ECE417 Guest Lecture
Real-time Face Detection

Zhen Li, with many slides from Jason X. Xu
Feb 19, 2013
Problem: Face Detection

• Given an image, determine whether there exist human faces:

![No Faces Image]
Problem: Face detection

- If there are human faces, mark their location and size.
Why is it important?

• First step in any fully automatic face recognition system
  – Person identification
  – Soft-biometrics: gender, age, ethnicity
• First step in many surveillance systems
• Lots of applications
  – Digital camera: face sensitive auto focus
• A step towards Automatic Target Recognition (ATR) or generic object detection/ recognition
  – One of the core problems in computer vision and pattern recognition
Why is it difficult?

- **Background**
  - Highly cluttered, unpredictable

- **Rotation**
  - In-Plane Rotation (Orientation): Rotations about the camera's optical axis
  - Out-of-Plane (Pose): Frontal, 45 degree, profile

- **Facial expression**

- **Occlusion**
  - Beards, mustaches, glasses

- **Imaging conditions**
  - Lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, gain control, lenses), resolution
Why is it difficult?
Methodologies

• **Knowledge Based**
  – Manually encode human knowledge of what constitutes a typical face (usually, the relationships between facial features)

• **Component Based**
  – Aim to find structural components (facial features) of a face that exist even when the pose, viewpoint, or lighting conditions change

• **(Holistic) Appearance Based**
  – Machine learning
  – The models (or templates) are learned from a set of training images which capture the representative variability of facial appearance
Knowledge Based Approach

• Building a set of human-coded rules, for example:
  – The center part of face has uniform intensity values
  – The difference between the average intensity values of the center part and the upper part is significant
  – A face often appears with two eyes that are symmetric to each other, a nose and a mouth

• Using these rules to search faces
Knowledge Based Approach

- Yang and Huang, 1994
  - Multi-resolution (focus-of-attention)
Knowledge Based Approach
Yang and Huang, 1994 - level 2

Nose rules. There may be a nose region, if
- there is a local minimum of grey levels in the vertical direction under the eye centered at 1 unit of distance from the center, and the grey level is near the local minimum in the horizontal direction.

Mouth rules. There may be a mouth region, if
- there is a local minimum of grey levels in the vertical direction under the nose at 1 unit of distance, and in the horizontal direction the grey level has a 2-3 unit region with low grey levels.
Knowledge Based Approach

- Kotropoulos and Pitas, 1994
  - Horizontal/vertical projection of pixel values
Knowledge Based Approach

• Pros:
  – Easy to implement
  – No training required
  – Usually fast

• Cons:
  – It’s hard to describe human knowledge precisely and comprehensively
    • Pose, illumination, ...
  – Not robust to cluttered background
Component Based Approach

- Bottom-up approach
- Detecting facial features (eyes, nose, mouth etc.) first
- Grouping features into face candidates and verify them
Component Based Approach

- Leung et al, 1995
Component Based Approach

• Pros:
  – Facial feature detection are robust to pose

• Cons:
  – Difficult to locate facial features due to illumination, imaging noise
  – Difficult to detect facial features in complex (cluttered) background

• Comment:
  – Revived in very recent years, due to introduction of better low level features (e.g. SIFT) and component detectors (e.g. AdaBoost)
  – Promising in handling occlusion and pose variation
Appearance Based Approach

- Exhaustively scanning the whole image
- At each location, extracting an image patch, determine whether it is a human face or not
- Reporting all locations where faces were found

- How to detect faces of variable
  - Sizes: Scaling the input image
  - Orientation: Rotating the input image
Detection as Classification

- Face Detection reduced to Classification
  - At each image location we only have to make a decision whether the patch is a human face or not
- Pattern Recognition

Courtesy MIT CBCL
Detection as Classification

• Machine Learning
  – No hand-coded human knowledge
  – Difference between faces and non-faces are automatically learned from training data
  – Human knowledge is passed to machine in the form of labeled training data
Pattern Recognition Perspective

