Face Detection

Ce Zhan

13th October 2009

Presentation outline

Introduction

What? *Face detection* involves finding the locations (bounding boxes) of an arbitrary number of faces within an image. It is a specific case of *object recognition*.

Why? Usually used as the first stage in a face recognition system.

How? Usually approached as a binary pattern classification problem.

Uses of face detection

Examples of uses include:

- Input for face recognition
- Human-machine interface (e.g. facial expression recognition)
- Auto-adjustments for digital photography
- Image database generation
- Face blurring for privacy

Figure: Face detection in Canon digital cameras (canon.com)

Pattern classification approach

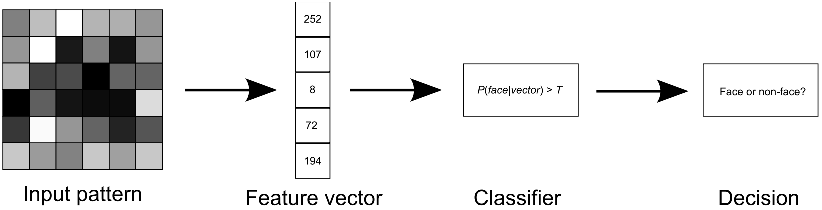


Figure: Basic model of a pattern classifier

Face detection model

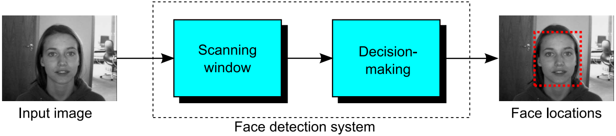


Figure: Basic face detection model. Multiple pattern classifiers are applied by the scanning window.

Popular methods in the literature

Seminal papers frequently cited and considered baseline methods:

- Turk & Pentland, 1991: PCA subspace method (“eigenfaces”)
- Osuna et al, 1997: non-linear support vector machine (SVM)
- Rowley et al, 1998: neural networks
- Schneiderman & Kanade, 2000: probabilistic wavelets
- Viola & Jones, 2001: AdaBoost ensemble of simple Haar-like features

Face detection methods

Two methods will be explained:

1. **Skin-based face detection** – faces detected based on colour and shape
2. **Viola & Jones face detection** – faces detected based on rectangular “Haar-like” features

Skin-based face detection

Skin tone can be used in face detection in two ways:

- **Segmentation** – To reduce the search space, speeding up detection but independent of the actual face detector (more common use)
- **Detection** – As a feature used directly for detection

We will consider the second case.

Statistical properties of skin colour

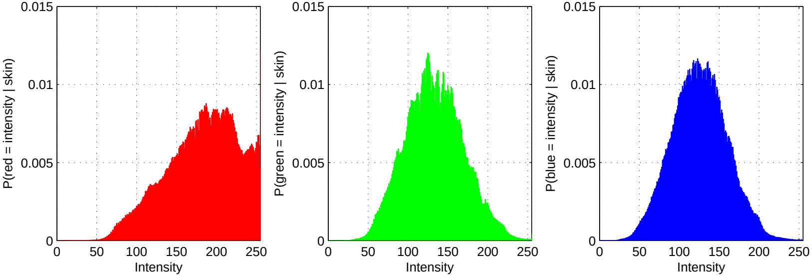


Figure: Probability distribution functions for skin colour channels (Gómez-Morales skin database)

Skin likelihood map

Phung et al, 2002, *A novel skin color model in YCbCr color space and its application to human face detection*, ICIP, vol. 1, pp 289-292:

Skin likelihood map

$$\Lambda(skin|c) = \frac{p(c|skin)}{p(c|nonskin)}$$

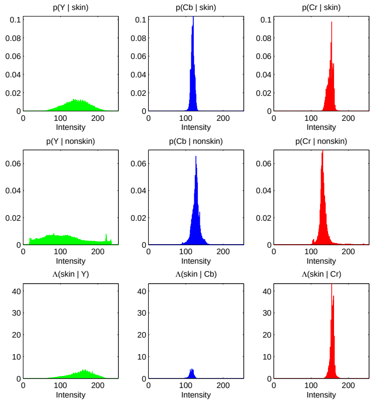


Figure: Skin likelihood (YCbCr)

