2D FACE RECOGNITION: AN EXPERIMENTAL AND REPRODUCIBLE RESEARCH SURVEY

Manuel Günther Laurent El Shafey
Sébastien Marcel

Idiap-RR-13-2017

APRIL 2017

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

2D Face Recognition: An Experimental and Reproducible Research Survey

Manuel Günther, Laurent el Shafey and Sébsatien Marcel
Idiap Research Institute, Switzerland

March 31, 2017

Abstract

Due to its wide range of applications, automatic face recognition is a research area with high popularity. Many different face recognition algorithms have been proposed in the last decades. Nearly every day there is a new face recognition paper sent to a conference or a journal. Often, researchers provide results that rely on a hand-made non-standard evaluation protocol and that are, hence, incomparable to state-of-the-art algorithms. Additionally, the source code for the algorithms is often not provided by the researchers. In consequence, face recognition survey papers can only report the results of other papers.

In this paper we provide to our best knowledge the first experimental and evaluative study of a variety of state-of-the-art face recognition algorithms that solely relies on open source software, including color-based linear discriminant analysis, local Gabor binary pattern histogram sequence, Gabor graphs using a Gabor-phase based similarity measure and inter-session variability modeling. Together with this paper we supply the source code to re-run all the experiments that we execute in this study. Experiments are performed on many freely available image databases, always following the evaluation protocols that are attached to them. First, we optimize the parameters of all tested algorithms on a single database. This includes finding the best image preprocessing for each algorithm. Then, we test the algorithms against facial variations as expressions, pose and occlusions using the Multi-PIE and the AR face database. Finally, we report the results of these algorithms on CAS-PEAL, MOBIO, SC face, GBU, FRGC and LFW and discuss some other properties of the algorithms.

The results show several trends, partially supporting and partially contradicting prevailing beliefs of the face recognition society. First, Gabor wavelet based algorithms perform better than algorithms relying on raw pixel values, and incorporating Gabor phases improves performance; second, color is an important cue for face recognition; third, the inter-session variability modeling algorithm can handle variations in facial expression and partial occlusions best; fourth, if more than one image is provided at enrollment or probe time, algorithms increase performance; and fifth, biased evaluation protocols as in FRGC or CAS-PEAL favor algorithms that make use of identity information at training time, such as linear discriminant analysis and inter-session variability modeling.

1 Introduction

After the first automatic face recognition algorithms [47, 87] appeared more than three decades ago, this area has attracted many researchers and there has been a lot of progress in this field. One of the reasons of its popularity is the broad field of applications of (automatic) face recognition. Mainly, these applications can be grouped into three different categories, where each category has its own characteristics:

First, there is the classic access control scenario, where a *client* wants to access a secured area. Usually, the client has an access card that stores biometric information — a so-called *client model* — about the client. In front of the secured area, an image — a *probe* — of the client is taken and compared with the model on the access card. If the similarity of the two is higher than a certain *threshold*, access is granted. The characteristics of this scenario is that the conditions in taking model and probe images can be controlled and the clients are willing to cooperate with the system, but an *impostor*, who might have stolen the access card, should not be allowed to access the secured area under any circumstance.

Surveillance based applications form the second category. *Closed circuit television* (CCTV) cameras are installed in public places and monitor whether one of a set of interesting persons pass through the eye of the camera. Since more and more CCTV cameras are installed and the number of human observers is limited, there is the need to automate this process. The issues arising in this scenario are manifold. Neither model nor probe images have been taken under controlled conditions and, thus, illumination, pose, facial expression and facial occlusions might differ. One sub-category of surveillance is forensics, where (facial) traces from a crime scene should be compared with the face of a suspect. In this case it is important to compute the probabilities of both hypothesis: the suspect committed the crime, or someone else did it.

A third use case, which is recently gained popularity, is the commercial use of face recognition algorithms. Nowadays, users take huge amounts of photos with digital cameras. To help the user to organize these images, they can be automatically tagged with the names of the persons they contain, e. g., by commercial applications like *Picasa* or *iPhoto*. The particularity of these applications is that the user usually knows the persons in the images and is able to correct mislabeled images. Thus, automatic face recognition algorithms should be able to incorporate these information and do a life-long learning.

1.1 Face recognition algorithms

Commonly, the face recognition task is composed of several stages. The first stage is face detection, in which location and scale of the face(s) in the image is estimated. Some face detection algorithms also account for in-plane rotated faces [79, 35], while most popular algorithms only detect upright faces [61, 94, 107]. Using these information, the image is geometrically regularized to a fixed image resolution. Since the main focus of this survey is face recognition rather than face detection, we rely on image regularization according to hand-annotated eye positions, keeping in mind that this simplifies the face recognition task.

The regularized face images are then subjected to a photometric enhancement step, which mainly reduces effects of illumination conditions. In the simplest case, raw pixel values are used directly, while more complex algorithms, e.g., perform histogram equalization [76], compute a self quotient image [98], execute a multi-step enhancement [90], or extract *local binary patterns* (LBP) [38]. In this paper the two stages, image regularization and photometric enhancement, are subsumed to the *preprocessing* step.

In the second step, image features that contain relevant information needed for face recognition are extracted. Again, the simplest "feature" is a concatenation of all pixels of the preprocessed image in a regular order [92]. Well-known face recognition algorithms rely on *local binary patterns* (LBP) [2]. Several other features rely on first applying a Gabor wavelet transform [99] with a discrete set of Gabor wavelets, and extracting Gabor graphs [99, 33] or *local Gabor binary patterns* (LGBP). For some features, the image is decomposed into several – possible overlapping – blocks and features like *discrete cosine transform* (DCT) features [82] or *LGBP histogram sequences* (LGBPHS) [108, 106] are extracted from the blocks. Finally, *scale invariant feature transform* (SIFT) features are also used in some applications [6, 27].

Based on these extracted features different face recognition algorithms have been engineered during the last decades. There are two major approaches to automatic face recognition. The first is the *discriminative* approach, which tries to classify whether features of model and probe belong to the same identity or not. Some examples of this approach that rely on raw image pixels are *eigenfaces* (PCA) [92], *Fisher faces* (LDA) [22, 111, 103] and the *Bayesian intrapersonal/extrapersonal classifier* (BIC) [62]. Also combinations of, e. g., Gabor wavelet transform features with LDA [25] or LGBP's with PCA [64] belong to this approach. All these algorithms project the extracted features into a subspace and compare features with a simple distance metric. But there are other discriminative algorithms. For example, Gabor graph based algorithms usually compare Gabor graphs with identical topology using specialized Gabor jet similarity functions [99, 46, 33] to define their similarity, while LGBPHS features are compared using histogram similarity measures [108].

The second major approach to automatic face recognition is the *generative* approach. Here, the idea is to compute the probability that a given client could have produced the probe sample. Prominent representative algorithms are the *unified background model* (UBM) – *Gaussian mixture model* (GMM) modeling of DCT block features [97] and its extension to the *inter-session variability* (ISV) modeling [96]. Also the *probabilistic LDA* (PLDA) [75, 21] belongs to the class of generative algorithms.

1.2 Databases

To evaluate face recognition algorithms, there are several publicly accessible databases of facial images. The number of identities and images in these databases vary from 400 images of 40 persons in the *AT & T database of faces* [81] to the extremes of over 750000 images of 337 identities in the *Multi-PIE* database [29] or more than 13000 images of 5749 people in *labeled faces in the wild* (LFW) database [41]. Commonly, there is only one face present in each image of the database, and often additional information about the images are provided, like the gender of the person, the facial expression in the image or the environment conditions the image was taken in. In nearly every database at least the locations of the left and right eye are annotated by hand and sometimes there are also more annotations, e. g., for mouth and nose.

