An Open Source Framework

for Standardized Comparisons of Face Recognition Algorithms

Manuel Günther, Roy Wallace, Sébastien Marcel

Idiap Research Institute
CH - 1920 Martigny

October 13th 2012

Outline

Introduction

What researchers want to have

1. Interesting paper
2. Source code from author
3. Implement own ideas
4. Re-run algorithm — same database
   - default protocol

   $\Rightarrow$ results directly comparable
5. Results are better
6. Publish paper $\Rightarrow$ accepted
7. Publish source code for other researchers

Introduction

What happens instead

1. Interesting paper
2. no source code from author $\rightarrow$ code yourself
   - missing parameters, bugs
3. Implement own ideas
4. Re-run algorithm — same database
   - no default protocol $\rightarrow$ implement own protocol

   $\Rightarrow$ results incomparable
5. Publish paper $\Rightarrow$ accepted if you are lucky
6. Not publishing source code

Introduction

Question

Is the modification really better than the original algorithm?

Answer

No-one can tell!

Ranking of algorithms

Surveys

- Unable to reproduce results
- Report results of published papers
- *it is really difficult to define a "winner" algorithm[1]*
- *different papers may use different parts of the databases for their experiments [2]*

Face Recognition Vendor Tests

- Focused on **one** database
- Closed source $\rightarrow$ not reproducible

FaceRecLib

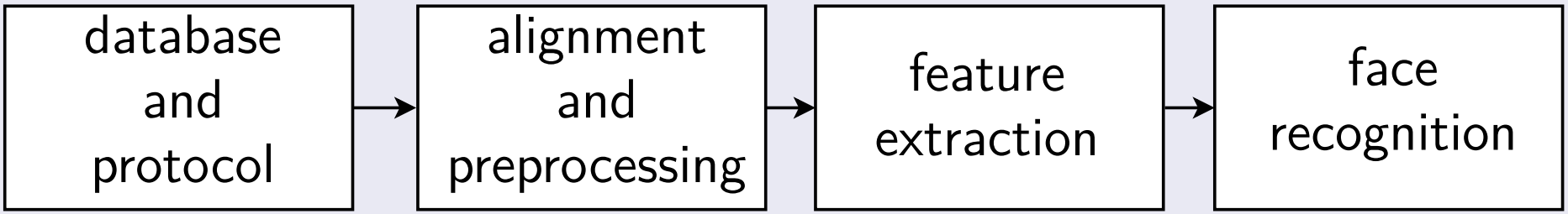
Capabilities

- Open Source
- Fixed evaluation protocols
- Defined meta-parameters
- Reproducible results

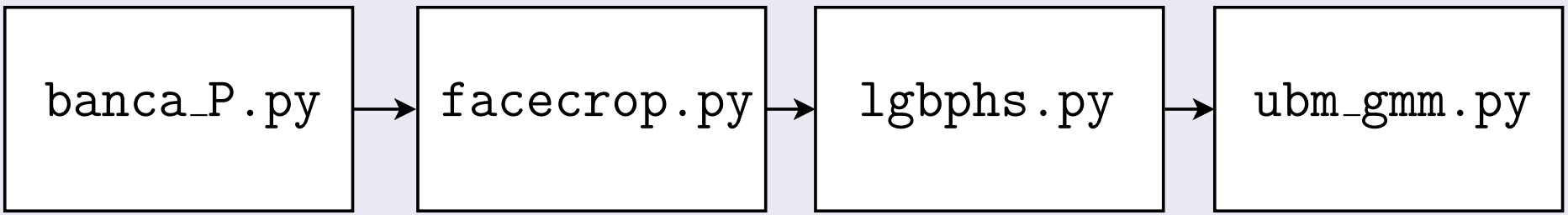
- Many image databases
- Variety of face recognition algorithms
- Extensible
- Rapid prototyping

FaceRecLib

Face recognition tool chain



Configuration files for each step



Running face recognition experiments

```
$ faceverify.py -d banca_P.py -p facecrop.py -f lgbphs.py -t ubm_gmm.py
```

Step one – database

Database

Original image

+

Annotations

+

Protocols

⇓

banca_P.py

```
import xbob.db.banca

# Define the database
name = 'banca'
database = xbob.db.banca.Database()

# Specify the protocol
protocol = 'P'

# Set the paths to the data
image_directory = "/idiap/.../images_gray/"
image_extension = ".ppm"
annotation_directory = "/idiap/.../eyecenter/"
annotation_type = 'eyecenter'
```