Example of skin likelihood

Figure: Skin likelihood trained on Gómez-Morales skin database, tested on a Caltech face database image

Skin RGB thresholds

Kovač et al, 2003, *Human skin colour clustering for face detection*, International Conference on Computer as a Tool, pp 144-148:

Skin detection using simple thresholds

```
R > 95 AND G > 40 AND B > 20 AND max(R,G,B)-min(R,G,B)
> 15 AND |R-G| > 15 AND R > G AND R > B

OR

R > 220 AND G > 210 AND B > 170 AND |R-G| <= 15 AND R
> B AND G > B
```

Demo application

1. Threshold the skin regions based on colour (using the method of Kovač)
2. Remove small regions (morphological opening)
3. Find aspect ratio of all regions (connected components labelling) and only retain those within a certain range
4. Remaining regions are classified as faces

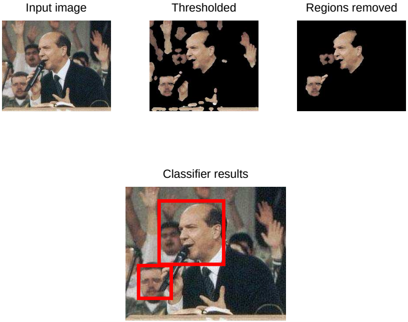


Figure: Example classification

Pros and cons of skin-based face detection

Advantages

- Efficient and simple to implement
- No or little training required
- Works on low resolution or blurred images – no face details required
- Rotation-invariant

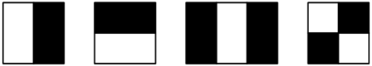
Disadvantages

- Sensitive to illumination and colour balance
- Poor accuracy – matches other body parts and skin coloured objects and mismatches beards, sunglasses, etc
- Requires colour images

Viola & Jones face detection

- Very popular method proposed by Paul Viola (MERL) and Michael Jones (Compaq) in 2001
- Uses a pattern classification approach:
    - Features – rectangular "Haar-like" features
    - Classifier – combination of multiple "weak" classifiers (AdaBoost)

Haar-like features



(a) Haar-like features

(b) Applied to a face (Viola & Jones, 2001)

Figure: Haar-like features

- Haar-like features encode intensity differences
- Feature value is summed pixel difference between white and black regions
- Scaled to different sizes and tested at all possible locations within the sub-window
- Sub-windows normalised for global illumination invariance

Example feature extraction

$$h_1 = 7.5 - 12.6 = -5.1$$

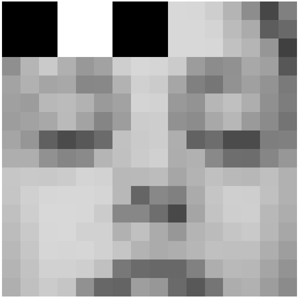


Figure: Haar-like feature extraction

Example feature extraction

$$h_2 = 7.6 - 14.0 = -6.3$$

Figure: Haar-like feature extraction

Example feature extraction

$$h_3 = 7.6 - 14.8 = -7.2$$

Figure: Haar-like feature extraction

Example feature extraction

$$h_4 = 7.6 - 15.1 = -7.4$$

Figure: Haar-like feature extraction

Example feature extraction

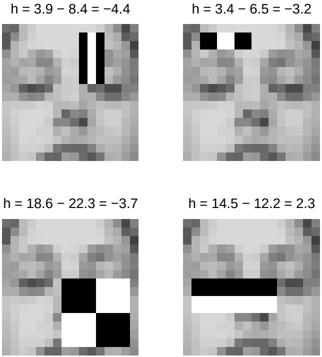


Figure: Feature value calculations

- All four Haar-like features are applied, all locations and sizes
- Produces a very large number of feature values; 2×10^8 values for a 24×24 image
- These are fed into the classifier (rather than the original image pixel values)

