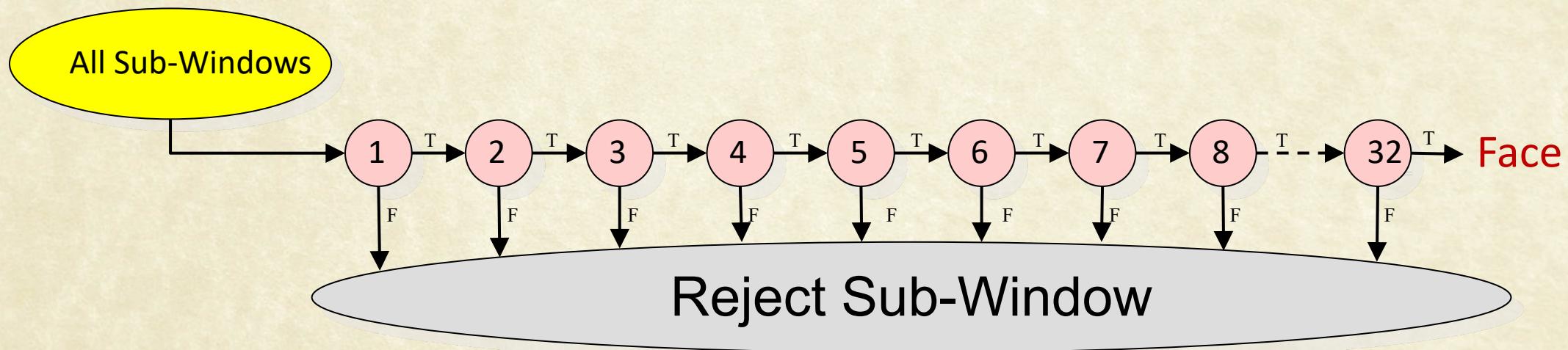




Structure of the Detector Cascade

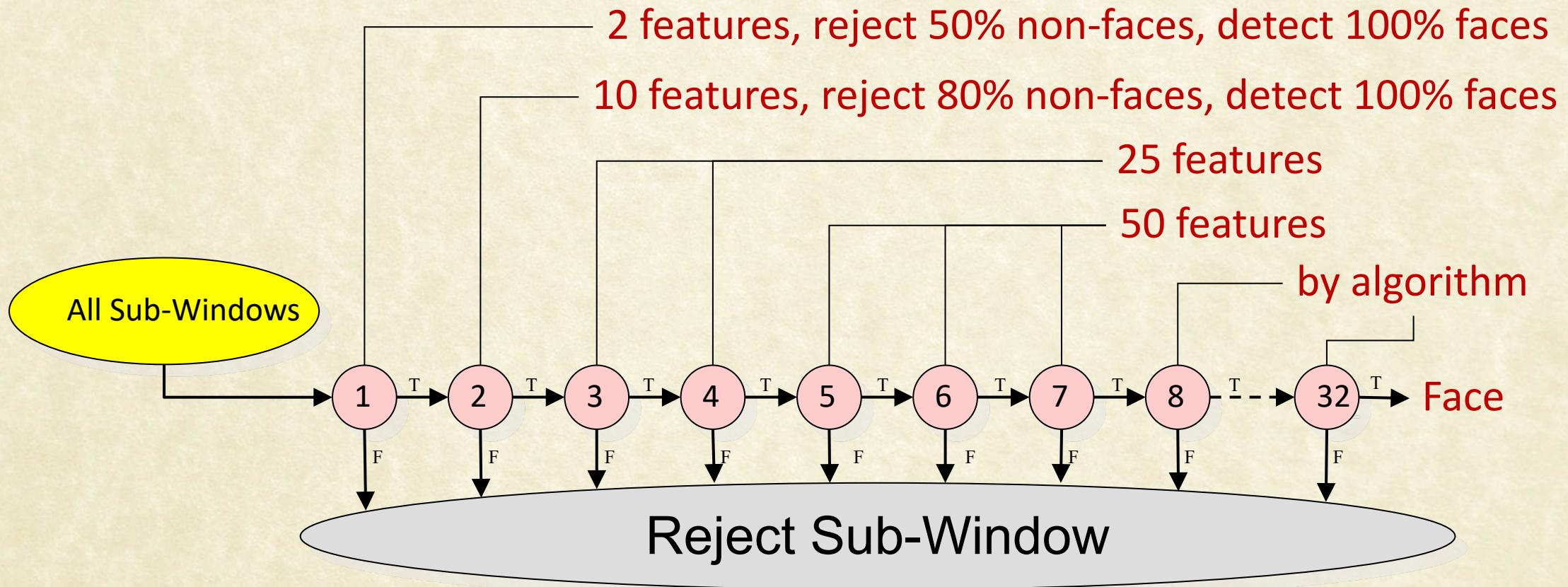
Increasingly complex classifiers in cascade