Depending on the intended task, the databases contain images taken under different environment conditions. Most databases include only images that are taken in strictly frontal pose, so that the effects of facial expressions, strong illuminations, partial occlusions, or human aging processes can be studied, i. e., the kind of variations that occurs in an access control scenario. Other databases like *mobile biometry* (MOBIO) [56] and LFW provide images in a completely unrestricted environment, corresponding to the private use case listed above. Unfortunately, there is only a very limited number of face databases[1] with images from the surveillance scenario and there is no publicly available forensic database.

1.3 Evaluation protocols

To ensure a fair comparison of face recognition algorithms, often image databases are accompanied with evaluation protocols. Most of these protocols define, which images of the database should be used for *training* the algorithm, and which images should be used to *evaluate* it. The evaluation set is divided into images used to *enroll* the client models, and images used to probe the system. Additionally to the evaluation set, some protocols define a similar *development* set, which should be used to optimize the algorithm configuration (cf. sec. 2) of the algorithms.

Evaluation protocols can be grouped into *biased* and *unbiased* protocols. While in an unbiased protocol, the subjects in the database are strictly separated between training, development and evaluation set, i. e., persons from development or evaluation set are not included in the training set, biased protocols allow the identities of these three sets to overlap. In some biased protocols [25] the training set even consists of the same images as the evaluation set.

In principle, the enrollment of a client model can integrate features of several images of a person. Most discriminative algorithms do not define a strategy to handle multiple images per client model, while generative face recognition algorithms make use of this fact in a principled way. However, in many protocols models are enrolled from a single image only, e. g., in access control or surveillance scenarios usually only a single mugshot image is available at enrollment time.

Evaluation protocols can further be subdivided into *identification* and *verification* protocols, which define different evaluation measures. While for identification the most similar models for a given probe are found and *ranked*, verification results in a binary yes/no decision for given pair of model and probe. Sec. 3.1 provides a more detailed description of the evaluation measures that we use in our experiments.

1.4 Algorithm evaluation

Without considering the comparability of their results, many researchers in face recognition base their experiments on small image databases like the AT & T

[1] For the time being we only know of the *surveillance camera* (SC) face database [28] and the *ChokePoint* database [101].

database of faces or other databases that have no attached evaluation protocol. Even if protocols are available, sometimes researchers run their experiments on protocols — in most cases biased ones — they have designed themselves. Unfortunately, this makes their results **incomparable** to the results of other researchers, which might even have chosen the same image database, but a different protocol. Additionally, often results can not be reproduced since the authors do not publish source code, and papers do not contain entire algorithm configurations. Thus, existing face recognition surveys like [110, 89, 84, 1, 44, 83, 39] can only provide the figures that are reported by other researchers, so "*it is really difficult to declare a winner algorithm*" (as stated in [89]) since "*different papers may use different parts of the database for their experiments*" (as written in [84]). In an attempt to categorize face recognition algorithms, [83] used a more advanced evaluation of the methods, but still they had to rely on the results published by the authors of the surveyed papers.

Some institutions already tried to provide an open source interface, in which different algorithms could be tested. For example, the *CSU Face Identification Evaluation System* [12] implemented and tested some algorithms on the FERET image database [73] using FERET protocols. Unfortunately, the interface was written in C++ and, thus, all algorithms needed to be re-implemented by hand. Also, algorithms were only tested on the FERET database using biased protocols.

More fair comparisons of algorithms are done by *face recognition vendor tests* (FRVT) [11, 72, 74] and similar tests, which are regularly held by the US *National Institute for Standards and Technology* (NIST). But since these tests are designed to confront commercial algorithms, the methodology of the participating programs are usually kept secret. Also, the databases and the protocols of these tests are not available and, hence, it is impossible to replicate the experiments.

1.5 Reproducible research

In this paper we provide the first **evaluative** study of a variety of different face recognition algorithms. We perform a **fair comparison** of all tested algorithms. We evaluate the algorithms on **several** publicly available image databases using their **fixed** evaluation protocols, and investigate the suitability of the algorithms under several image conditions. Additionally, we provide the **source code**[2] not only for the algorithms, but also for the **complete experiments** from the raw images to the final evaluation including the **figures and tables** that can be found in this paper.

Every experiment is solely based on open source software. Most of the algorithms that we run use *Bob* [7], which is a free signal processing and machine learning toolbox for researchers.[3] Some algorithms are taken from the *CSU Face Recognition Resources* (CSU),[4] which provide the baseline algorithms for *the good, the bad & the ugly* (GBU) challenge [69, 53]. Finally, all experiments are executed using the FaceRecLib [34],[5] which provides an easy interface to run

[2] The source code never got published. Please check its successor: http://pypi.python.org/pypi/bob.chapter.FRICE

[3] http://www.idiap.ch/software/bob

[4] http://www.cs.colostate.edu/facerec/algorithms/baselines2011.php

[5] http://pypi.python.org/pypi/facereclib

face recognition experiments either using already implemented face recognition algorithms, or rapidly prototyping novel ideas. In the FaceRecLib, interfaces for several image databases are provided and changing the algorithm or the database is as easy as changing a command line option.

We intentionally provide the source code that is generating the results and we want to encourage other researchers to publish their source code as well. One vast argument for sharing source code is the fact that published papers with associated source code on average gains 5 times more citations [93]. Together with Bob, the FaceRecLib and several other open source packages that we provide in the *python package index* (PyPI) we have set up a simple and effective way of sharing source code, allowing anyone to reproduce the results that are published.

The remaining of this paper is structured as follows: In sec. 2 we give an overview of the features and the algorithms that are used in this paper. Sec. 3 describes the image databases and the protocols that we consider. Sec. 4 contains the experimental results that we achieved, while secs. 5 and 6 close the paper with a detailed discussion of the tested face recognition algorithms and a final conclusion.

2 Face recognition algorithms

Though we present and test a variety of face recognition algorithms, there is a common execution order (a figure displaying the execution order is shown in the online appendix) to perform a face recognition experiment. Given a raw image from a certain image database, the first stage is to detect the face(s) in the image, remove the background information and geometrically normalize the face image. Throughout our experiments, we use the hand-labeled eye annotations provided by the databases to geometrically normalize the face by aligning the eye positions to certain positions in the image. The aligned image might be further processed by some preprocessing algorithm, usually to attenuate the effects of illumination.

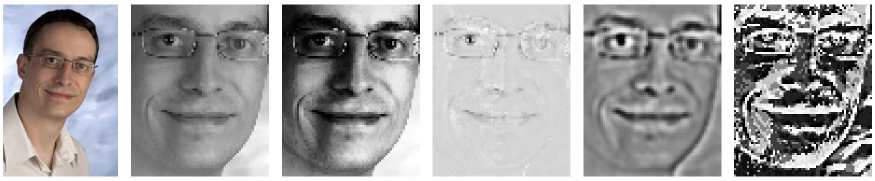
In the next step, features are extracted from the preprocessed images. Some of the feature extractors need to be adapted to the given image database. For this, the feature extraction potentially might have a training stage, which is using the preprocessed images from the training set of the database.