Single feature classifiers

- Simple “stump” classifier uses a single feature and thresholds it
- There is a large possible number of such classifiers (one for each feature value, i.e. 2×10^8) – need a way to select the best ones
- This is where AdaBoost comes in

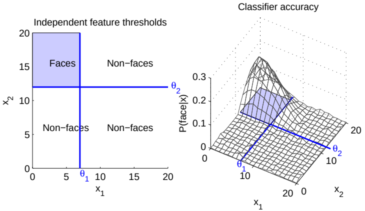


Figure: Applying two simple stump classifiers

AdaBoost

- Introduced by Freund & Schapire (1999)
- Combines and weights a large number of classifiers ("weak classifiers") to generate a more accurate classifier ("strong classifier"):

$$F(\mathbf{x}) = \sum_{m=1}^{M} \alpha_m f_m(\mathbf{x})$$

- Serves two purposes:
  - Feature selection – selects the most appropriate subset of all of the possible stump classifiers (features) by re-weighting them
  - Classifier – generates the strong classifier which makes the final face/non-face decision

AdaBoost

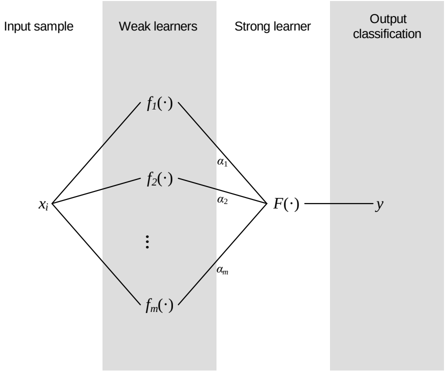


Figure: AdaBoost as a network

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:

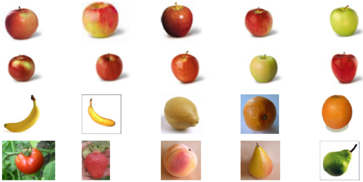


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:
  1. Circular

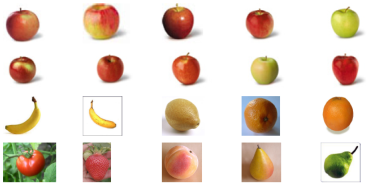


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:
  1. Circular
  2. Usually red

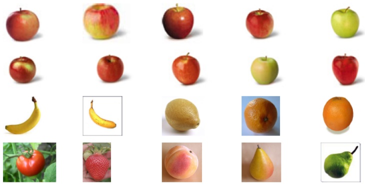


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:
  1. Circular
  2. Usually red
  3. Possibly green

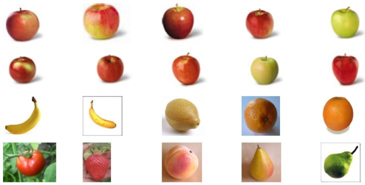


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:
  1. Circular
  2. Usually red
  3. Possibly green
  4. Have a stem

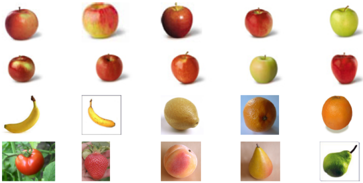


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- Multiple rules used in combination
- Apples are:
  1. Circular
  2. Usually red
  3. Possibly green
  4. Have a stem
- These "weak" rules can be combined to generate a "strong" rule

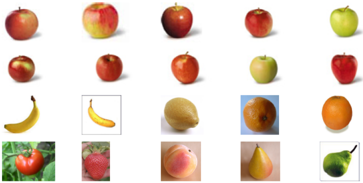


Figure: Classifying fruit (Lin, 2005)

