Nov 21st, 10 AM – Midterm exam Saturday

Pedestrian Detection

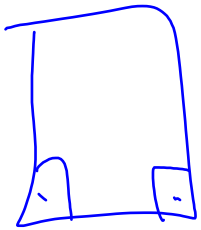
Scanning window

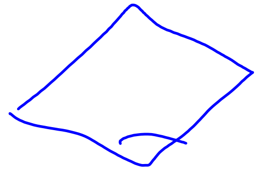
But first...
The Viola/Jones Face Detector
(2001)

- A widely used method for real-time object detection.
- Training is slow, but detection is very fast.
- The same ideas are used for pedestrian detection

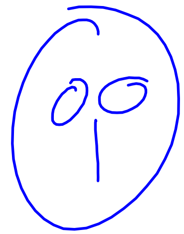
(Most slides from Paul Viola)

Classifier is Learned from Labeled Data





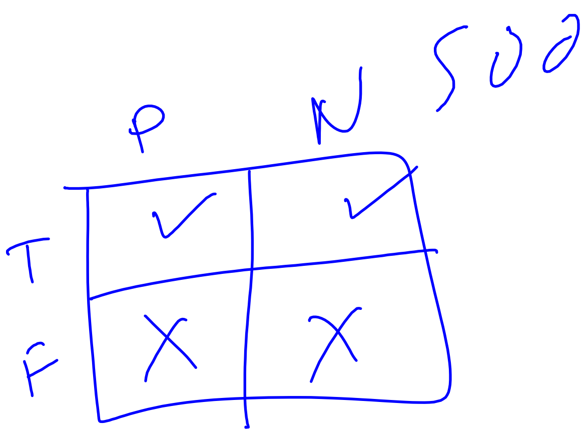
- Training Data
  - 5000 faces
    - All frontal
  - 300 million non faces
    - 9400 non-face images
  - Faces are normalized
    - Scale, translation

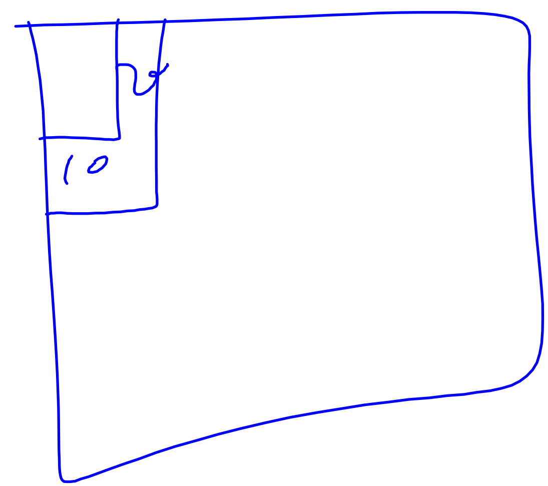


- Many variations
  - Across individuals
  - Illumination
  - Pose (rotation both in plane and out)

Key Properties of Face Detection

- Each image contains 10 - 50 thousand locs/scales
- Faces are rare 0 - 50 per image
    - 1000 times as many non-faces as faces
- Extremely small # of false positives: 10^{-6}

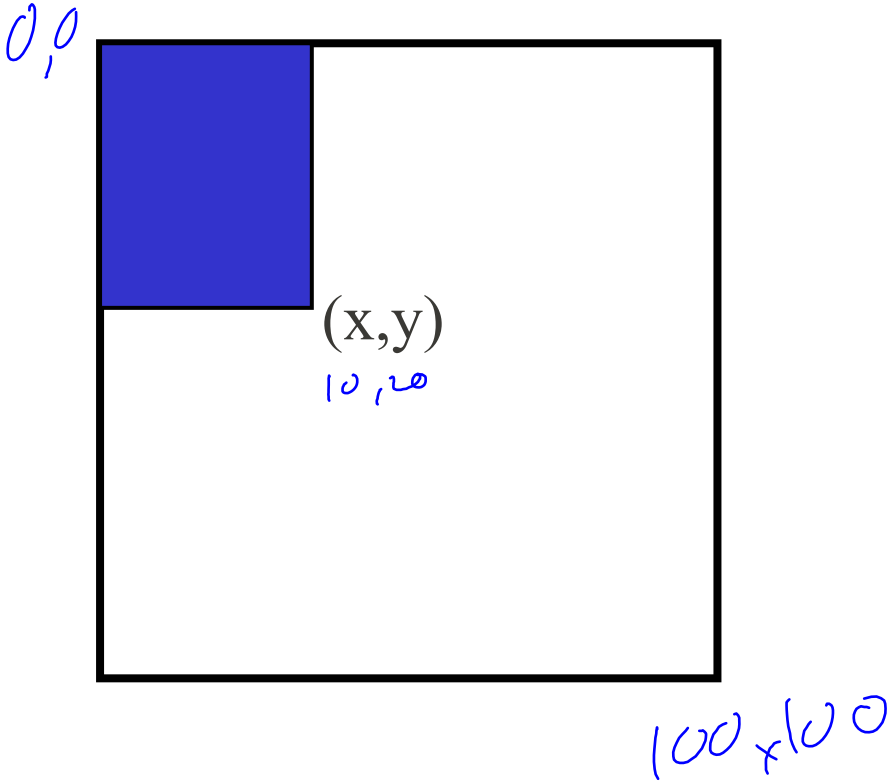




10^6

The Integral Image

- The *integral image* computes a value at each pixel (x,y) that is the sum of the pixel values above and to the left of (x,y), inclusive.
- This can quickly be computed in one pass through the image

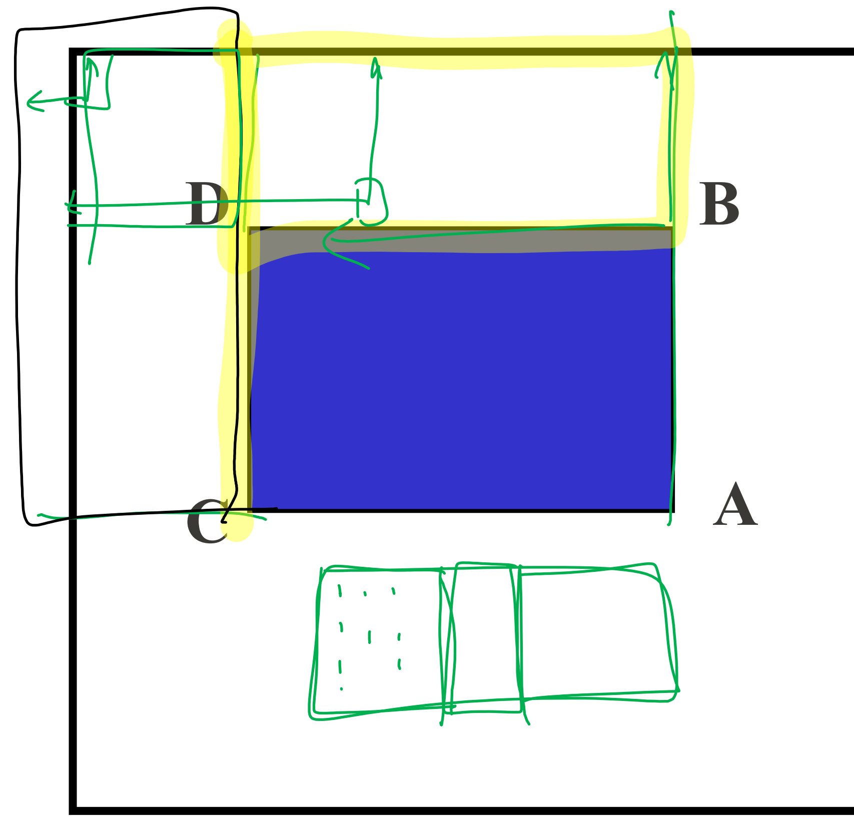


Computing Sum within a Rectangle

- Let A,B,C,D be the values of the integral image at the corners of a rectangle
- Then the sum of original image values within the rectangle can be computed:

  sum = A – B – C + D

- Only 3 additions are required for any size of rectangle!
  - This is now used in many areas of computer vision