• Consider a thumbnail
  - 19×19 pattern, 256 grayscale levels
• $256^{361}$ possible combination!
  - $256^{361} = 2^{2888}$
  - Total world population (as of 2004) = 6.4 billion ≈ $2^{32}$
• Extremely huge space!
Classifiers

- Naïve Bayes
- Neural Network: Multilayer Perceptrons
- Principal Component Analysis (PCA), Factor Analysis
- Mixture of PCA, Mixture of factor analyzers
- Hidden Markov Model
- Sparse Network of Winnows (SNoW)
- Kullback relative information
- Inductive learning: C4.5
- Support Vector Machine (SVM)
- AdaBoost
Appearance Based Approach

• Pros:
  – Employing solid machine learning algorithms
  – Demonstrated good empirical results
  – Fairly robust

• Cons
  – Exhaustive search over space and scale
  – Needs lots of positive and negative examples
  – Capability relies on training data
    • Pose (out-of-plane rotation)
Real-time Face Detection Using AdaBoost

• AdaBoost
  – One of the most successful algorithms in machine learning
    • Best off-the-shelf classifier (L. Breiman)
  – Simple, but Powerful; Easy, but Deep

• Key Idea: Boosting
  – For a complicated classification problem, we don’t know how to directly design a good (strong) classifier
  – Instead, we design a series of simple (weak) ones, that are complement to each other
  – Finally, we combine all the weak classifiers into a strong one (vote)
Boosting (Schapire 1989)

• Randomly select $n_1 < n$ samples from $D$ without replacement to obtain $D_1$
  - Train weak learner $C_1$

• Select $n_2 < n$ samples from $D$ with half of the samples misclassified by $C_1$ to obtain $D_2$
  - Train weak learner $C_2$

• Select all samples from $D$ that $C_1$ and $C_2$ disagree on
  - Train weak learner $C_3$

• Final classifier is vote of weak learners
AdaBoost - Adaptive Boosting

- Instead of sampling, re-weight
- Learn weak classifiers based on the weighted training sample set
- Final classification based on weighted vote of weak classifiers
AdaBoost - Algorithm

Given \((x_1, y_1, \ldots, x_m, y_m)\) where \(x_i \in X, y_i \in Y = \{-1, +1\}\)

Initialize \(D_1(i) = 1/m\)

For \(t = 1..T\)

Select the best weak classifier using distribution \(D_t\)

Get weak hypothesis \(h_t : X \rightarrow \{-1, +1\}\) with error \(\varepsilon_t = \Pr_{x \sim D_t}[h_t(x) \neq y_i]\)

Choose \(\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right)\)

Update \(D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}\)

where \(Z_t\) is a normalization factor

Output the final hypothesis:

\[ H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]
Initial

$D_1$
Round 1
Round 2

\[ \varepsilon_1 = 0.30 \]
\[ \alpha_1 = 0.42 \]

\[ \varepsilon_2 = 0.21 \]
\[ \alpha_2 = 0.65 \]
Final classifier

\[
\text{sign}(0.42) + 0.65 + 0.92
\]
AdaBoost - Algorithm, Again

Given \((x_1, y_1), \ldots, (x_m, y_m)\) where \(x_i \in X, y_i \in Y = \{-1,+1\}\)

Initialize \(D_1(i) = 1/m\)

For \(t = 1..T\)

Select the best weak classifier using distribution \(D_t\)

Get weak hypothesis \(h_t : X \to \{-1,+1\}\) with error \(\varepsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]\)

Choose \(\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \varepsilon_t}{\varepsilon_t} \right)\)

Update \(D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases}\)

where \(Z_t\) is a normalization factor

Output the final hypothesis:

\[ H(x) = \text{sign} \left( \sum_{t=1}^{T} \alpha_t h_t(x) \right) \]
AdaBoost for Face Detection

How to find weak classifiers?