Intuition behind AdaBoost

- The weak classifiers are selected one by one
- The classifiers with higher accuracy (during training) are weighted higher than those with lower accuracy (using the logit function):

$$\alpha_m = \log \frac{1 - \epsilon_m}{\epsilon_m}$$

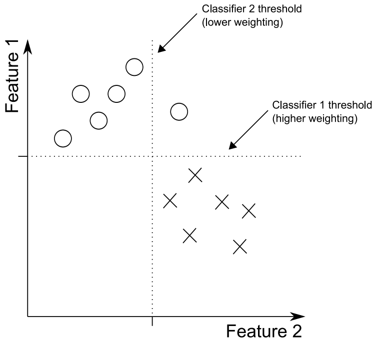


Figure: Weighting classifiers

AdaBoost training

- Trained with thousands of face and non-face images, all cropped and scaled to the same size (e.g. 24×24 pixels)
- Adding weak learners during training drives down the training error
- Some authors use more sophisticated weak learners, such as tree classifiers using multiple features each

Training steps

AdaBoost selects weak learners (features) one by one that successively focus on the more "difficult" samples to classify.

1. Initialise sample weights
2. Repeat for M rounds:
    1. Train all possible weak learners and select the one with the lowest weighted error
    2. Update and re-normalise sample weights based on weak learner output
3. Output final strong classifier as sum of weak classifiers

Viola & Jones train multiple strong classifiers, and all must be passed to be classified as a face.

Classification

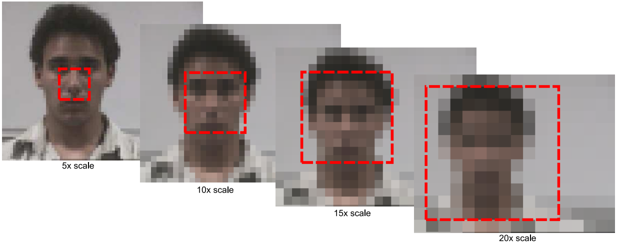


Figure: Scanning sub-window

- A sub-window sweeps the image at different resolutions, applying the classifier at each position
- Each sub-window is classified as a face or non-face

Example of classifications at multiple resolutions

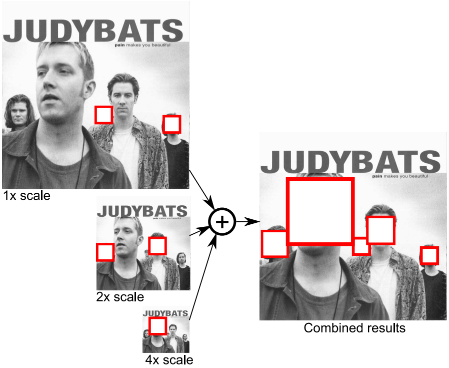


Figure: Scale-invariant face detection

Merging detections

- Multiple detections generally occur at nearby scales and spatial locations
- Merging algorithm combines them, usually based on proximity, percentage of overlap, etc
- Single isolated detections are often false positives (e.g. see top-right detection in figure)

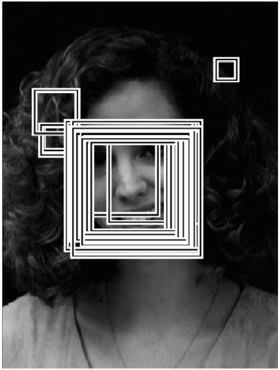


Figure: Multiple detections must be merged (Rowley, 1998)

AdaBoost variants

Many variants of AdaBoost in the literature, the ones most often used for face detection are:

- **Discrete AdaBoost** – weak classifiers output discrete values of $\{+1,-1\}$ and strong classifiers are constructed as a weighted linear combination of weak classifiers
- **Real AdaBoost** – generalisation of discrete AdaBoost where weak classifiers return a class probability estimate between $[0,1]$
- **Gentle AdaBoost** – modification of real AdaBoost that uses adaptive Newton steps rather than exact optimisation

Efficient implementation

Viola & Jones suggest two techniques to speed up the algorithm:

1. The Haar-like features can be calculated quickly using an *integral image* representation
2. The classification can be sped up using an *attentional cascade*

Integral image

- New image where every pixel is the sum of the pixels above and to the left of it in the original image:

$$I(x,y) = \sum_{x'=0}^{x} \sum_{y'=0}^{y} f(x',y')$$

- Speeds calculation of Haar-like features, trades memory for speed
- Calculated in a single pass
- Allows the computation of a rectangle with just 4 look-ups in the integral image

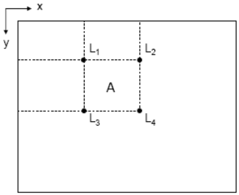


Figure: Calculation of a rectangle's area

Attentional cascade

- Cascade is formed as a series of strong classifiers
- Samples can be rejected at any stage; faces are a *rare event*, so non-faces should be processed quickly
- Most non-faces can be eliminated quickly in the first few nodes
- Each node is slightly more complex than the last and 'focuses attention', because each node is trained with the false positives of the prior

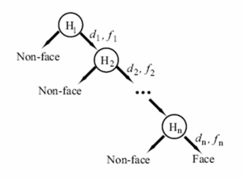


Figure: Attentional cascade allows early exits for non-faces

Extensions to Viola & Jones

Some have different features (e.g. for detecting tilted heads), others use different classifiers (e.g. more accurate AdaBoost variants)

- Jones, 2003: non-frontal faces
- Lienhart, 2003: Gentle AdaBoost
- Lienhart, 2003: rotated features, multi-view
- Baluja, 2004: improved training speed
- Li, 2004: floating feature selection
- Torralba, 2004: joint boosting
- M.Yang, 2006: hardware implementation
- T.Yang, 2006: video-based face tracking

. . . and many more

Demo application

- OpenCV uses a detector proposed by Lienhart (2003)
- Intel open source C++ library for computer vision and machine learning
- Provided with several pre-trained classifiers (frontal face, profile face, human body, etc)

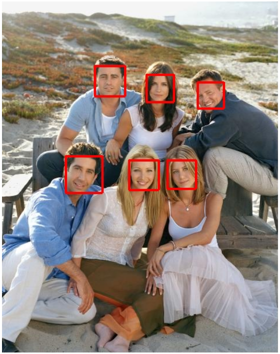


Figure: Example classification

Salient facial features

Interesting observations can be made (which agree with neuroscience literature, e.g. Sinha 2006):

- Features are selected from *most important* to *least important*
- Usually eyes and eyebrows, followed by the nose, and then the rest of the face
- Low spatial frequency to high spatial frequency ("coarse to fine")
- Most appearance-based face detectors follow this same sequence

Pros and cons of Viola & Jones face detection

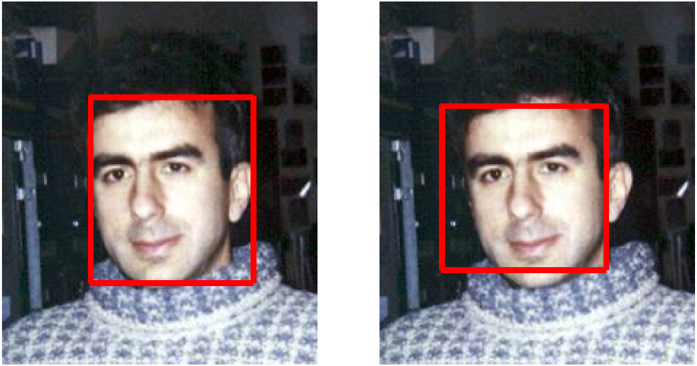
Advantages

- High accuracy
- Reasonably efficient
- Works on greyscale images

Disadvantages

- Long training process required
- Variable classification time
- Sensitive to pose