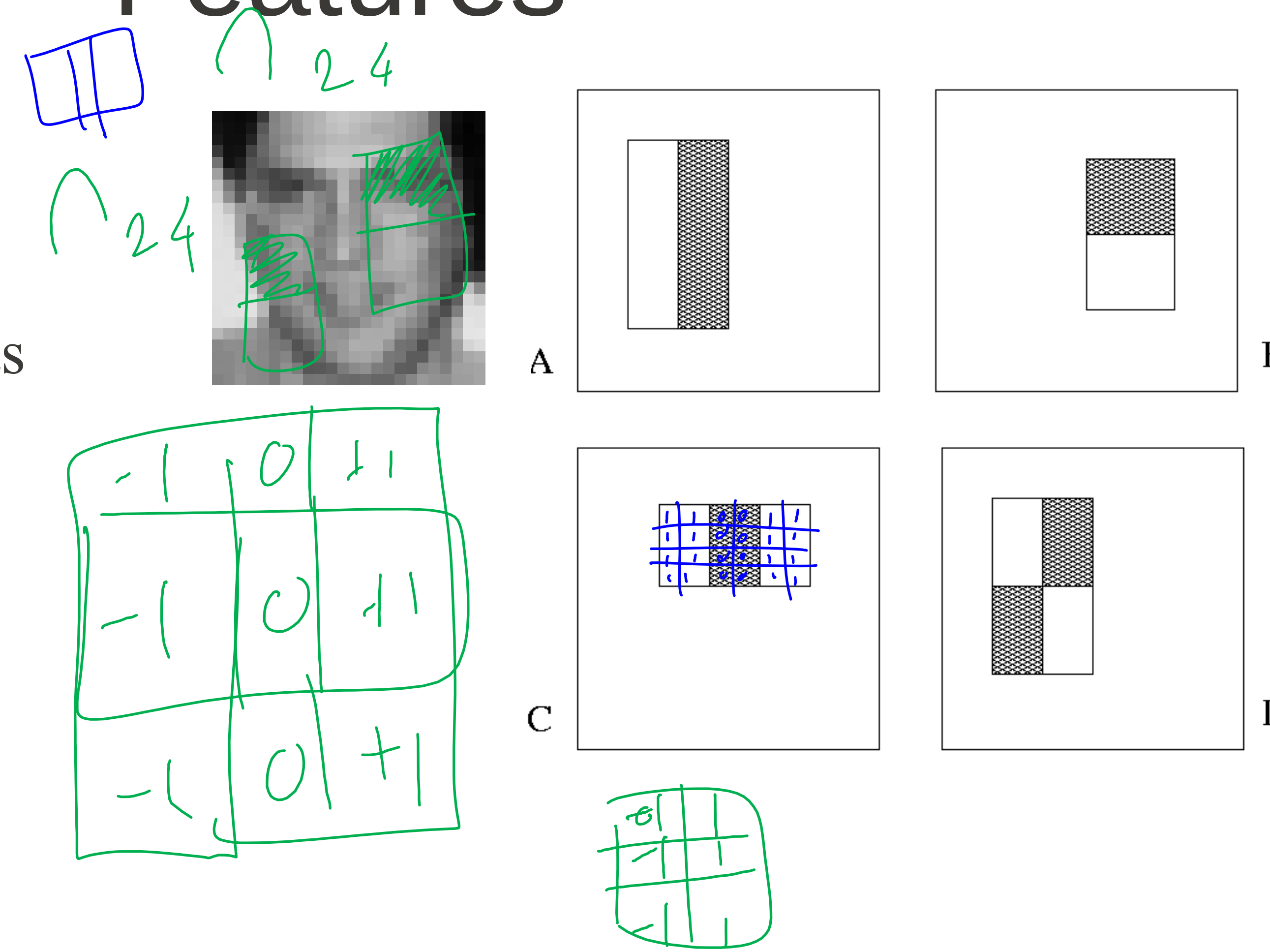
Boosted Face Detection: Image Features

"Rectangle filters"

Similar to Haar wavelets

Papageorgiou, et al.

Add white parts
Add dark parts
Use the difference

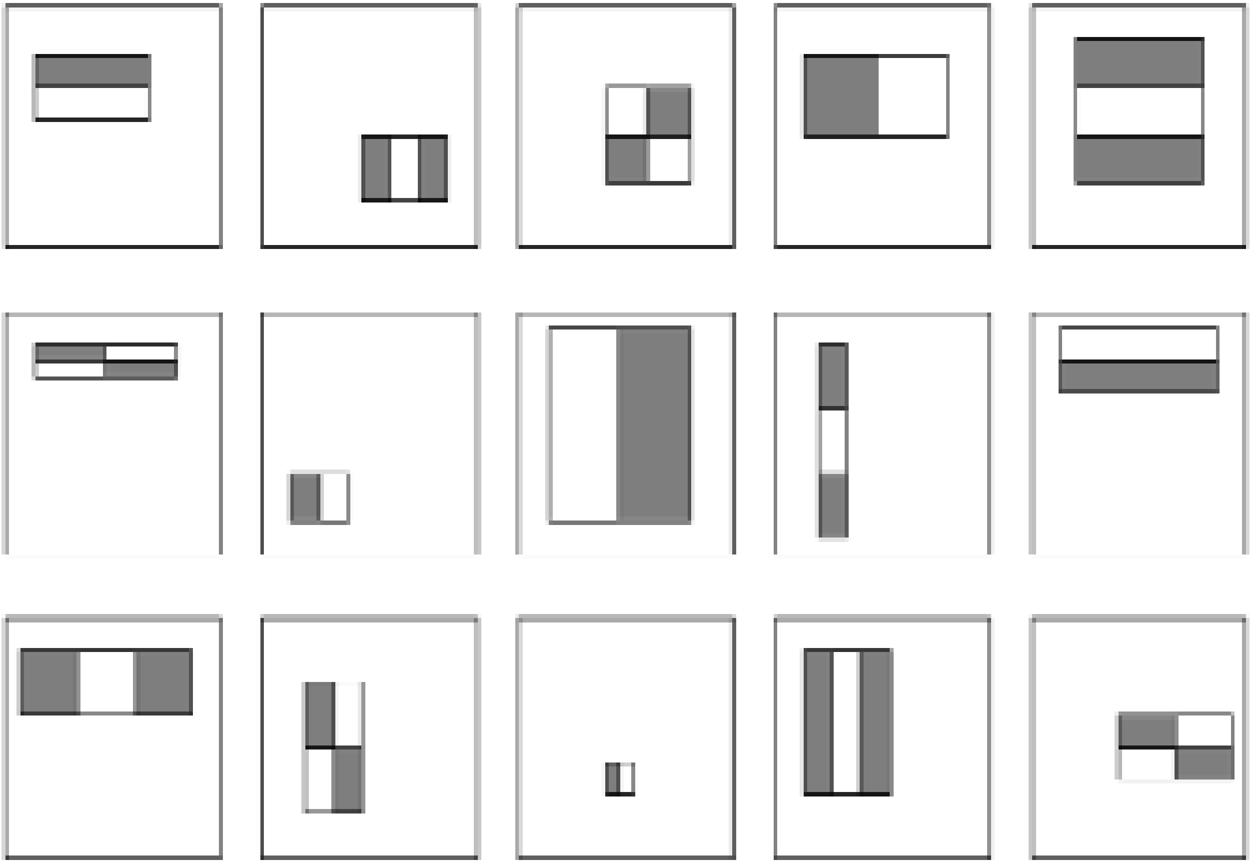


For a 24x 24 image 180,000 features to choose from

Boosting

Applying rectangle filters for faces

- For a 24x24 detection region, the number of possible rectangular features is ~180,000!