Many face recognition algorithms compare features in a certain *feature space*. This feature space is usually adapted to the database, to which the projector is trained using the extracted features from the training set. Afterward, all features are projected into feature space.

Based on the evaluation protocol, some (projected) features are used to enroll *models* $\mathcal{M}^{(c)}$ of several *clients* $c = 1, \ldots, C$. Each model $\mathcal{M}^{(c)}$ is enrolled using the features from Z_c enrollment images — in most cases models are enrolled using only $Z_c = 1$ image. Again, some face recognition algorithms need a training step to adapt the model enrollment to the database.

During face recognition, client models are compared with probe features, where the protocol defines, which probe feature is compared with which model. A *score* is assigned for each pair of model and probe.

Finally, the scores are used to compute an evaluation performance, which depends on the experimental setup defined by the protocol. Identification



(a) Orig **(b)** None **(c)** HE **(d)** SQ **(e)** T&T **(f)** LBP

Figure 1: Image preprocessing algorithms. *This figure shows the effect of different image preprocessing algorithms on the (a) original image: (b) no preprocessing, (c) histogram equalization, (d) self quotient image, (e) Tan & Triggs algorithm and (f) LBP feature extraction.*

performance is reported as a *recognition rate* (RR) or in *cumulative match characteristics* (CMC). Verification performance is measured by *correct acceptance rates* (CAR), *equal error rates* (EER) or *receiver operating characteristics* (ROC).

To improve recognition performance the raw scores can undergo a score normalization before the final performance measure is computed. It already has been shown [96] that score normalization can boost face verification performance of algorithms. However, in this paper we do not perform any kind of score normalization, but we leave the impact of score normalization on the various face recognition systems as an open question for further research.

2.1 Image preprocessing

2.1.1 Image alignment

Before a preprocessing algorithm is applied the image $\mathcal{I}$ is converted to gray scale and aligned by geometrically normalizing the image such that the left and right[6] eyes $\vec{a}_l^*$ and $\vec{a}_r^*$ are located at certain positions in the aligned image $\mathcal{I}^*$:

$$\mathcal{I}^*(\vec{x}) = \mathcal{I}\left(\frac{1}{s}\,Q_{-\alpha}\left(\vec{x} - \vec{o}^*\right) + \vec{o}\right) \tag{1}$$

where the scale s and the angle α are computed as:

$$s = \frac{\|\vec{a}_r^* - \vec{a}_l^*\|}{\|\vec{a}_r - \vec{a}_l\|} \qquad \alpha = \arctan\left(\frac{a_{r,y} - a_{l,y}}{a_{r,x} - a_{l,x}}\right) \qquad Q_\alpha = \begin{pmatrix}\cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha\end{pmatrix} \tag{2}$$

with $\vec{a}_l$ and $\vec{a}_r$ being the hand-labeled annotations of the left and right eye, $\vec{o}$ and $\vec{o}^*$ the transformation offsets in the original and the aligned image — usually, the center between the eyes is used in both cases — and Q_α the rotation matrix. After aligning the image to the eye positions, the image is cut to a specific *image resolution* $\mathcal{R} = (r_x, r_y)^\top$. Fig. 1(b) exemplary shows the result of the alignment of the image shown in fig. 1(a) to image resolution $\mathcal{R} = (64, 80)^\top$ with $\vec{a}_l = (48, 16)^\top$ and $\vec{a}_r = (15, 16)^\top$.

[6] Left and right are referred to from the perspective of the subject that is shown in the image.

2.1.2 Preprocessing algorithms

To reduce the impact of illumination in this work we test four different preprocessing algorithms, which are always executed on the aligned image $\mathcal{I}^*$:

Histogram equalization The first algorithm performs a *histogram equalization* (HE) [76]. According to the distribution of pixel gray values in the aligned image $\mathcal{I}^*$, the gray values are adapted:

$$\mathcal{I}_{\text{HE}}(\vec{x}) = \frac{\text{cdf}\left(\mathcal{I}^*(\vec{x})\right) - \text{cdf}_{\text{min}}}{R - \text{cdf}_{\text{min}}} \cdot 255 \tag{3}$$

using the *cumulative distribution function* (cdf) and the number cdf_{min} of pixels, which have the lowest pixel value in $\mathcal{I}^*$:

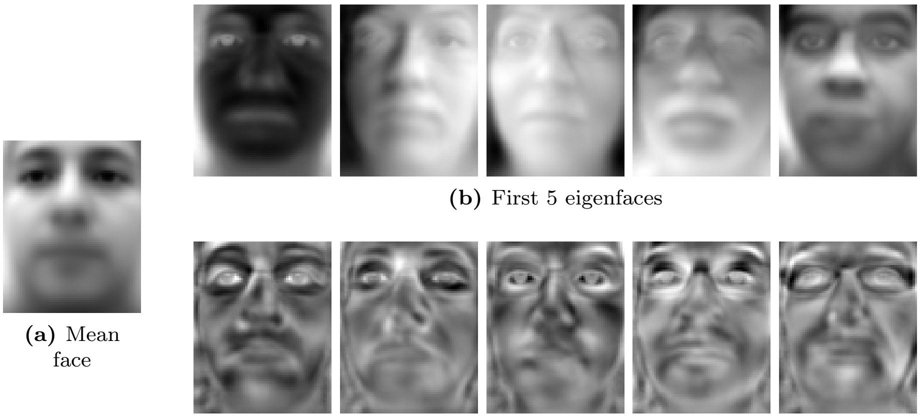
$$\text{cdf}(t) = \sum_{\vec{x}} |\{\mathcal{I}^*(\vec{x}) \leq t\}| \qquad\qquad \text{cdf}_{\text{min}} = \min_t\{\text{cdf}(t) \neq 0\} \tag{4}$$

where $R = r_x\, r_y$ is the number of pixels in $\mathcal{I}^*$. The result of the histogram equalization process of the image from fig. 1(b) is given in fig. 1(c).

Self quotient image The second preprocessing algorithm that we investigate is the *self quotient* (SQ) image. We here use SQ as introduced in [98], which differs from the one implemented in the CSU toolkit. The main idea is to divide the image $\mathcal{I}^*$ by a smoothed version of it. More precisely, a Gaussian-based anisotropic filter is employed that aims at removing low-frequency light effects like shadows while preserving regions with many edges. This is achieved by computing a distinct convolution kernel at each location of the image $\mathcal{I}^*$, based on pixel intensities in a close neighborhood. After computing the quotient between $\mathcal{I}^*$ and its smoothed version the logarithm function is applied pixel-wise to compress the dynamic range and, hence, reduce high-frequency noise. The resulting SQ filtered version of fig. 1(b) using a Gaussian standard deviation of 5 [98] can be seen in fig. 1(d).

Tan & Triggs' algorithm Third, we examine the multistage preprocessing algorithm as presented by [90]. It starts with *gamma correction*, performs *difference of Gaussian* filtering and, finally, applies *contrast equalization*. In this work we stick to the parameters of each step as the authors reported in [90]. An example of the preprocessed Tan & Triggs image can be obtained from fig. 1(e).

Preprocessing with local binary patterns Finally, we use a preprocessing algorithm [38] based on *local binary patterns* (LBP). To achieve photometric normalization 8-bit non-uniform LBP features with radius 2 [68] (see also fig. 4(b)) are extracted from the aligned images. To avoid border condition problems, the LBP extraction requires aligned images with a slightly higher resolution of 4 pixels in both direction, i. e., 2 times the LBP radius. This is implemented to assure that preprocessed images of all preprocessing algorithms have exactly the same image resolution. An exemplary image that is preprocessed using the LBP operator is given in fig. 1(f). Note that due to the nature of the LBP extraction, each bit of the LBP code is similarly important. This characteristic is consciously ignored by this preprocessing when reinterpreting LBP codes as pixel gray values. Interestingly, this does not seem to harm face recognition [38].