Step two – preprocessing

Preprocessing

Original image

+

Annotations

Aligned image

facecrop.py

```
import facereclib

# Declare the preprocessor to be used
preprocessor = facereclib.preprocessing.FaceCrop

# Size of the cropped image
CROPPED_IMAGE_HEIGHT = 80
CROPPED_IMAGE_WIDTH  = 64

# Eye positions in the cropped images
RIGHT_EYE_POS = (16, 15)
LEFT_EYE_POS  = (16, 48)
```

Step three – feature extraction

Feature extraction

Aligned image

Extracted features

lgbphs.py

```
import facereclib
import math

# feature extraction
feature_extractor = facereclib.features.LGBPHS

# Block setup
BLOCK_HEIGHT = 10
BLOCK_WIDTH = 10
BLOCK_Y_OVERLAP = 4
BLOCK_X_OVERLAP = 4

# LBP parameters
RADIUS = 2
NEIGHBOR_COUNT = 8
IS_UNIFORM = True
IS_CIRCULAR = True
IS_ROTATION_INVARIANT = False

# Gabor parameters
GABOR_DIRECTIONS = 8
GABOR_SCALES = 5
GABOR_SIGMA = math.sqrt(2.) * math.pi
GABOR_MAXIMUM_FREQUENCY = math.pi / 2.
GABOR_FREQUENCY_STEP = math.sqrt(.5)
```

Step four – face recognition

Face recognition

Extracted features

+

Protocol

Model enrollment

Model – probe – scores

ubm_gmm.py

```
import facereclib
import bob

tool = facereclib.tools.UBMGMMTool

# GMM Training
GAUSSIANS = 512
K_MEANS_TRAINING_ITERATIONS = 500
GMM_TRAINING_ITERATIONS = 500
GMM_TRAINING_THRESHOLD = 0.0005
GMM_VARIANCE_THRESHOLD = 0.0005
UPDATE_WEIGTHS = True
UPDATE_MEANS = True
UPDATE_VARIANCES = True
NORMALIZE_BEFORE_K_MEANS = True


# GMM Enrollment and scoring
RELEVANCE_FACTOR = 4
GMM_ENROLL_ITERATIONS = 1
RESPONSIBILITY_THRESHOLD = 0

scoring_function = bob.machine.linear_scoring
```

Implemented Interfaces

Databases

- ARface
- AT&T
- BANCA
- CAS-PEAL
- FRGC
- GBU
- LFW
- Mobio
- Multi-PIE
- SCface
- XM2VTS

Preprocessors

- Face cropping
- Hist. Equal.
- Self Quotient
- Tan & Triggs
- I-Norm-LBP

Features

- Pixels
- DCT blocks
- LGBPHS
- Gabor graphs
- SIFT

Algorithms

- PCA
- PCA+LDA
- BIC
- Histogram intersection
- Gabor jet similarities
- UBM/GMM
- ISV
- PCA+PLDA
- LR-PCA
- LDA-IR

Bob

Signal processing and machine learning toolbox [3]

http://www.idiap.ch/software/bob

- Signal and image processing techniques
  - filtering, LBP, SIFT, optical flow etc.
- Machine learning algorithms
  - PCA, LDA, MLP, SVM, JFA, GMM, clustering etc.
- Image database support
- Satellite packages

https://github.com/idiap/bob/wiki/Satellite-Packages

Step five — evaluation

Evaluation

⇓

Score file(s)

↓

ROC curves

+

EER and HTER

```
probe id, model id, probe file, score
103  103  m103/m103_02_f12_i0_0  4904.21515413
103  103  m103/m103_02_f13_i0_0  6041.20061168
103  103  m103/m103_02_f14_i0_0  6457.26529403
103  103  m103/m103_02_f15_i0_0  5726.05947192
...
104  103  m104/m104_04_f18_i0_0  7.02726051809
104  103  m104/m104_04_f19_i0_0  193.676140904
104  103  m104/m104_04_f20_i0_0  -445.768318634
104  103  m104/m104_04_f21_i0_0  213.431047733
...
103  108  m103/m103_02_f16_i0_0  -1115.46444995
103  108  m103/m103_02_f17_i0_0  -1621.60598761
103  108  m103/m103_02_f18_i0_0  -1807.30024395
103  108  m103/m103_02_f19_i0_0  -1429.40971486
...
108  108  m108/m108_04_f12_i0_0  2037.48075016
108  108  m108/m108_04_f13_i0_0  2022.42360897
108  108  m108/m108_04_f14_i0_0  1949.7535052
108  108  m108/m108_04_f15_i0_0  2463.02478421
...
```