AdaBoost:Super Efficient Feature Selector

- Features = Weak Classifiers
- Each round selects the optimal feature given:
  - Previous selected features
  - Exponential Loss

Boosting

- Boosting is a classification scheme that works by combining weak learners into a more accurate ensemble classifier.
- Weak learner: Classifier with accuracy that need to be only better than chance
- Weak learners defined based on rectangular features:

Boosting 2

- Initially, assign equal weight to each training example
- Iterative training procedure

– Find best weak learner for current weighted training set

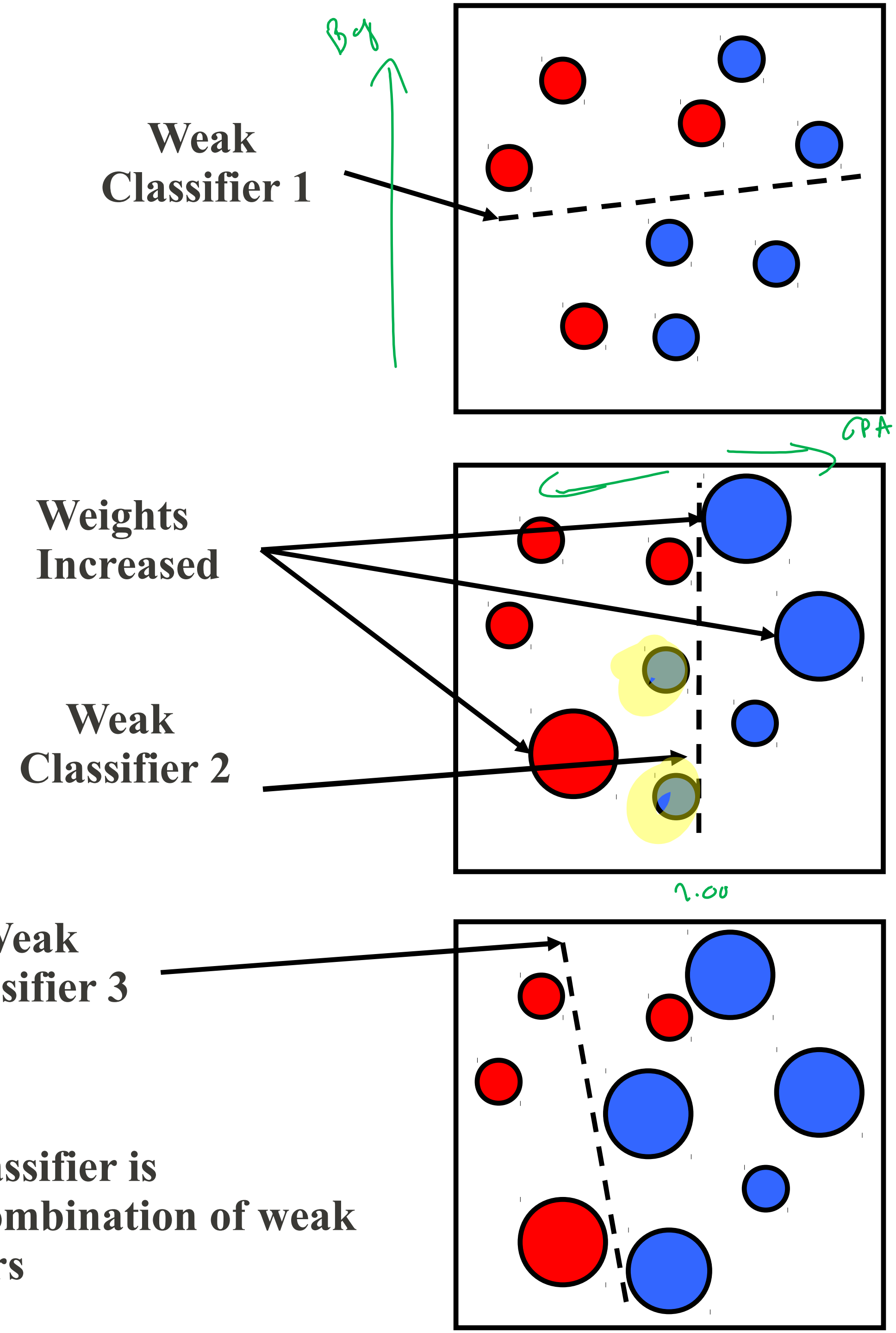
- Seek one feature with minimum error.

– Raise the weights of training examples misclassified by current weak learner

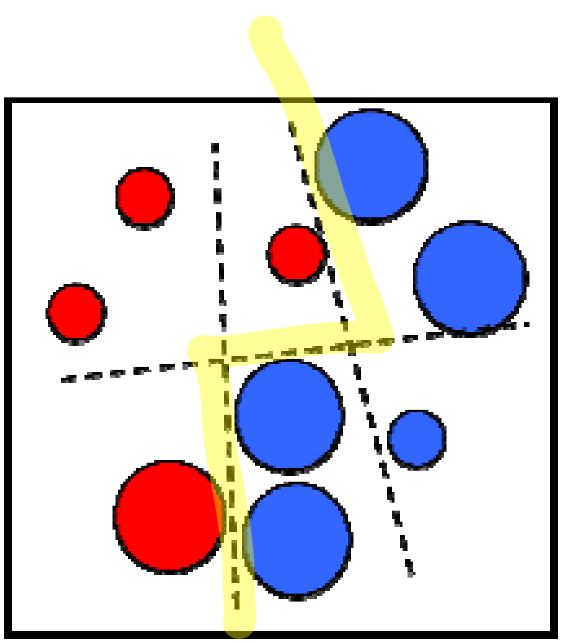
- Final classifier as a linear combination of all weak learners (weight of each learner is related to its accuracy)

AdaBoost

Freund & Shapire



Final classifier is linear combination of weak classifiers



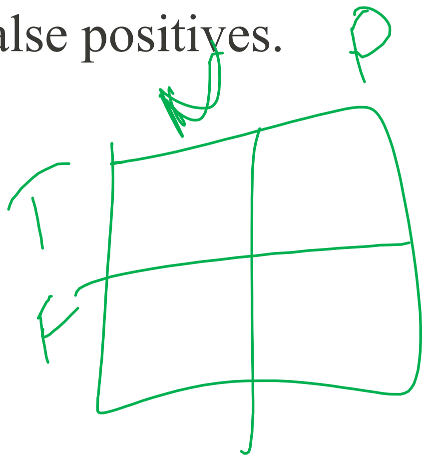
Example Classifier for Face Detection

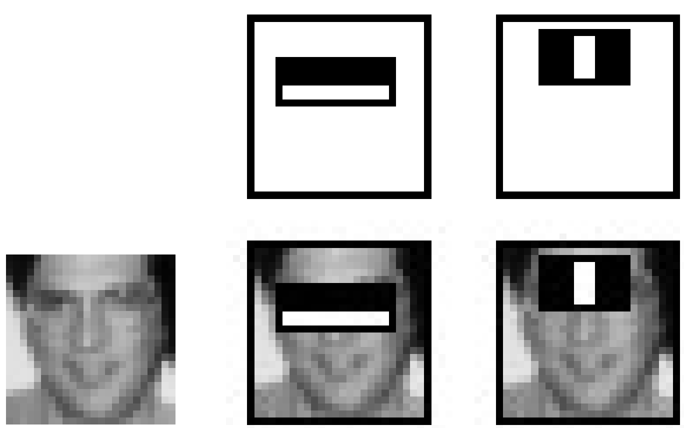
A classifier with 200 rectangle features was learned using AdaBoost