(a) Mean face

(b) First 5 eigenfaces

(c) First 5 Fisher faces

Figure 2: EIGEN- AND FISHER FACES. *This figure displays (a) the mean face and the first five (b) eigen- and (c) Fisher faces obtained from the training set of the MOBIO database.*

2.2 Eigenface based face recognition methods

2.2.1 Eigenfaces

Since [87] proposed the first feature-based automatic face recognition system in the late 1980s, eigenface based methods enjoy a great popularity because they are easy to understand and fast to implement. Traditionally, the methods directly act on the pixels of the preprocessed images by stringing together the pixels into a feature vector $\vec{v}$ of dimension R, i. e., the number of pixels in the image. Using the training features $\vec{v}^{(z)}, z = 1, \dots, Z$ from the training set of size Z, the mean $\vec{\mu}$ and covariance matrix Σ are computed:

$$\vec{\mu} = \frac{1}{Z} \sum_z \vec{v}^{(z)} \qquad\qquad \Sigma = \frac{1}{Z-1} \sum_z (\vec{v}^{(z)} - \vec{\mu})(\vec{v}^{(z)} - \vec{\mu})^\top \tag{5}$$

The covariance matrix is factorized into $\Sigma^{\text{-}1} = \Phi \Lambda^{\text{-}1} \Phi^\top$ using the well-known *Karhunen-Loève transform* (KLT), resulting in the orthonormal projection matrix Φ that defines the *face space.* Since the rows of Φ, which correspond to the eigenvectors of Σ, look like faces when they are reinterpreted as images, they are called *eigenfaces.* An exemplary mean face $\vec{\mu}$ and the five eigenfaces with the highest eigenvalues are shown in fig. 2(a) and fig. 2(b), respectively. Commonly, only the M eigenfaces with the highest eigenvalues are kept, while the others are regarded as noise. In this work, we chose to select the optimal M^* not fixed, but according to the percentage of eigenvalues λ_i that are needed to capture a certain amount of variation σ_{PCA} in the PCA subspace:

$$M^* = \arg\min_M \frac{\sum\limits_{i=1}^{M} \lambda_i}{\sum\limits_{i=1}^{R} \lambda_i} \geq \sigma_{\text{PCA}} \tag{6}$$

2.2.2 Linear discriminant analysis

A more task-oriented way of computing a projection matrix uses Fishers *linear discriminant analysis* (LDA). It does not only estimate how facial images are distributed, but also which image variations are *intrapersonal* and which are *extrapersonal.* Therefore, the *within class* and *between class scatter matrices* $\mathcal{S}_w$ and $\mathcal{S}_b$ are analyzed:

$$\mathcal{S}_w = \sum_c \frac{Z_c}{Z} \Sigma_c \qquad\qquad \mathcal{S}_b = \sum_c \frac{Z_c}{Z} (\vec{\mu} - \vec{\mu}_c)(\vec{\mu} - \vec{\mu}_c)^\top \tag{7}$$

where c iterates over all C training identities and Z_c is the number of training images for identity c, while $\vec{\mu}_c$ and Σ_c are mean and covariance matrix (which are calculated similar to eq. (5)) of it. In LDA, the projection matrix Φ maximizes Fishers optimization criterion by solving the generalized eigenvalue problem [111]:

$$\mathcal{S}_b \Phi = \Lambda \mathcal{S}_w \Phi \tag{8}$$

i. e., by computing the eigenvectors for $\mathcal{S}_w^{-1}\mathcal{S}_b$, which requires $\mathcal{S}_w$ to have full rank. Finally, the number of eigenvectors of the LDA projection matrix is usually reduced. In theory the LDA feature space cannot exceed dimension $C - 1$ since $\mathcal{S}_b$ and therewith $\mathcal{S}_w^{-1}\mathcal{S}_b$ have rank $C - 1$.

As [111] showed it is beneficial to combine PCA and LDA rather than performing LDA directly on the features. Since both methods are linear the final projection matrix: $\Phi_{\text{PCA+LDA}} = \Phi_{\text{PCA}}\,\Phi_{\text{LDA}}$ can be obtained by multiplying PCA and LDA projection matrices. Again, the columns of the combined projection matrix can be reinterpreted as images. The first five *Fisher faces* of the MOBIO database are displayed in fig. 2(c).

In literature, many state-of-the-art face recognition algorithms use PCA or LDA for dimensionality reduction of any type of features. For example, [25] showed that the Gabor-based PCA+LDA algorithm works better than PCA and PCA+LDA on raw pixel values. In [32] the best algorithm used a multi-representation PCA combining Gabor features, local binary patterns and color information, while the second best algorithm performed LDA on features learnt by a convolutional neural network.

2.2.3 Face recognition in subspaces

For PCA or LDA based face recognition algorithms features are projected into face space: $\vec{y} = \Phi^\top \vec{v}$ and compared by a distance measure. Well-known measures are the Manhattan distance d_1, the Euclidean distance d_2, the Canberra distance d_C, the Mahalanobis distance d_Λ, the normalized scalar product $d_{\cos}$ and the correlation d_{cor}:

$$\begin{aligned}
d_1(\vec{y}, \vec{y}') &= \sum_i |y_i - y_i'| & d_2(\vec{y}, \vec{y}') &= \sum_i (y_i - y_i')^2 \\
d_C(\vec{y}, \vec{y}') &= \sum_i \frac{|y_i - y_i'|}{y_i + y_i'} & d_\Lambda(\vec{y}, \vec{y}') &= \sum_i \frac{(y_i - y_i')^2}{\lambda_i} \\
d_{\cos}(\vec{y}, \vec{y}') &= 1 - \frac{\sum_i y_i\, y_i'}{\|\vec{y}\|\,\|\vec{y}'\|} & d_{\text{cor}}(\vec{y}, \vec{y}') &= 1 - \frac{(\vec{y} - \overline{y})^\top (\vec{y}' - \overline{y}')}{\|(\vec{y} - \overline{y})\|\,\|(\vec{y}' - \overline{y}')\|}
\end{aligned} \tag{9}$$

For subspace-based face recognition algorithms there is no default way of enrolling a model $\mathcal{M}^{(c)}$ from more than one projected feature of client c. In this paper we test several *scoring strategies*. In all cases we store all features in the model: $\mathcal{M}^{(c)} = \{\vec{y}^{(z)} \mid z = 1, \ldots, Z_c\}$. Afterward, we either compute the average distance:

$$d^{\text{avg}}(\mathcal{M}^{(c)}, \vec{y}) = \frac{1}{Z_c} \sum_{z=1}^{Z_c} d(\vec{y}^{(z)}, \vec{y}) \tag{10}$$

which corresponds to averaging the model features, or we use the maximum, median or minimum distance:

$$d^{\texttt{operator}}(\mathcal{M}^{(c)}, \vec{y}) = \underset{z \in \{1,\ldots,Z_c\}}{\texttt{operator}}\, d(\vec{y}^{(z)}, \vec{y}) \quad \texttt{operator} \in \{\max, \text{med}, \min\} \tag{11}$$

Several variations of PCA and LDA algorithms are proposed in literature, e. g. [104, 65, 51, 50] to cite only a few, a more exhaustive survey can be found in [44]. It is impossible to test all of them in this paper. Still, we select two additional variations: *local region PCA* (LRPCA), which computes PCA's for several local regions of the face like the eyes, the nose, and the mouth [69]; and LDA-IR,[7] which exploits color information of two color layers of the image [53]. We employ these algorithms for two reasons. First, they are the baseline algorithms for *the good, the bad and the ugly* face recognition challenge [69, 53], which is part of the *multi biometrics grand challenge* (MBGC) evaluation.[8] The second reason is simply that the source code for the algorithms is provided by the CSU. The configurations of the algorithms, which are provided by the authors, are optimized to the GBU database. Besides disabling the cohort score normalization of the LDA-IR algorithm since we are not dealing with score normalization in this paper, we use these optimized configurations in our tests.