Examples from the Caltech web faces database



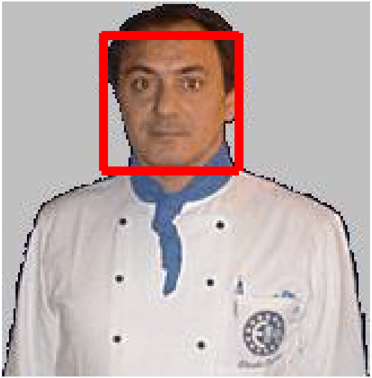
(a) Skin-based detector

(b) Viola & Jones detector

Figure: Simple portrait image

Example classifications

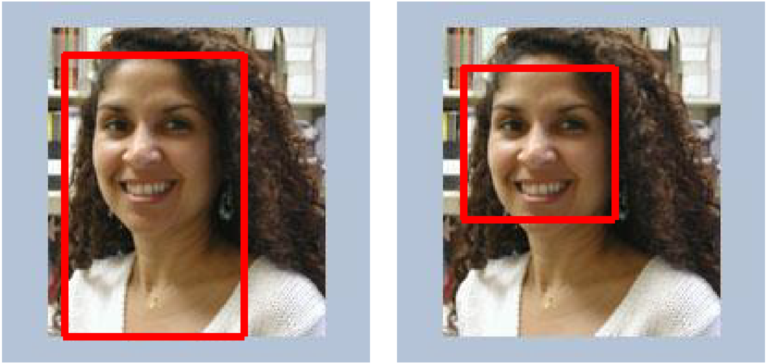
(a) Skin-based detector



(b) Viola & Jones detector

Figure: Simple portrait image

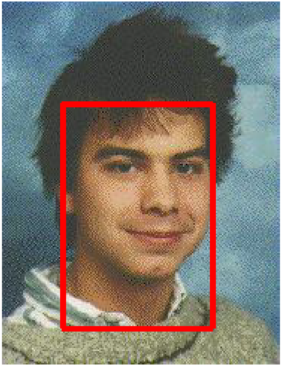
Example classifications



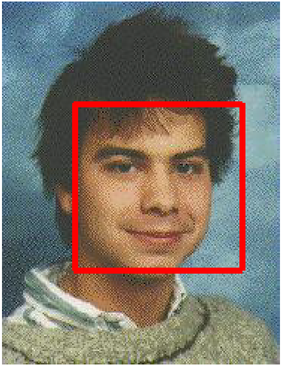
(a) Skin-based detector (b) Viola & Jones detector