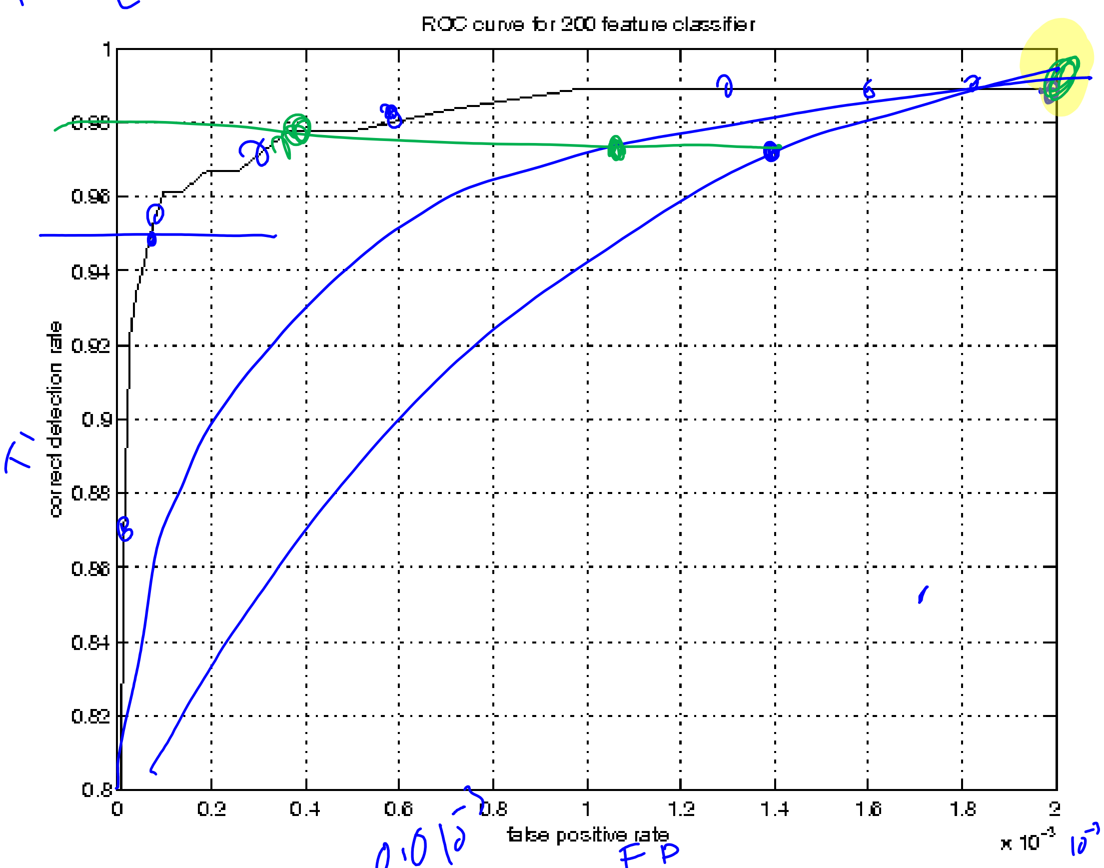
95% correct detection on test set with 1 in 14084 false positives.