2.3 Gabor wavelet based algorithms

2.3.1 Gabor wavelet transform

Another set of face recognition algorithms rely on *Gabor features*, which are found to model the (retinal) image processing in the primary visual cortex of mamal brains [18]. A *Gabor wavelet* [102]:

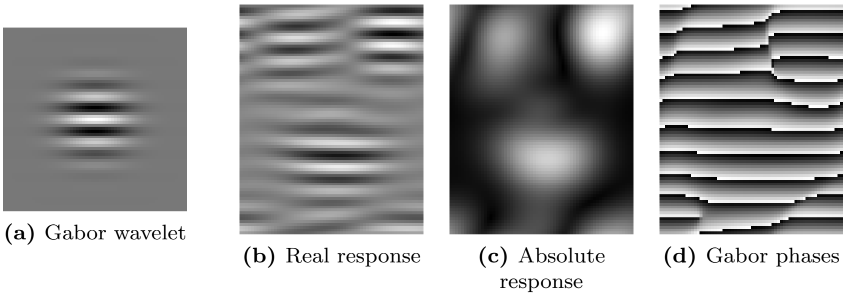
$$\psi_{\vec{k}_j}(\vec{x}) = \frac{\vec{k}_j^2}{\sigma^2}\, e^{-\frac{\vec{k}_j^2 \vec{x}^2}{2\sigma^2}} \left[e^{i\vec{k}_j^\top \vec{x}} - e^{-\frac{\sigma^2}{2}} \right] \tag{12}$$

is an image filter that consists of a planar complex wave $e^{i\vec{k}_j^\top \vec{x}}$ that is confined by an enveloping Gaussian and normalized to be mean free. A Gabor wavelet is parametrized by the width σ of the Gaussian, as well as the spatial orientation φ and frequency k of the complex wave, which are encoded in the wave vector $\vec{k}_j$:

$$\vec{k}_j = \vec{k}_{\nu,\mu} = k_\nu \begin{pmatrix} \cos \varphi_\mu \\ \sin \varphi_\mu \end{pmatrix} \tag{13}$$

[7] In [53] the LDA-IR algorithm is called CohortLDA. Since we do not use any score normalization in our experiments we also disable the cohort. To avoid confusions we choose the name LDA-IR, which we take from a former version of the CSU baseline code.

[8] `http://www.nist.gov/itl/iad/ig/mbgc.cfm`



(a) Gabor wavelet **(b)** Real response **(c)** Absolute response **(d)** Gabor phases

Figure 3: GABOR WAVELET RESPONSES. *This figure displays the (b) real part, (c) absolute values, and (d) Gabor phases of the convolution of the aligned image from fig. 1(b) with the (a) Gabor wavelet.*

Commonly [84], a *family* of 40 Gabor wavelets are used to extract Gabor features. This family is defined by a discrete set of wave vectors [100]:

$$k_\nu = k_{\max}\, k_{\text{fac}}^{\nu} \qquad\qquad \varphi_\mu = \frac{\mu\, 180^\circ}{\mu_{\max}} \tag{14}$$

including [99, 106, 33] $\nu_{\max} = 5$ frequencies: $\nu = 0, \ldots, 4$ starting with the highest frequency $k_0 = k_{\max} = \frac{\pi}{2}$, and $\mu_{\max} = 8$ orientations: $\mu = 0, \ldots, 7$.

Complex valued Gabor features are extracted by convolving the image with each of the 40 Gabor wavelets:

$$\forall j = 1, \ldots, 40\colon \mathcal{I}^{(j)}(\vec{x}) = \left(\mathcal{I} * \psi_{\vec{k}_j}\right)(\vec{x}) \tag{15}$$

Traditionally, only the absolute parts of these complex valued features are taken into account [46, 35, 25, 108], whereas lately the complex phases of Gabor features raised the interest of face recognition researchers [109, 106, 33].

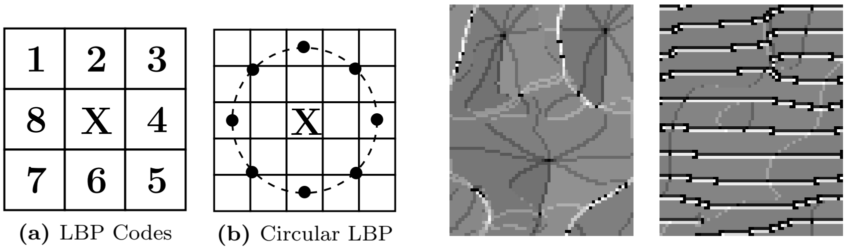
Based on Gabor wavelet responses, several algorithms are defined. The first and most well-known example is the *elastic bunch graph matching* (EBGM) that was first proposed in the late 1990s [99]. Later, [25] used Gabor wavelet responses and proposed the GPCA+LDA algorithm, whereas, e. g., [105] selected the most effective Gabor features using AdaBoost. A good overview of the usage of Gabor wavelet responses in face recognition can be found in [84, 83]. As state-of-the-art representatives we investigate two face recognition algorithms, one of which is based on local Gabor binary patterns, and one performing Gabor graph comparisons.

2.3.2 Local Gabor binary pattern histogram sequence

First, we explore the *local Gabor binary pattern histogram sequence* (LGBPHS) [108, 109], which is an extension of the *local binary pattern histogram sequence* (LBPHS) [2].

A *local binary pattern* (LBP) [67] is generated by comparing the gray value of pixel $\mathcal{I}(\vec{x})$ with the gray values of neighboring pixels. Each neighbor $\vec{x}_i$ ($i = 1, \ldots, 8$) (how the indexes i relate to the neighbor positions is shown in fig. 4(a)) defines a bit in the LBP code:

$$\mathcal{I}_{\text{LBP}}(\vec{x}) = \sum_{i=1}^{8} 2^{i\text{-}1} t_i \qquad\qquad t_i = \begin{cases} 0 & \text{if } \mathcal{I}(\vec{x}) < \mathcal{I}(\vec{x}_i) \\ 1 & \text{else} \end{cases} \tag{16}$$



(a) LBP Codes **(b)** Circular LBP **(c)** LGBP **(d)** ELGBP

Figure 4: Local Gabor binary patterns. *This figure displays the generation process of (a) LBP codes and (b) the circular LBP$^{u2}_{8,2}$ operator. Additionally, the results of the LBP$^{u2}_{8,2}$ operator on (c) the absolute Gabor wavelet responses and (d) the Gabor phases (as given in as given in fig. 3(c) and fig. 3(d)) are shown.*

Later, LBP codes were extended to LBP$^{u2}_{P,R}$, which are circular with P bits and radius R (see fig. 4(b) for an 8 bit circular LBP with $R = 2$) and *uniform* (u2) [68].

