Face Authentication using Pixel-based Weak Classifiers

Sébastien Marcel and Yann Rodriguez
marcel@idiap.ch

IDIAP Research Institute
Martigny, Switzerland

http://www.idiap.ch
Outline

• Introduction
• State-of-the art Approaches
• The Proposed Approach
• Database and Experimental Protocol
• Experimental Results
• Conclusion
Introduction

• Identity Verification/Authentication has many real-life applications:
  – access control,
  – transaction authentication (in telephone banking or remote credit card purchases for instance),
  – voice mail,
  – secure teleworking.

• A face authentication system requires to build a model for each client,

• Two kinds of events. The person claiming a given identity:
  – is the one who he claims to be (client),
  – is not the one who he claims to be (impostor),

→ thus it takes two decisions: either accept the client or reject him and decides he is an impostor.
State-of-the-art Approaches

- **Main approaches:**
  - Global (holistic): process the input face image as a whole,
  - Local: process the input face locally.

- **Face representation/Features:**
  - raw pixels, PCA, LDA,
  - 2D-DCT, wavelets, ...

- **Classification:**
  - Normalized Correlation ([Li, Kittler and Matas, 2000](#)),
    MLP ([Marcel and Bengio, 2002](#)), SVM ([Kostin et al., 2002](#)),
  - GMM ([Cardinaux, Sanderson and Marcel, 2003](#)),
    HMM ([Cardinaux, Sanderson and Bengio, 2004](#)), ...
Outline

- Introduction
- State-of-the art Approaches
- The Proposed Approach
- Database and Experimental Protocol
- Experimental Results
- Conclusion
The Proposed Approach

- Boosting algorithms (AdaBoost, FloatBoost, ...):
  - boost the performance of simple (weak) classifiers by combining them iteratively,
  - proposed for object detection ([P. Viola and M. Jones, 2001]) and applied successfully to face detection ([V. Pavlovic and A. Garg, 2001]),
  - boosted classifiers involves hundreds parameters while standard approaches (MLP, SVM, ...) involves thousands of parameters.

- Boosting pixel-based weak classifiers using AdaBoost:
  - Features: the pre-processed raw image,
  - Classification: “weak” classifiers using pixel values.
Face Extraction and Pre-Processing

- Face modeling: computes face bbx using eyes coordinates

- Pre-processing
  - histogram equalization,
  - smoothing using Gaussian convolution (3x3, $\sigma = 0.25$).
Boosting pixel-based weak classifiers

- Weak classifier (simplification):

\[ h_t(x) = \begin{cases} 
  +1 & : \ x_{f_t} \leq \theta_t \\
  -1 & : \ x_{f_t} > \theta_t
\end{cases} \]

- \( x \) is the given input image,
- \( f_t \) is the index of a pixel in \( x \),
- \( \theta_t \) is a threshold.

- AdaBoost estimates iteratively the best feature \( \{f_t, \theta_t\} \) for \( 1 \leq t \leq 300 \)

- Final strong classifier: \( f(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \)
Outline

- Introduction
- State-of-the-art Approaches
- The Proposed Approach
- Database and Experimental Protocol
- Experimental Results
- Conclusion
Database and Experimental Protocol

- XM2VTS database:
  - 295 subjects,
  - 4 sessions (one month interval),
  - 2 shots per session.

- Experimental Protocol: Lausanne Protocol ([Lüttn and Maître, 1998](#)),
  - 200 clients, 25 development (\(D\)) impostors, 70 evaluation (\(E\)) impostors,
  - training client accesses: 3,
  - development client accesses: 600,
  - development impostor accesses: 40,000 (25 \(\times\) 8 \(\times\) 200),
  - evaluation client accesses: 400 (200 \(\times\) 2),
  - evaluation impostor accesses: 112,000 (70 \(\times\) 8 \(\times\) 200).
Performance Measure

- False Rejection ($FRR$): when the system rejects a client,
- False Acceptance ($FAR$): when the system accepts an impostor,

- the decision threshold $\Theta$ chosen is the one that reaches the Equal Error Rate (when $FRR = FAR$), on a given data set:
  - *a posteriori threshold* ($\Theta^*$) optimized on the evaluation set ($\mathcal{E}$) and applied on ($\mathcal{E}$),
  - *a priori threshold* ($\hat{\Theta}$) optimized on the development set ($\mathcal{D}$) and applied on ($\mathcal{E}$).
- An unique measure is used to combine $FRR$ and $FAR$: Half Total Error Rate ($HTER = (FRR + FAR)/2$).
Outline

- Introduction
- State-of-the art Approaches
- The Proposed Approach
- Database and Experimental Protocol
- Experimental Results
- Conclusion
Experimental Results

- all experiments performed using Torch (www.torch.ch),

<table>
<thead>
<tr>
<th>Model</th>
<th>FAR</th>
<th>FRR</th>
<th>HTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>NC [Li, Kittler and Matas, 2000]</td>
<td>3.46</td>
<td>2.75</td>
<td>3.1</td>
</tr>
<tr>
<td>AdaPix 50</td>
<td>3.34</td>
<td>4.0</td>
<td>3.67</td>
</tr>
<tr>
<td>AdaPix 100</td>
<td>3.16</td>
<td>3.5</td>
<td>3.33</td>
</tr>
<tr>
<td>AdaPix 150</td>
<td>3.11</td>
<td>3.5</td>
<td>3.3</td>
</tr>
<tr>
<td>AdaPix 200</td>
<td>2.75</td>
<td>3.0</td>
<td>2.87</td>
</tr>
<tr>
<td>AdaHaar3 100</td>
<td>2.29</td>
<td>5.0</td>
<td>3.64</td>
</tr>
</tbody>
</table>

- the performance of AdaPix increases with the number of classifiers,
- AdaPix is comparable to NC,
- Surprisingly, AdaPix outperforms AdaHaar7 with much less classifiers.
Conclusion

• Discussion:
  – the proposed boosting approach reaches average performances (in controlled conditions),
  – performances might be degraded in uncontrolled conditions.

• Future work:
  – perform additional experiments on a more difficult database (various recording conditions),
  – boosting other features (PCA, LDA and Gabor wavelets).
Torch3 vision

- features available:
  - image I/O: reads/writes basic image formats (pgm, ppm, gif, tif, jpeg)
  - image processing: edge detection, photometric normalisations, rotation, flip, ...
  - feature extraction: histograms, PCA, LDA, 2D-DCT, ...
  - geometry: to create, manipulate and draw 2D objects
  - video I/O: for decoding and encoding video files (avi, mpeg1/2).

- packages under development:
  - stump classifiers for boosting algorithms,
  - face detection and verification tools and algorithms,
  - connected components and morphological operators,
  - video acquisition (frame grabber, USB, DV).

- more details at http://www.idiap.ch/~marcel section Programming/Torch3vision
References