• 1-D: Easy, just thresholding
• But, recall that our pattern to be classified is $19 \times 19$
  - Reduce it to 1-D
  - Summarize the 361-vector into one real number: The Feature
• What are good features?
  - Simple and fast to compute
  - Reasonably discriminative
  • Stump? No, too weak...

www.beckman.uiuc.edu
Viola-Jones Detector
(P. Viola and M. Jones, 2001)

Two-rectangle

Three-rectangle

Four-rectangle

Feature Value =
\[ \sum \text{(pixels in white area)} - \sum \text{(pixels in black area)} \]
Efficient Computation: Integral image

- To compute Viola-Jones (Haar Wavelet) features, we only need to compute pixel integration over rectangular regions.

- Can be done very efficiently:
  - 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.

\[
\sum_{x' \leq x, y' \leq y} i(x', y')
\]

\((x, y)\)

www.beckman.illinois.edu
Constructing Integral image

- Computed in one pass over the original image.

\[ ii(-1,*) = ii(*,-1) = 0 \]
\[ ii(x,y) = i(x,y) + ii(x-1,y) + ii(x,y-1) - ii(x-1,y-1) \]

- Only needs to be done once for the whole input image.

\[ ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x', y') \]
Efficient Computation

With the help of the integral image, now we can compute the value of any rectangular sum in constant time.

Example:
The integral sum inside rectangle D we can compute as:

$$\ii(4) + \ii(1) - \ii(2) - \ii(3)$$
V-J Detector: Cascade Structure

All Sub-windows

1 \[\rightarrow T \rightarrow 2 \rightarrow T \rightarrow 3 \rightarrow T\]

Further Processing

Reject Sub-window
V-J Detector: Cascade Structure

- We start with simple classifiers which reject many of the negative sub-windows while detecting almost all positive sub-windows.
- Positive results from the first classifier triggers the evaluation of a second (more complex) classifier, and so on.
- A negative outcome at any point leads to the immediate rejection of the sub-window.

*Why bother?*

- For speed-up!
  - Sub-windows with no hope to be faces can be rejected as soon as possible, avoiding further unnecessary processing.
Stages in the cascade

- First stage: Two most important features
  - Miss: 0%, False alarm: 50%

- Deeper stages use more features, which achieve higher detection rate and lower false positive rate, but take more time.

Figure courtesy of Viola & Jones, IJCV2004
Viola-Jones Face Detector

- **Real-time**
  - On a 700MHz Pentium III processor, the face detector can process a 384 by 288 pixel image in about .067 seconds (30 fps)

- **Try it yourself!**
  - A good implementation in the OpenCV library ([http://sourceforge.net/projects/opencvlibrary/](http://sourceforge.net/projects/opencvlibrary/))
Multiview detection

- **Pose adaptive** (V&J01, Rowley 98, Schneiderman 04)
  - Pose estimator as a first stage
  - Train different classifier for each pose class
  - On a non face, the pose can be considered random

- **Universal approach**
  - Train on all cases of pose
  - Use the same detector for all input

- **3D model**
  - Difficulty:
    - Computational cost
    - Lack of training data
Conclusion

• AdaBoost
  – Building a strong classifier out of a series of weak classifiers
  – Re-weighting training samples, focusing on samples mis-classified by previous weak classifiers
  – One of the most influential achievements in machine learning

• Viola-Jones Face Detector
  – AdaBoost + Haar Wavelet Features
  – Accurate and fast
  – One of the most important results in computer vision. Inspired lots of work in face processing (detection, recognition, etc.)
References

- P. Viola and M. Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features”, *CVPR*, 2001
- S. Z. Li and Z. Zhang. “FloatBoost Learning and Statistical Face Detection”, *TPAMI*, v29 n9, 2004
Credits

- This presentation is partly based on slides authored by following people:
  - Dr. Ming-Hsuan Yang
  - Dr. Derek Hoeim
  - Dr. Stan Z. Li
  - Ce Liu
  - Ariel Parnes