- 32 stages
- included a total of 4297 features





Structure of the Detector Cascade





Feature Selection

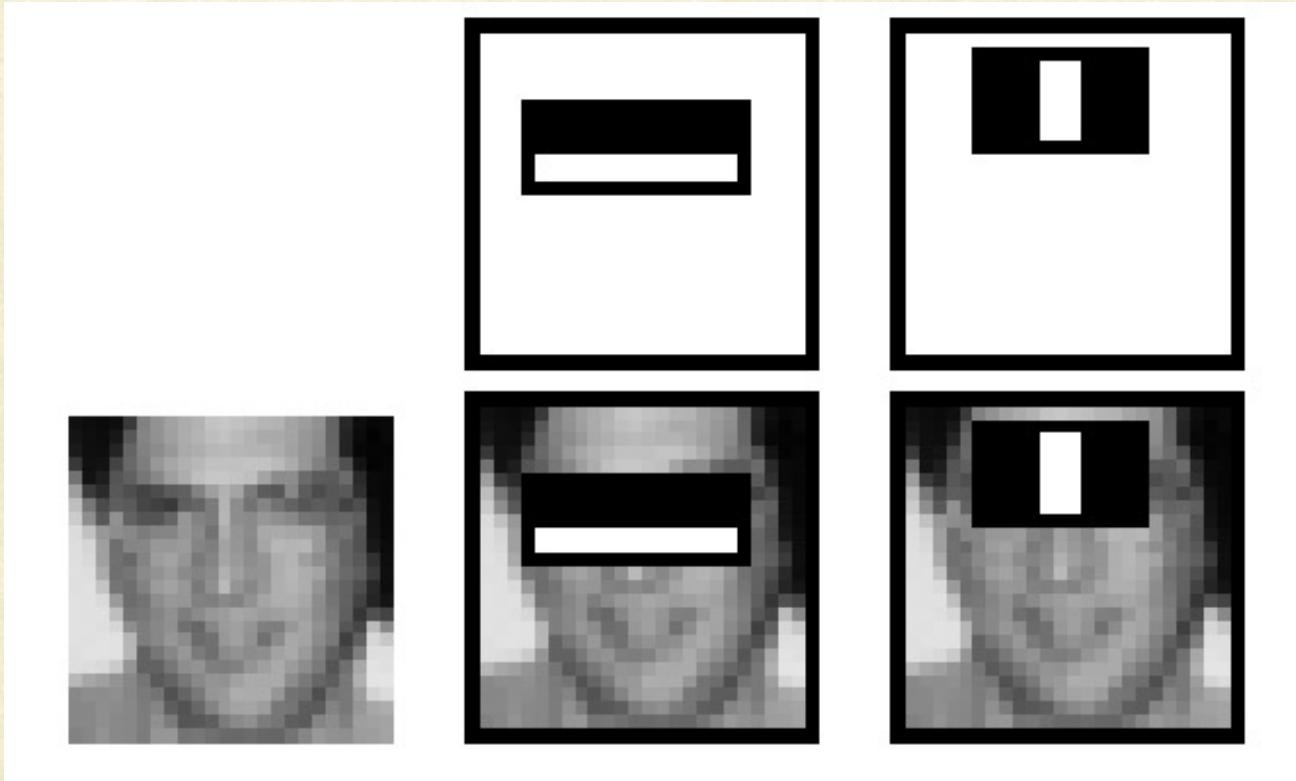


Figure 5: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature 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.