Example runs of the FaceRecLib

State-of-the-art algorithms

1. Tan & Triggs + LGBPHS + χ^2
2. Tan & Triggs + Gabor graph + S_{n+C}
3. Tan & Triggs + DCT blocks + ISV
4. LDA-IR from PythonFaceEvaluation
   $\rightarrow$ Colorado State University (CSU) [4]

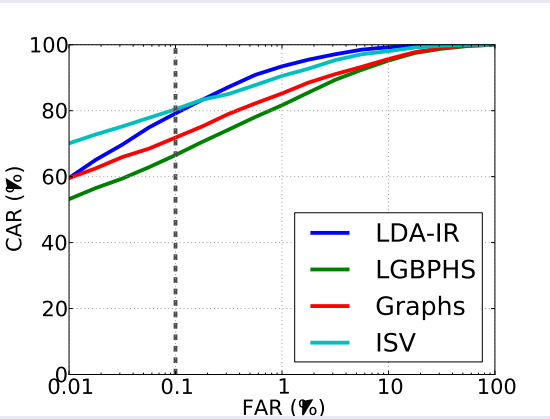
Image Databases

1. GBU with default protocols
2. BANCA with protocol P

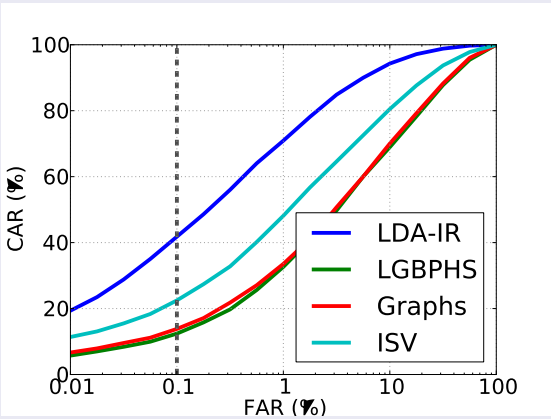
Results

ROC on GBU

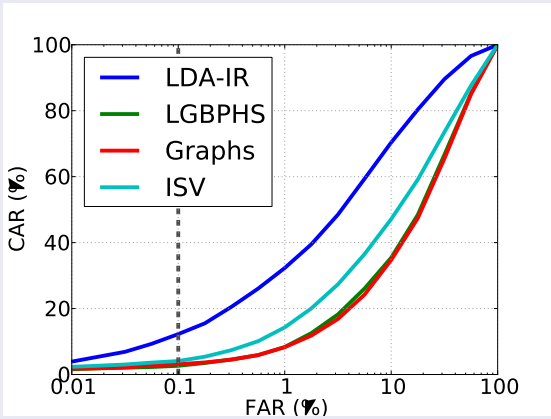
Good



Bad



Ugly



HTER on BANCA

	LDA-IR	**LGBPHS**	**Graphs**	**ISV**
$\mathrm{HTER}_{\mathrm{test}}$	27.2%	16.1%	12.4%	10.9%

Conclusion

First face recognition tool **ever** that

- Is open source (soon) [5]
- Generates reproducible results
- Includes many image databases and protocols
- Includes many state-of-the-art algorithms
- Is easily extensible
- Is easily configurable
- Is well documented
- Is the perfect play-ground for researchers

Outlook

Other experiments

- Face identification
- Facial video recognition
- Speaker verification

More features and algorithms

- SVM, Kernel-SVM, ...
- Nullspace LDA, ...
- your algorithm

Thank you!

References:

X. Tan, S. Chen, and Z.Z.F. Zhang.
Face recognition from a single image per person: A survey.
Pattern Recognition, 39:1725–1745, 2006.

L. Shen and L. Bai.
A review on Gabor wavelets for face recognition.
Pattern Analysis and Applications, 9(2):273–292, September 2006.

A. Anjos, L. El Shafey, R. Wallace, M. Günther, C. McCool, and S. Marcel.
Bob: a free signal processing and machine learning toolbox for researchers.
In *20th ACM Conference on Multimedia Systems*. ACM Press, 2012.
http://www.idiap.ch/software/bob.

Ross Beveridge and D.S. Bolme.
CSU Face Recognition Resources.
http://www.cs.colostate.edu/facerec/algorithms/baselines2011.php, 2011.

M. Günther, L. El Shafey, R. Wallace, S. Marcel, et al.
The FaceRecLib: Standardized comparisons of face recognition algorithms.
https://www.github.com/bioidiap/facereclib.