$\alpha_1 w_1 + \alpha_2 w_2 + w_3 \alpha_3, \ldots$

ROC





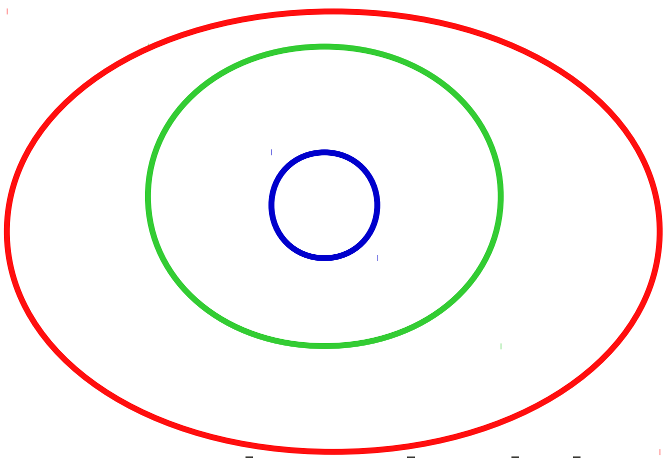


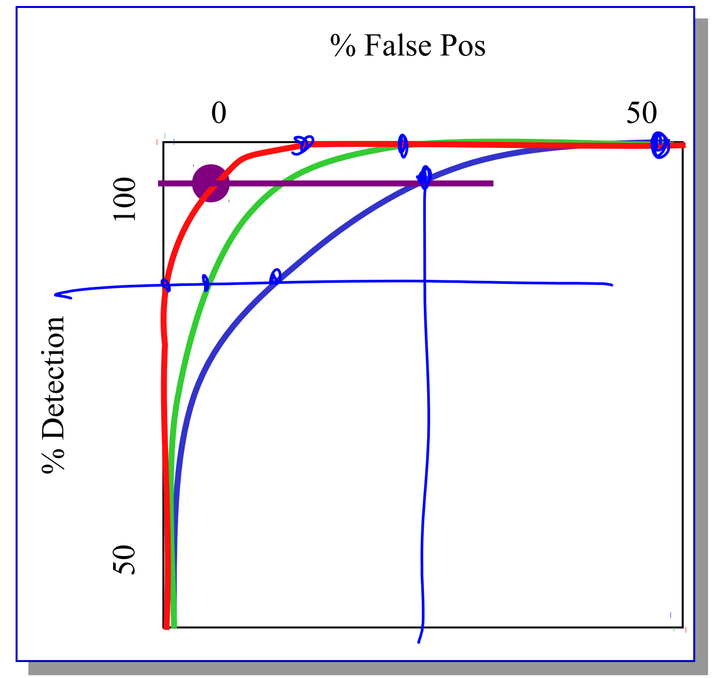
ROC curve for 200 feature classifier

Building Fast Classifiers

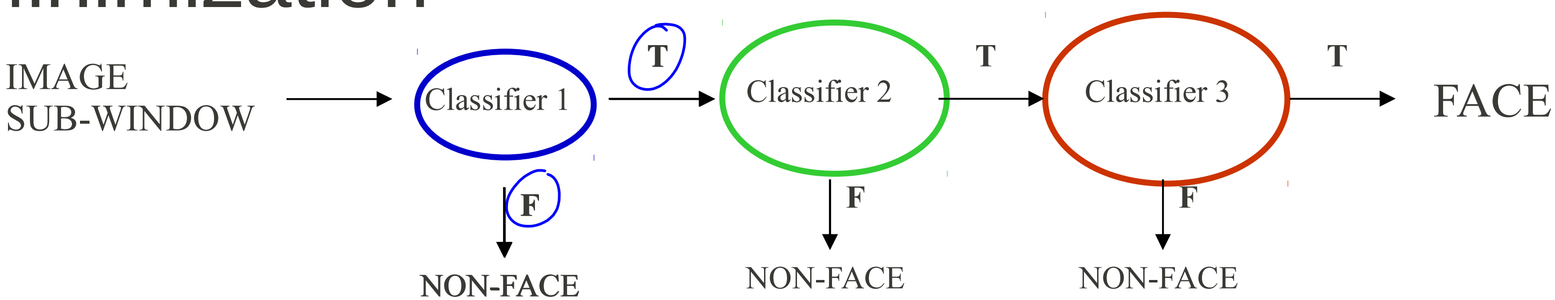
Cascaded classifiers

- Given a nested set of classifier hypothesis classes

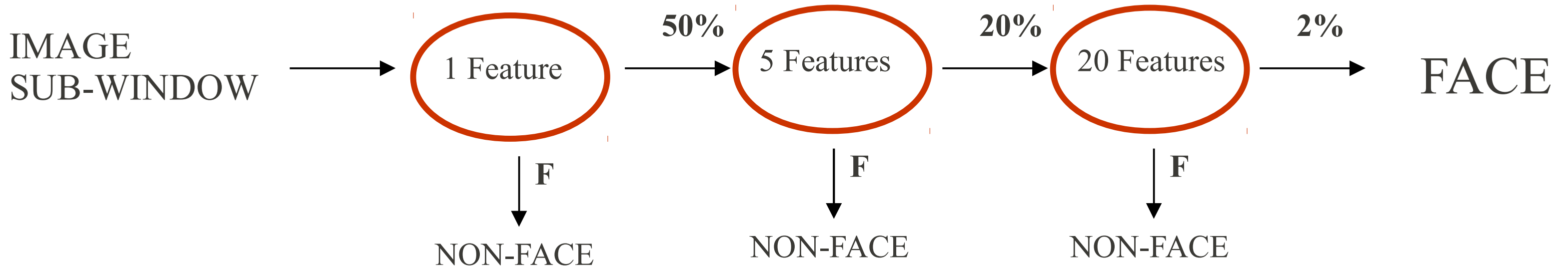




- Computational Risk Minimization

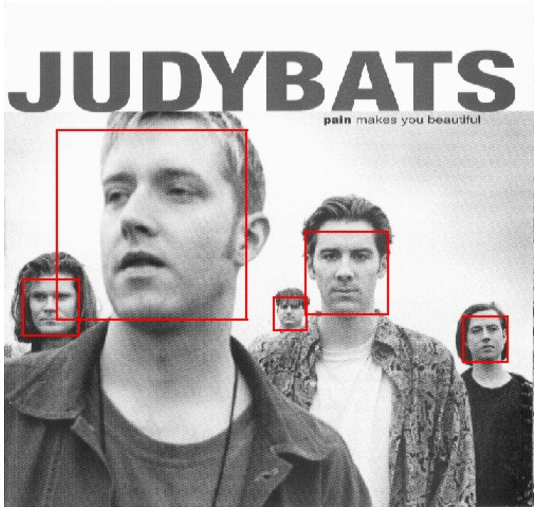


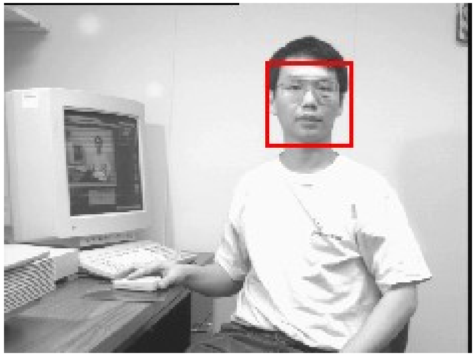
Cascaded Classifier

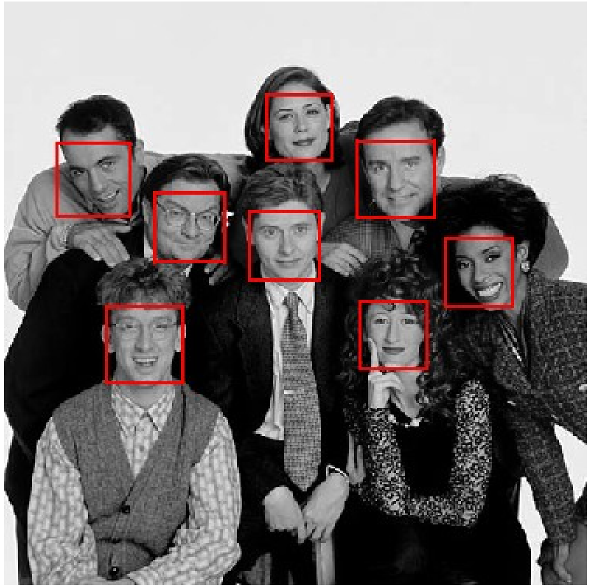


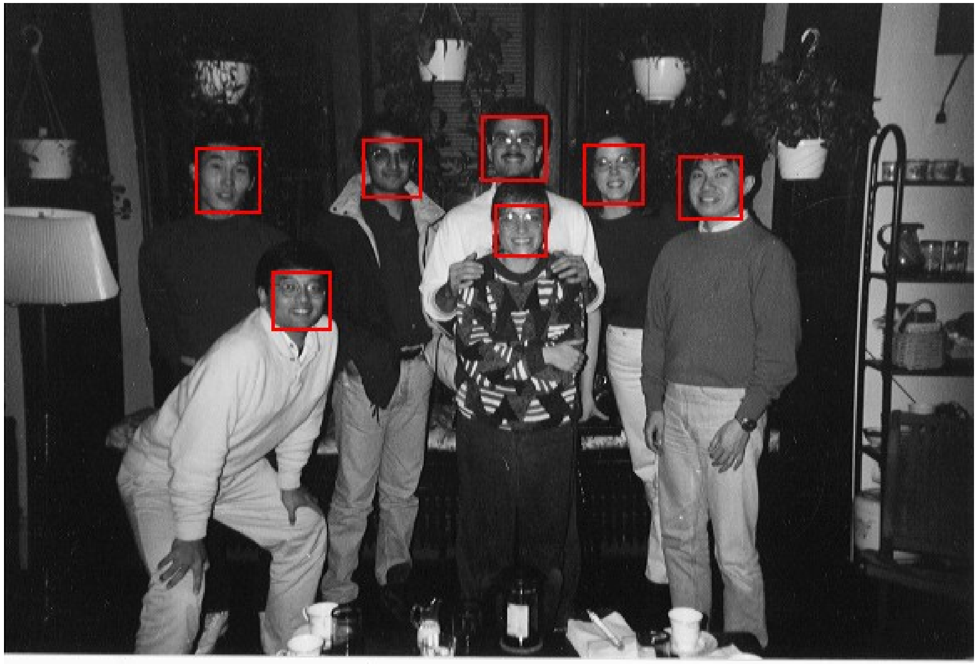
- A 1 feature classifier achieves 100% detection rate and about 50% false positive rate.
- A 5 feature classifier achieves 100% detection rate and 40% false positive rate (20% cumulative)
  - using data from previous stage.
- A 20 feature classifier achieve 100% detection rate with 10% false positive rate (2% cumulative)