Figure: Simple portrait image

Example classifications



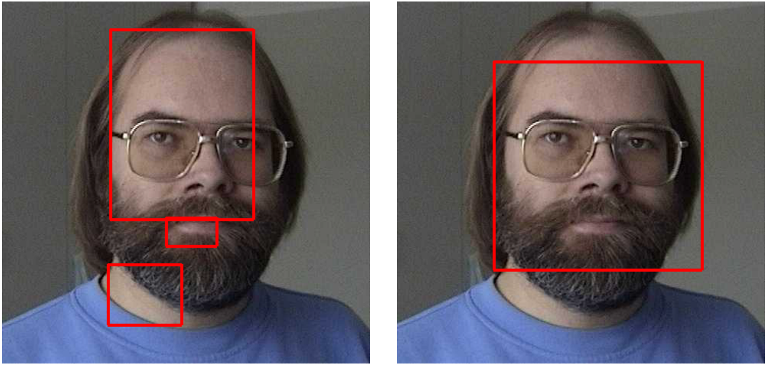
(a) Skin-based detector



(b) Viola & Jones detector

Figure: Simple portrait image

Example classifications

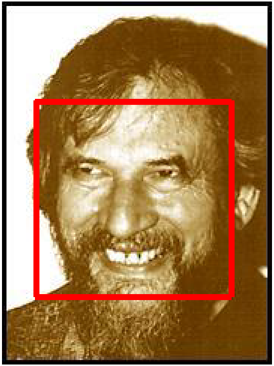


(a) Skin-based detector (b) Viola & Jones detector

Figure: Beard

Example classifications

(a) Skin-based detector



(b) Viola & Jones detector

Figure: Beard

Example classifications

(a) Skin-based detector

(b) Viola & Jones detector

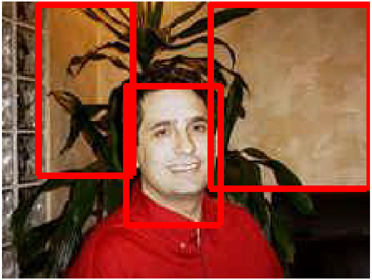
Figure: Moustache

Example classifications

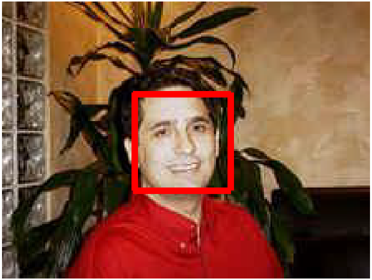
(a) Skin-based detector (b) Viola & Jones detector

Figure: Beard

Example classifications



(a) Skin-based detector



(b) Viola & Jones detector

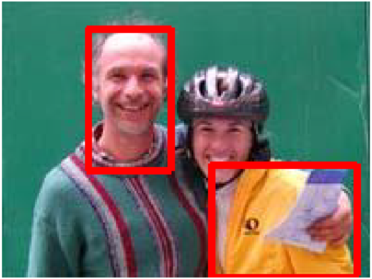
Figure: Skin-coloured background

Example classifications

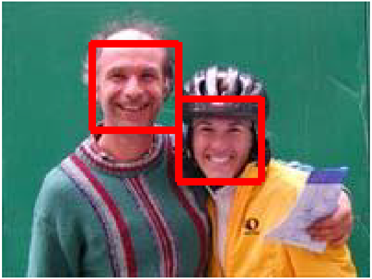
(a) Skin-based detector (b) Viola & Jones detector