Since each bit of the LBP code is similarly important these codes cannot be compared with a simple distance function. Instead, LBP codes are collected in a histogram:

$$H_q = |\{\mathcal{I}_{\text{LBP}}(\vec{x}) \mid \mathcal{I}_{\text{LBP}}(\vec{x}) = q\}| \tag{17}$$

where q iterates over the different LBP codes that can be generated. Additionally, the image can be split into B (possibly overlapping) blocks, and a histogram $\mathcal{H}_b$ is computed for each block b [2]:

$$H_{b;q} = |\{\mathcal{I}_{\text{LBP}}(\vec{x}) \mid \mathcal{I}_{\text{LBP}}(\vec{x}) = q \wedge \vec{x} \in \text{block } b\}| \tag{18}$$

The extraction of LBP features from the boundaries of the blocks is not detailed in literature. We here choose the blocks to have additional R pixels in all directions. This includes that the borders of the image also have to be extended by R pixels into all direction, which already has to be accounted for during image preprocessing.

In [108] the circular LBP is applied to the absolute values of Gabor wavelet responses. An example of a LBP codes extracted from the absolute responses of a Gabor wavelet is displayed in fig. 4(c). A histogram is extracted for the response $\mathcal{I}^{(j)}$ of each Gabor wavelet $\psi_{\vec{k}_j}$, $j = 1, \ldots, 40$ and for each block $b = 1, \ldots, B$:

$$H_{j;b;q} = \left|\{\mathcal{I}^{(j)}_{\text{LBP}}(\vec{x}) \mid \mathcal{I}^{(j)}_{\text{LBP}}(\vec{x}) = q \wedge \vec{x} \in \text{block } b\}\right| \tag{19}$$

Similarly, the LBP codes are also extracted from image blocks of Gabor phases [109]; exemplary LBP codes extracted from Gabor phases are shown in fig. 4(d). Concatenating all these histograms into one *histogram sequence* $\mathcal{H}$ ends up in a huge feature vector, which is called the *extended local Gabor binary pattern histogram sequence* (ELGBPHS) [109].

2.3.3 Face recognition with histogram sequences

Comparisons of histograms are usually carried out using histogram similarity measures. Several methods have been proposed, amongst them are the histogram intersection d_{HI}, the χ^2 measure d_{χ^2} and the Kullback-Leibler divergence d_{KL}:

$$\begin{aligned} d_{\mathrm{HI}}(H,H') &= -\sum_q \min\{H_q, H'_q\} \\ d_{\chi^2}(H,H') &= \sum_q \frac{(H_q - H'_q)^2}{H_q + H'_q} \\ d_{\mathrm{KL}}(H,H') &= \sum_q (H_q - H'_q)\,\log\frac{H_q}{H'_q} \end{aligned} \tag{20}$$

Comparing two histogram sequences can be performed as a weighted sum over the histograms [108]:

$$d(\mathcal{H},\mathcal{H}') = \sum_{j;b} w_{j;b}\, d(\mathcal{H}_{j;b}, \mathcal{H}'_{j;b}) \tag{21}$$

whereas in this work we use identical weights $w_{j;b} = 1$, throughout.

The model enrollment stage of (E)LGBPHS is, again, not detailed in any publication. Here, we enroll a model $\mathcal{M}^{(c)}$ from several $\mathcal{H}^{(z)}$ features of the same identity by computing the average histogram sequence:

$$\mathcal{M}^{(c)} = \frac{1}{Z_c}\sum_z \mathcal{H}^{(z)} \tag{22}$$

ending in non-integral numbers of elements in the model histogram.

2.3.4 Gabor graphs

Another type of features based on Gabor wavelet responses is the Gabor jet [99], which we here use in grid graphs [33]. A Gabor jet is a local texture feature. It is generated by concatenating the responses of all Gabor wavelets at a certain position in the image:

$$\forall j \in \{1,\dots,40\}: \quad \mathcal{J}^{\mathcal{I}}_j(\vec{x}) = \mathcal{I}^{(j)}(\vec{x}) \tag{23}$$

Commonly [99, 33], the Gabor jet $\mathcal{J}_j = a_j\, e^{i\phi_j}$ stores the responses as absolute values a_j and phases ϕ_j. It is beneficial [46] to normalize the absolute values in a Gabor jet to unit Euclidean length: $\sum_j a_j^2 = 1$.

A face representation is generated by extracting Gabor jets at several positions in the image. Many approaches [99, 35, 63, 86] generate *face graphs* by taking Gabor jets at fiducial locations in the face — so-called facial landmarks. In this work, we use grid graphs instead, because a) we do not have hand-labeled locations for all landmarks for all databases and it is difficult to detect them automatically; and b) [88] indicated that grid graphs on average perform better than face graphs. The grid graph $\mathcal{G}$ is defined by extracting the Gabor jets in a regular grid with a given inter-node-distance g:

$$\mathcal{G}_n = \mathcal{J}^{\mathcal{I}}(\vec{x}_n) \quad \vec{x}_n = \frac{\mathcal{R}}{2} - \left(\left\lfloor\frac{\mathcal{R}}{g}\right\rfloor - 1\right)\frac{g}{2} + g\begin{pmatrix} n_x \\ n_y \end{pmatrix} \quad \begin{matrix} 0 \le n_x < \left\lfloor \frac{r_x}{g} \right\rfloor \\ 0 \le n_y < \left\lfloor \frac{r_y}{g} \right\rfloor \end{matrix} \tag{24}$$

where n enumerates all variations of n_x and n_y.

The model enrollment here uses the *bunch graph* [99] concept. For each node position, the Gabor jets from all enrollment graphs are stored:

$$\mathcal{M}_n^{(c)} = \{\mathcal{G}_n^{(z)} \mid z \in 1, \ldots, Z_c\} \tag{25}$$

For the comparison of model $\mathcal{M}^{(z)}$ and probe $\mathcal{G}$, we investigate several scoring strategies:

$$S(\mathcal{M}^{(c)}, \mathcal{G}) = \frac{1}{N\, Z_c} \sum_z \sum_n S(\mathcal{G}_n^{(z)}, \mathcal{G}_n) \tag{26}$$

$$S(\mathcal{M}^{(c)}, \mathcal{G}) = \frac{1}{Z_c} \sum_z \underset{n}{\texttt{operator}}\, S(\mathcal{G}_n^{(z)}, \mathcal{G}_n) \tag{27}$$

$$S(\mathcal{M}^{(c)}, \mathcal{G}) = \frac{1}{N} \sum_n \underset{z}{\texttt{operator}}\, S(\mathcal{G}_n^{(z)}, \mathcal{G}_n) \tag{28}$$

where `operator` can be any of $\{\min, \max, \text{med}\}$ and $S(.,.)$ is one of the Gabor jet similarity functions [99, 46, 33]:

$$\begin{aligned} S_a(\mathcal{J}, \mathcal{J}') &= \sum_j a_j\, a'_j & S_D(\mathcal{J}, \mathcal{J}') &= \sum_j a_j\, a'_j \cos(\phi_j - \phi'_j - \vec{k}_j^\top \vec{d}) \\ S_C(\mathcal{J}, \mathcal{J}') &= \frac{1}{40} \sum_j \frac{|a_j - a'_j|}{a_j + a'_j} & S_n(\mathcal{J}, \mathcal{J}') &= \frac{1}{40} \sum_j \cos(\phi_j - \phi'_j - \vec{k}_j^\top \vec{d}) \end{aligned} \tag{29}$$

or the combination $S_{n+C}(\mathcal{J}, \mathcal{J}') = S_n(\mathcal{J}, \mathcal{J}') + S_C(\mathcal{J}, \mathcal{J}')$ [33]. In eq. (29), S_a and S_C similarity functions use only the absolute values of the Gabor jets, while S_n and S_D exploit differences of Gabor phases. Since the simple phase difference $\phi_j - \phi'_j$ is unstable — small displacements in the image can lead to large changes in Gabor phases, see fig. 3(d) — they are corrected using the *disparity* $\vec{d}$ that is estimated from the two Gabor jets (cf. [33] for details).