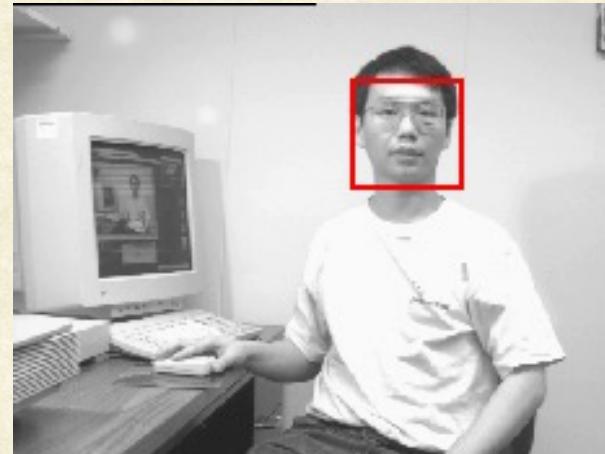
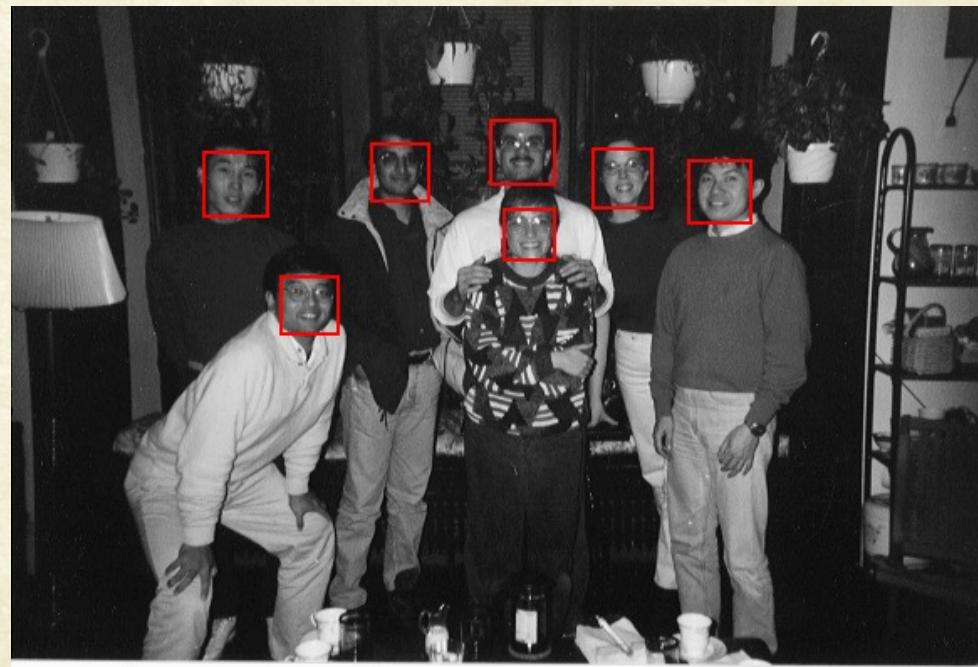
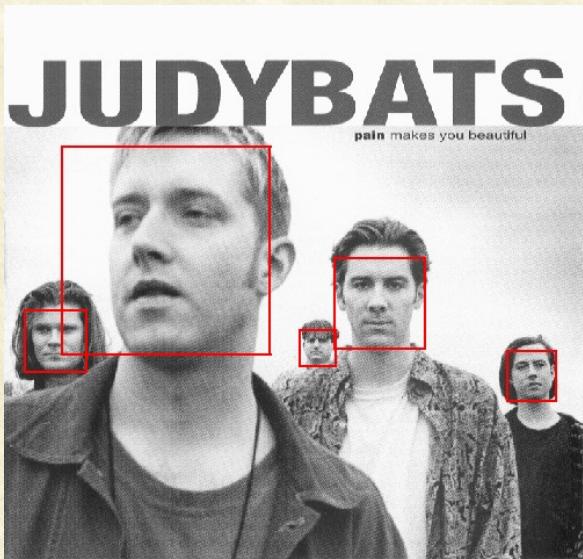


Speed of the Final Detector

- On a 700 Mhz Pentium III processor, the face detector can process a **384×288** pixel image in about “**.067 seconds**”
 - 15 times faster than previous detector of comparable accuracy (Rowley et al., 1998)
- **Average of 8 features** evaluated per window on test set



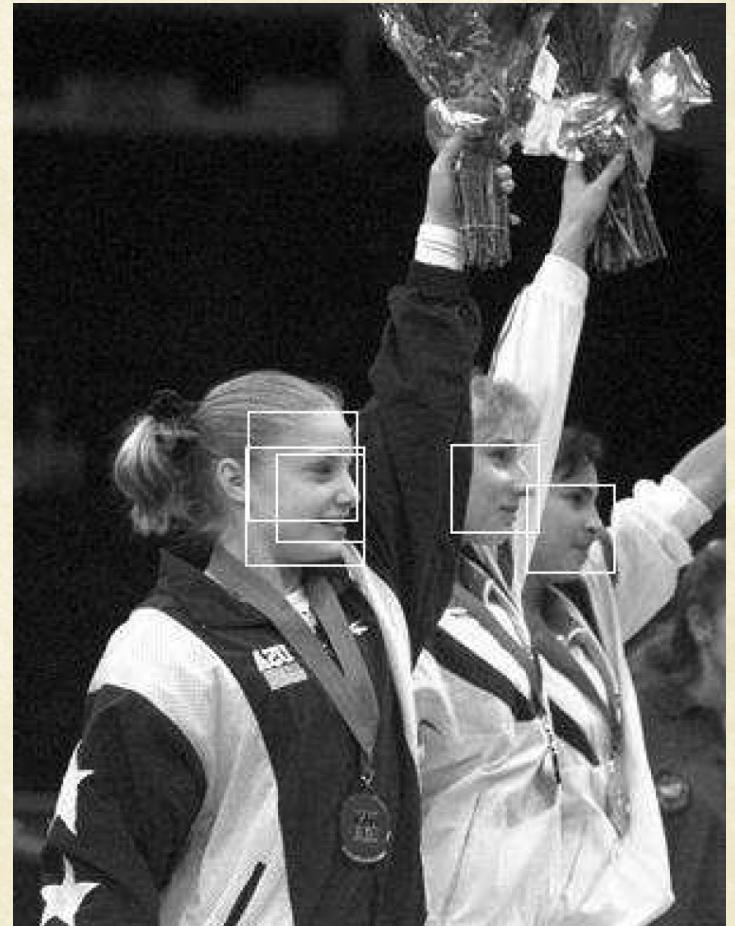
Output of Face Detector on Test Images





Profile Face Detection

- Additional profile faces were used for training



- As the solution is learning based, use of appropriate training data can improve the results in specific use cases.



Thank You