Figure: Skin-coloured background

Example classifications



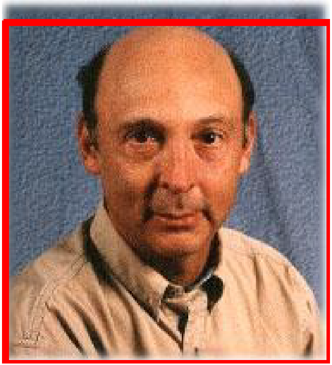
(a) Skin-based detector



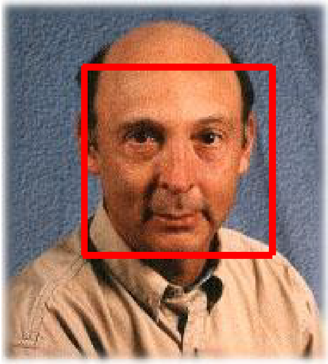
(b) Viola & Jones detector

Figure: Skin-coloured clothing

Example classifications



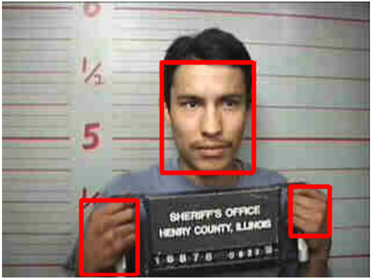
(a) Skin-based detector



(b) Viola & Jones detector

Figure: Skin-coloured clothing

Example classifications

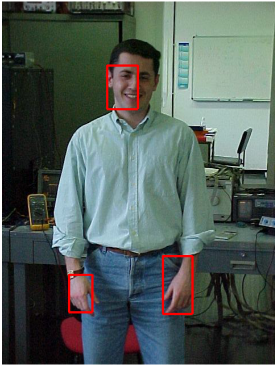


(a) Skin-based detector

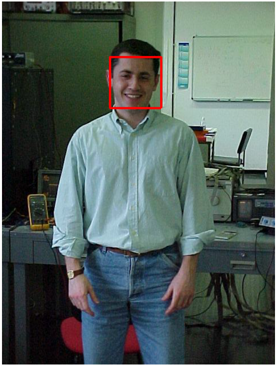
(b) Viola & Jones detector

Figure: Other body parts

Example classifications



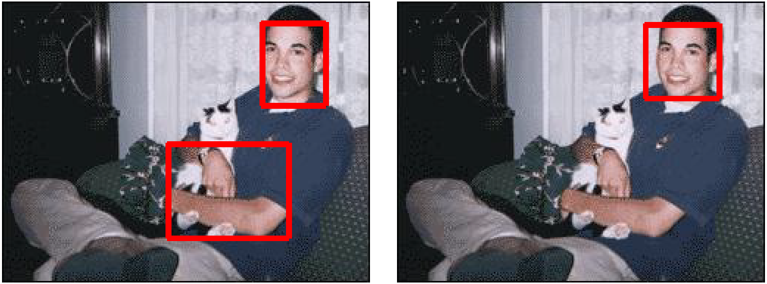
(a) Skin-based detector



(b) Viola & Jones detector

Figure: Other body parts

Example classifications

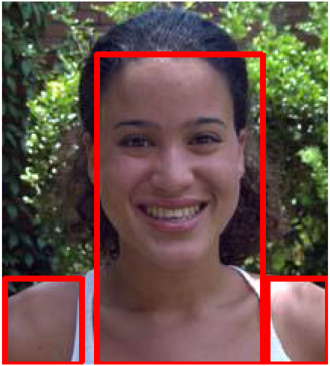


(a) Skin-based detector

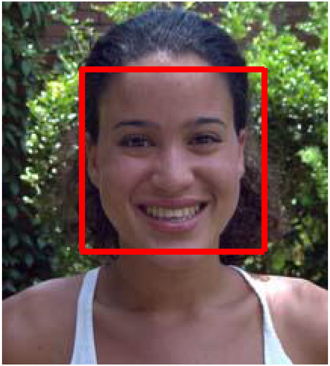
(b) Viola & Jones detector

Figure: Other body parts

Example classifications



(a) Skin-based detector



(b) Viola & Jones detector

Figure: Other body parts

Example classifications

(a) Skin-based detector

(b) Viola & Jones detector

Figure: Illumination

Example classifications

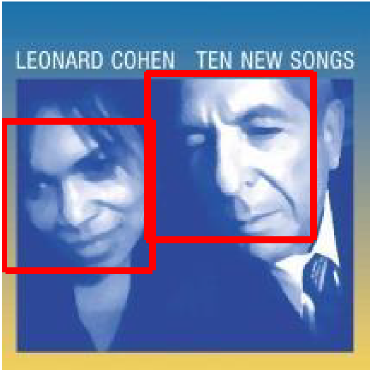
(a) Skin-based detector

(b) Viola & Jones detector

Figure: Non-frontal pose angle

Example classifications

(a) Skin-based detector



(b) Viola & Jones detector

Figure: Colour balance

Example classifications

(a) Skin-based detector

(b) Viola & Jones detector

Figure: Colour balance

Challenges in face detection

Can be summarised as:

- Low quality or low resolution images
- Occlusion by foreign objects
- Illumination variations (or camera settings)
- Facial expressions
- Pose (in-plane and out-of-plane rotation)

Figure: Examples of illumination variations (CMU-PIE database)

Detection Vs recognition

- Same feature can be uesed for both tasks to represent the face
- *Have opposing demands*
    - Detection requires extracting what is common to all faces
    - Recognition of individuals requires a fine-grained analysis to extract the ways in which each face differs from the others despite the fact that all faces share the same basic configuration
    - Detection can act as a domain-specific filter to make the recognition process more efficient

Bibliography

M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34–58, January 2002.

P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, IEEE Computer Society, 2001.

J. Kovač, P. Peer, and F. Solina, "Human skin colour clustering for face detection," in *International Conference on Computer as a Tool*, pp. 144–148, 2003.

Thanks! Any questions?