2.4 Generative algorithms

An alternative to previously detailed discriminative approaches to automatic face recognition is to describe the face of a client by a generative model. The overall idea is to extract local features from the image of a subject's face before modeling the distribution of these features with a generative model [82, 15], instead of concatenating them as usually done in discriminative algorithms. It has been shown that such an approach offers descent performance with a reasonable complexity [14]. In addition, extensions of this model, such as *inter-session variability* (ISV) modeling, have recently been proposed [96, 58] to improve the robustness to image variations.

2.4.1 Parts-based features

Parts-based features [82] are extracted by decomposing preprocessed images into B overlapping blocks. In these blocks the intensity of the pixels is normalized to zero mean and unit variance in order to reduce the impact of residual illumination variations. Afterward, a *2D discrete cosine transform* (DCT) is applied

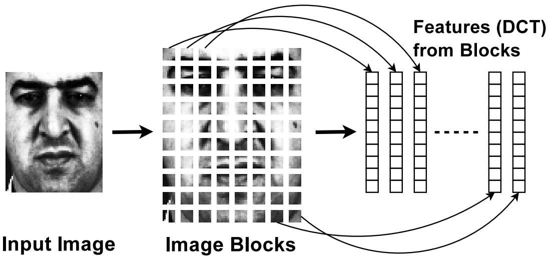


Figure 5: DCT FEATURES EXTRACTION. *This figure shows the computation of parts-based features by decomposing an image into a set of blocks and extracting DCT features from each block.*

to each block, before extracting the F lowest-frequency DCT coefficients that form the descriptor of a given block. For a given image the resulting block-based feature vectors are normalized to zero mean and unit variance in each dimension [97]. Each preprocessed image is finally described by a set of B features vectors — so-called *observations* — $O = \{\vec{o}^1, \ldots, \vec{o}^B\}$ each of dimensionality F.

2.4.2 Gaussian mixture models

The distribution of the features for a given client are modeled by a *Gaussian mixture model* (GMM). A GMM is a generative model that consists of K multivariate Gaussian components [77]. Each component k is defined by a mean vector $\vec{\mu}_k$, a co-variance matrix Σ_k, which can be assumed to be diagonal [15], and a weight w_k. A GMM is fully described by a set of parameters $\Theta = \{w_k, \vec{\mu}_k, \Sigma_k\}_{k=1,\ldots,K}$. Given these parameters Θ the likelihood of the feature vectors O is:

$$P(O \mid \Theta) = \prod_b \sum_k w_k \mathcal{N}\left[\vec{o}^b \mid \vec{\mu}_k, \Sigma_k\right] \tag{30}$$

where $\mathcal{N}$ is the multivariate Gaussian with mean $\vec{\mu}_k$ and covariance matrix Σ_k.

To use GMMs for face recognition a model $\mathcal{M}^{(c)}$ needs to be generated for each client c. The main difficulty is that the number of enrollment images is usually limited, possibly to a single sample. To overcome this issue [77] proposed to use a *universal background model* (UBM) as a prior and to adapt this prior to the enrollment samples of a client c. This UBM $\mathcal{M}^{(\mathrm{ubm})}$ is a Gaussian mixture model, which is trained on feature vectors extracted from the images of the training set by using the iterative *expectation-maximization* (EM) algorithm [20]. The client model $\mathcal{M}^{(c)}$ is adapted from $\mathcal{M}^{(\mathrm{UBM})}$ towards the enrollment features using a *maximum a posteriori* (MAP) estimation [26]. This allows to generate client models from limited enrollment samples.

It has been shown [26, 52, 14] that mean-only MAP adaptation, where only the means $\vec{\mu}_k$ of the Gaussian components are adapted, performs well in practice. Mean-only MAP adaptation can be written in a compact form when using the GMM super-vector notation. The mean super-vector $\vec{m}$ of a GMM is obtained by concatenating the means $\vec{\mu}_k$ of its Gaussian components: $\vec{m} = \left[\vec{\mu}_1^\top, \ldots, \vec{\mu}_K^\top\right]^\top$. The enrollment of client model $\mathcal{M}^{(c)}$ can be written as:

$$\vec{m}^{(c)} = \vec{m}^{(\mathrm{ubm})} + \vec{d}^{(c)} \tag{31}$$

where $\vec{m}^{(c)}$ is the mean-super-vector of client model $\mathcal{M}^{(c)}$, $\vec{m}^{(\text{ubm})}$ the mean-super-vector of $\mathcal{M}^{(\text{ubm})}$ and $\vec{d}^{(c)}$ a client specific offset that is estimated by the MAP algorithm.

In the generative approach the scoring procedure is different than in the discriminative case. For a given set of observations (feature vectors) $O = \{\vec{o}^1, \ldots, \vec{o}^B\}$ of a probe image, the probability that these observations are generated by a given client model $\mathcal{M}^{(c)}$ is estimated as the *log-likelihood ratio* (LLR):

$$S_{\text{GMM}}\left(O \mid \mathcal{M}^{(c)}\right) = \log\left(\prod_{b=1}^{B} \frac{P\left(\vec{o}^b \mid \vec{m}^{(c)}\right)}{P\left(\vec{o}^b \mid \vec{m}^{(\text{ubm})}\right)}\right) \tag{32}$$

This LLR is a measure for the probability that the probe observations are generated by the client model $\mathcal{M}^{(c)}$ rather than by anyone else, i. e., by the $\mathcal{M}^{(\text{ubm})}$. It is worth mentioning that this LLR is of particular interest of forensic science. Indeed, generative approaches provide a tool for forensic experts and judiciary to interpret evidence directly in terms of probabilities for the *prosecution* and the *defense* hypotheses [5].

2.4.3 Session variability modeling

Challenges in face recognition are often caused by variations in pose, expression, illumination or environment, which are coined as *session variability.* In the context of a GMM-based system, *inter-session variability* (ISV) modeling [95] is a technique that has been successfully employed for face recognition [96].

In contrast to the GMM approach as introduced above, ISV aims at enrolling the model by suppressing session-dependent components and yielding the true session-independent client Gaussian mixture model $\mathcal{M}^{(c)}$, which is described by its mean-super-vector $\vec{m}^{(c)}$:

$$\vec{m}^{(c)} = m^{(\text{ubm})} + \vec{d}^{(c)} + \vec{u}_z^{(c)} \tag{33}$$

Compared to eq. (31) the additional vector $\vec{u}_z^{(c)}$ models the session-dependent components that are contained in the z^{th} enrollment image of the client.