Output of Face Detector on Test Images



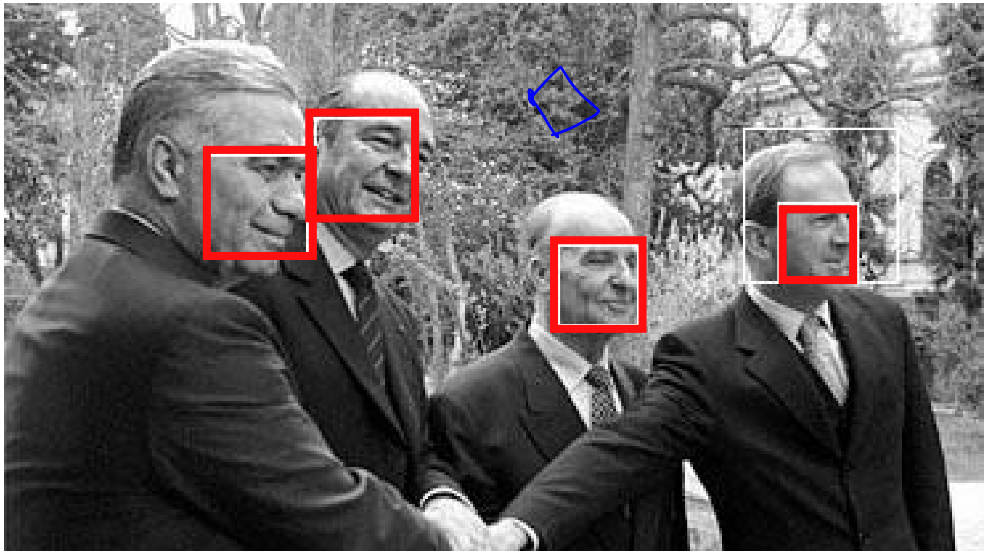






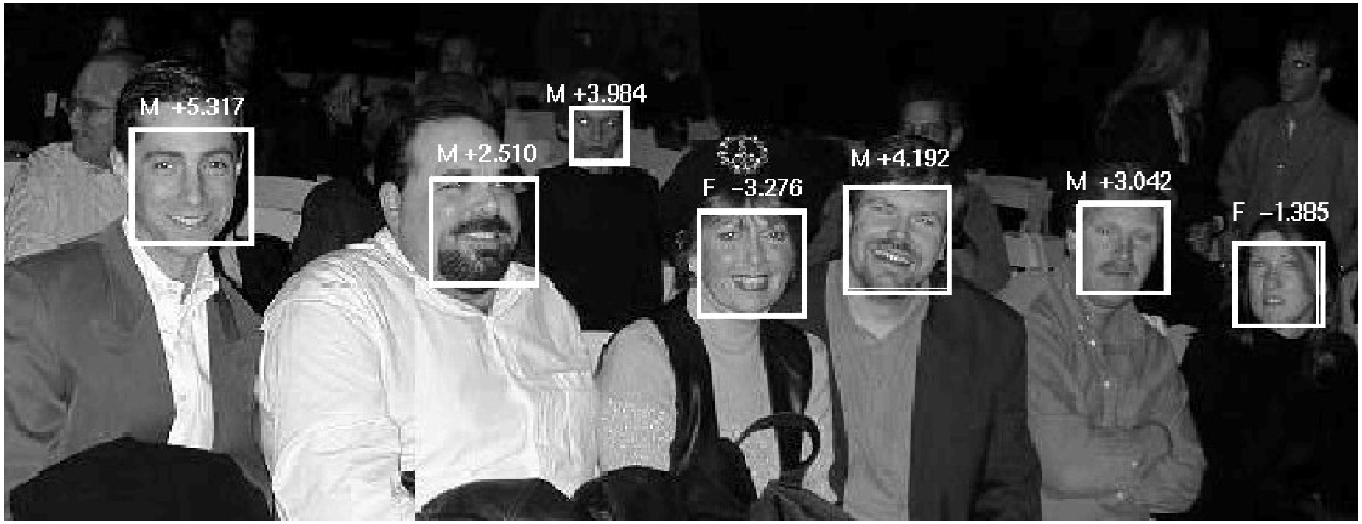
Solving other "Face" Tasks

Facial Feature Localization



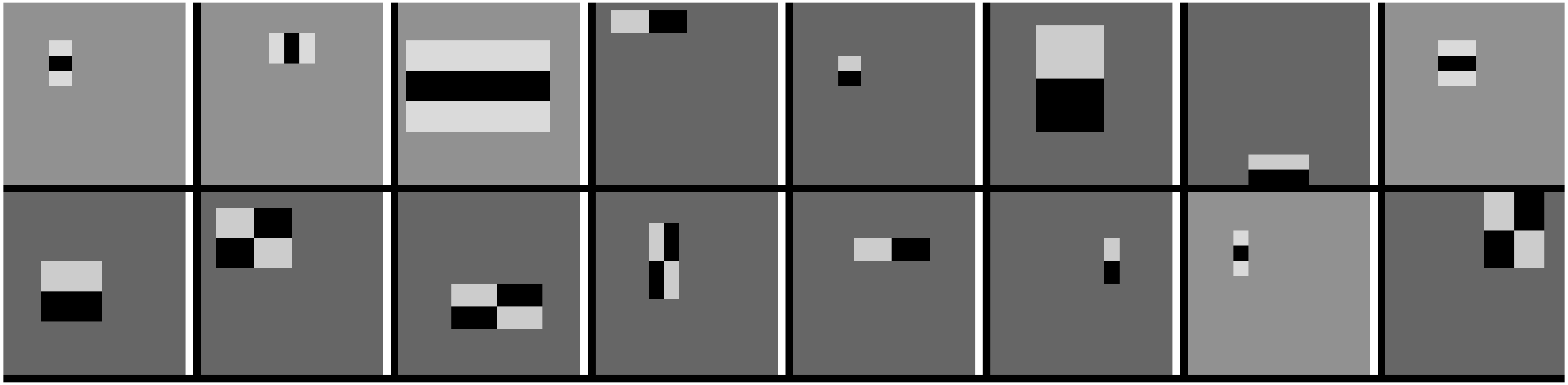
Profile Detection

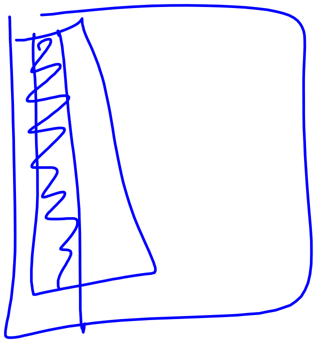
Demographic Analysis



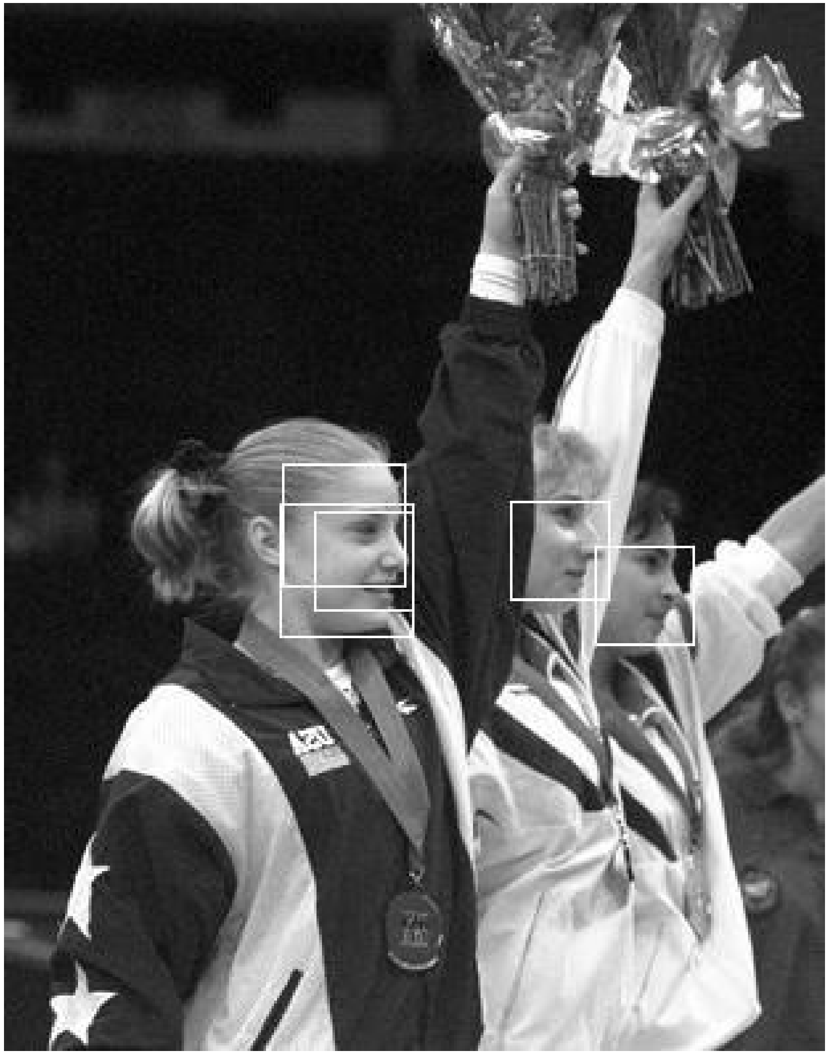
Feature Localization Features

- Learned features reflect the task

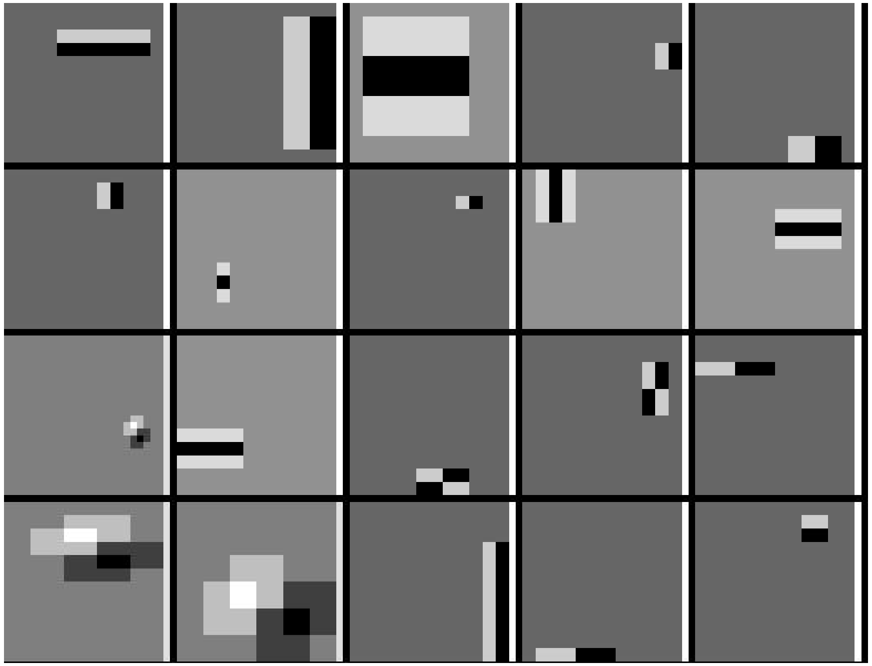




Profile Detection



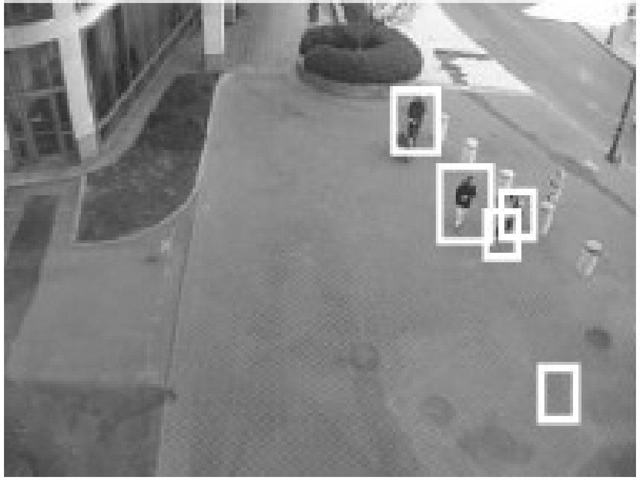
Profile Features

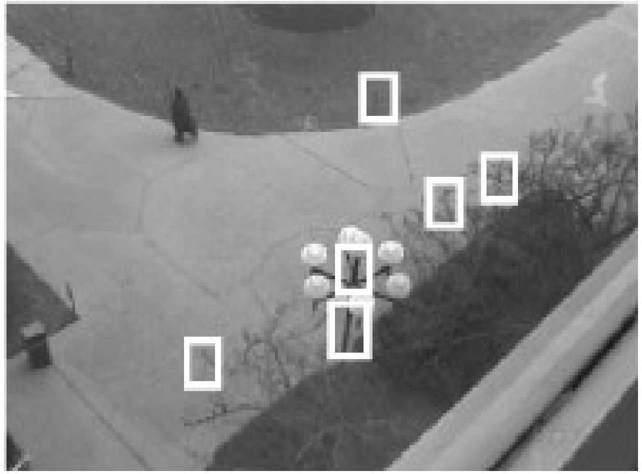


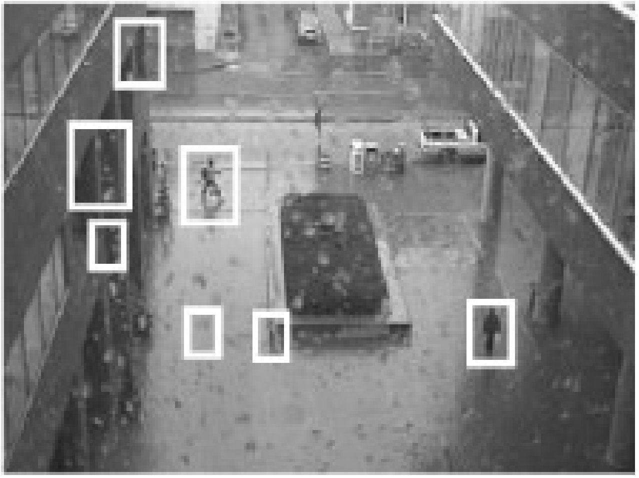
Back to Pedestrian Detection

- If we use exactly the same face detection system trained on pedestrians:

Pedestrians with face detection







Pedestrian detection

- We need to use extra features such as motion information

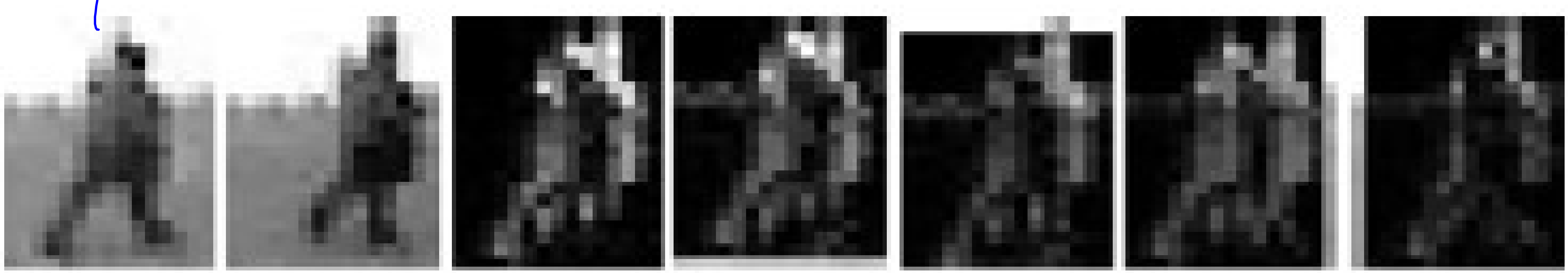
$$\Delta^l = abs(I_t^l - I_{t+1}^l)$$

$$U^l = abs(I_t^l - I_{t+1}^l \uparrow)$$

$$L^l = abs(I_t^l - I_{t+1}^l \leftarrow)$$

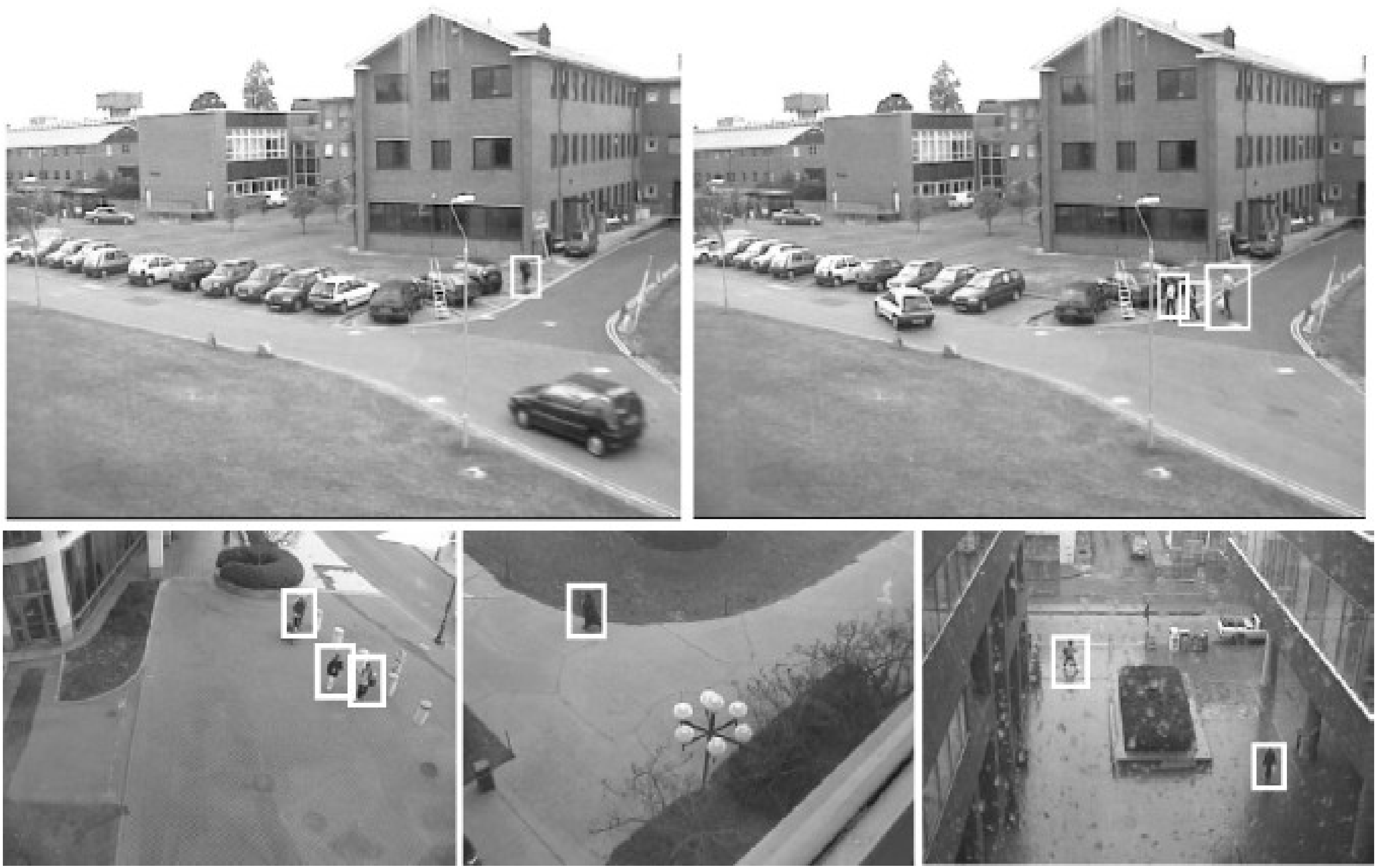
$$R^l = abs(I_t^l - I_{t+1}^l \rightarrow)$$

$$D^l = abs(I_t^l - I_{t+1}^l \downarrow)$$

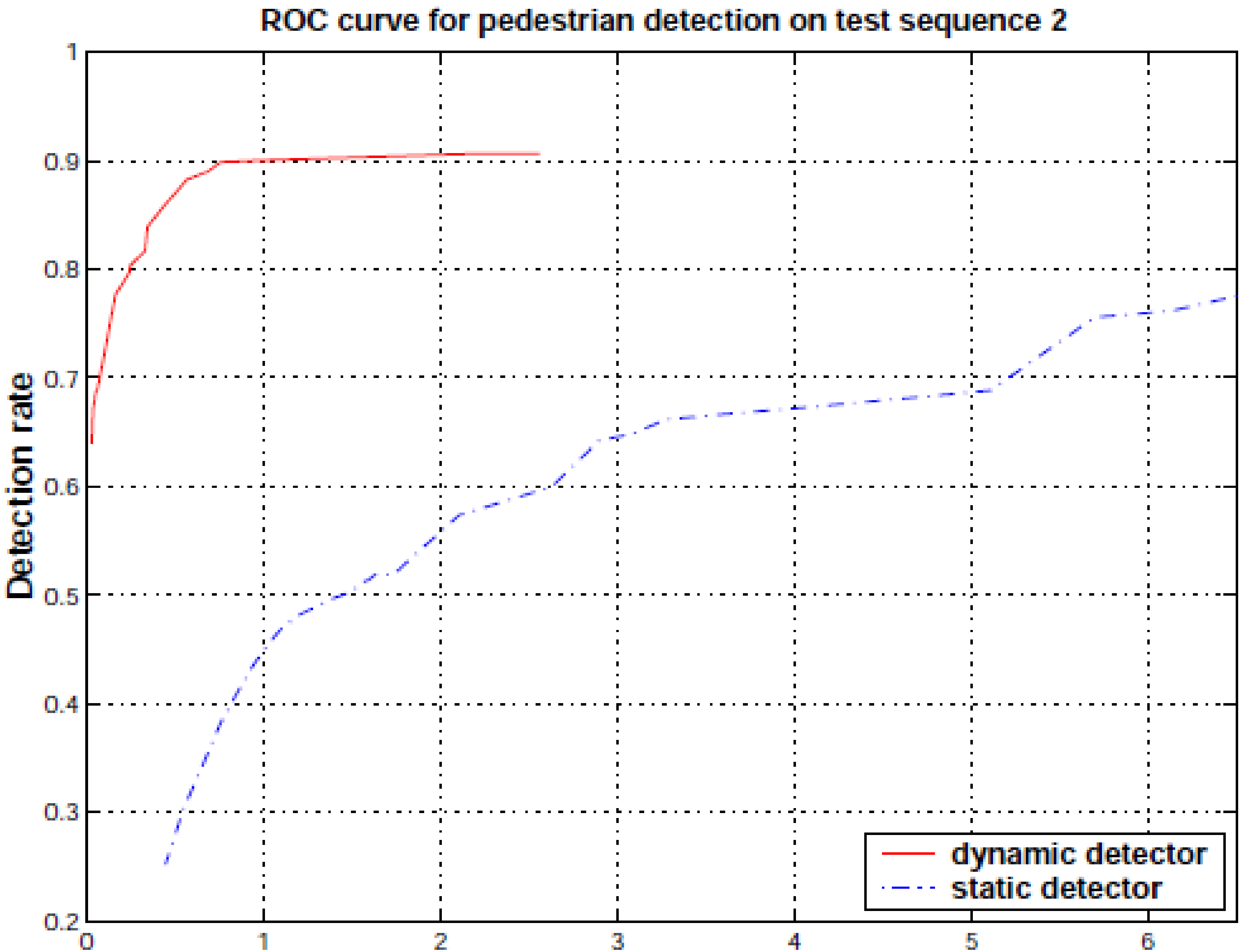


Frame 1 Frame 2 Δ U D L R

Pedestrians detection



Pedestrian Detection



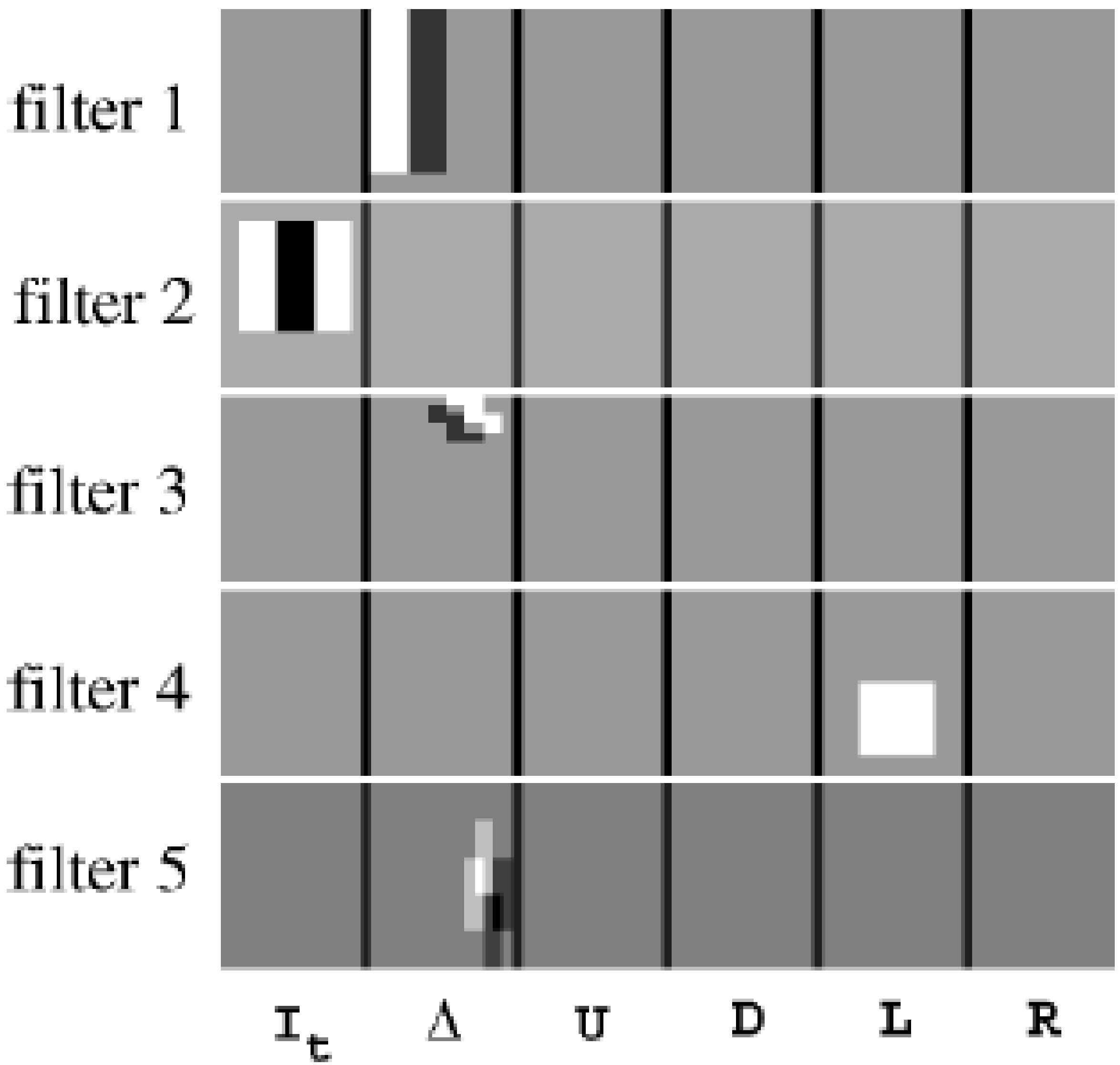


Figure 6: The first 5 filters learned for the dynamic pedestrian detector. The 6 images used in the motion and appearance representation are shown for each filter.