At test time, given feature vectors O extracted from a probe sample, session offsets $\vec{u}$ against both the client model $\mathcal{M}^{(c)}$ and the UBM $\mathcal{M}^{(\text{ubm})}$ are estimated before computing the LLR score:

$$S_{\text{ISV}}\left(O \mid \mathcal{M}^{(c)}\right) = \log\left(\prod_{b=1}^{B} \frac{P\left(\vec{o}^b \mid \vec{m}^{(c)} + \vec{u}^{(c)}\right)}{P\left(\vec{o}^b \mid \vec{m}^{(\text{ubm})} + \vec{u}^{(\text{ubm})}\right)}\right) \tag{34}$$

One of the main assumption of ISV is that the session offset lies in a linear subspace U of the GMM mean super-vector space: $\vec{u} = U\vec{x}$, with $\vec{x}$ being a latent variable. A more detailed description of this algorithm can be found in [95, 58].

2.5 Algorithm configurations

Each of the introduced algorithms has a couple of meta-parameters, which might affect the algorithm performance and which have to be carefully chosen by the researcher. To not confuse these meta-parameters with parameters that are automatically estimated in the training steps of the algorithms, we refer to the meta-parameters as the *algorithm configuration*.

For PCA or PCA+LDA based algorithms, the dimensions of PCA and LDA sub-spaces is part of the configuration, as well as the employed distance function and scoring strategy. Gabor wavelet based algorithms have to specify several parameters that are bound to the Gabor wavelet family. These parameters are the number of frequencies $\nu_{\max}$ and orientations $\mu_{\max}$ as well as the highest frequency $k_{\max}$, the (logarithmic) distance between two frequencies k_{fac}, and the size σ of the enveloping Gaussian. Additionally to that, LGBPHS has to select the configuration of the local binary patterns, i. e., the radius, the number of neighbors, whether to use only uniform LBP codes and whether to exploit the Gabor phases, as well as the size and overlap of the blocks, from which the histograms are extracted, and the employed histogram comparison measure. The Gabor graphs algorithm has to set the Gabor wavelet parameters (see above), the inter-node distance, the Gabor jet similarity measure and the scoring strategy. Finally, for ISV the size and overlap of the blocks, from which the DCT features are extracted, need to be selected and the dimensionality of the feature vectors F, number of Gaussian components K and dimension of the U subspace have to be determined.

3 Databases and evaluation protocols

To guarantee a fair comparison of algorithms it is required that all algorithms are given the same image data for training and enrollment, and the same pairs of models and probe images are evaluated. This is achieved by defining evaluation protocols, which might either be *biased*, i. e., having (partially) the same identities in the training and the test set, or *unbiased* by splitting the identities between the sets. For all databases listed below we provide an implementation of the protocols on GitHub and PyPI. A list of implemented database interfaces — so-called *satellite packages* — is given on the Bob website.[9]

One possible separation between image databases is the way, image variations like illumination, facial expression, occlusion and pose are handled. In *controlled* databases some or all image variations are enforced, while *uncontrolled* databases include images as they would occur in every day life conditions.

3.1 Evaluation protocols

In this paper we use several image databases including their default evaluation protocols. Depending on the database these protocols define either an identification or a verification scenario. Identification protocols usually report results in terms of *recognition rates* (RR) or *cumulative match characteristics* (CMC) curves. To compute CMC curves, for each probe image $\mathcal{I}^{(p)}$, the *rank*:

$$r(\mathcal{I}^{(p)}) = \left|\left\{\mathcal{M}^{(c)} \mid S(\mathcal{M}^{(c)}, \mathcal{I}^{(p)}) \geq S(\mathcal{M}^{(c^*)}, \mathcal{I}^{(p)}), c \in \{1, \ldots, C\}\right\}\right| \tag{35}$$

is computed as the number of models $\mathcal{M}^{(c)}$ that are more similar than model $\mathcal{M}^{(c^*)}$ of the correct identity. If the correct model has the highest similarity,

[9] `http://github.com/idiap/bob/wiki/Satellite-Packages`

rank $r = 1$ is assigned. For each rank r the CMC curve counts how many probe images have a rank r or lower, normalized by the total number of probe images. Finally, the RR is extracted from rank $r = 1$ of the CMC curve and, hence, is the relative number of correctly identified probe images.

For verification protocols there exist several evaluation measures. All of them are built on top of the *false acceptance rate* (FAR) and the *false rejection rate* (FRR). To compute these rates, the scores have to be split up into *client scores* $s_{\text{cli}} = S(\mathcal{M}^{(c^*)}, \mathcal{I}^{(p)})$ comparing model and probe from the same identity, and *impostor scores* $s_{\text{imp}} = S(\mathcal{M}^{(c)}, \mathcal{I}^{(p)}),\ c \neq c^*$ comparing model and probe of different identities. FAR and FRR are defined over a certain threshold θ:

$$\text{FAR}(\theta) = \frac{|\{s_{\text{imp}} \mid s_{\text{imp}} \geq \theta\}|}{|\{s_{\text{imp}}\}|} \qquad \text{FRR}(\theta) = \frac{|\{s_{\text{cli}} \mid s_{\text{cli}} < \theta\}|}{|\{s_{\text{cli}}\}|} \tag{36}$$

Based on these two curves, different quality measures are defined. The *receiver operating characteristics* (ROC) plots the *correct acceptance rate* (CAR $= 100\% - \text{FRR}$) over the FAR. If a single value is required for a simple comparison, commonly [71] the CAR at FAR $= 0.1\%$ is chosen, which is also known as the *verification rate* (VR).

Some protocols also split the data in three sets: a training set, a development set and an evaluation set. Scores and FAR/FRR are computed for both the development and the evaluation set independently. Then, a threshold θ^* is obtained based on the intersection point of FAR and FRR curves of the development set. This threshold is used to compute the *equal error rate* (EER) on the development set and the *half total error rate* (HTER) on the evaluation set:

$$\text{EER} = \frac{\text{FAR}_{\text{dev}}(\theta^*) + \text{FRR}_{\text{dev}}(\theta^*)}{2} \qquad \text{HTER} = \frac{\text{FAR}_{\text{eval}}(\theta^*) + \text{FRR}_{\text{eval}}(\theta^*)}{2} \tag{37}$$

Some databases like the LFW database (see sec. 3.3.4 below) also define their own type of evaluation measure. In this case, we respect these evaluation measures in our tests.

3.2 Controlled databases

Four of the databases, where all image variations are controlled, are Multi-PIE, CAS-PEAL, XM2VTS and AR face. Two other databases that we consider to be in the group of controlled image databases are FRGC and GBU. Though the images of the latter databases have (partially) been taken in environments with unrestricted illumination conditions and with some facial expressions, all faces in the images are not occluded and perfectly frontal, i. e., show no out-of-plane rotation.

3.2.1 CMU Multi-PIE

The CMU Multi-PIE database [29], where PIE stands for *pose, illumination, expression*, consists of 755,370 images shot in 4 different sessions from 337 subjects. The Multi-PIE database itself does not provide evaluation protocols, but we generated and published several unbiased face verification protocols ourselves. All these protocols are split up into a training, a development and an evaluation set, where the identities between the sets are disjoint. The training set is composed of all 208 individuals that did not participate in all four sessions, while the size of development set (64 identities) and evaluation set (65 identities) is almost equal.

In this work we use protocols for controlled illumination, expression and pose. In each protocol 5 images per person with neutral facial expression, neutral illumination and frontal pose are used for model enrollment. The probe sets contain images with either non-frontal illumination (U), facial expressions (E) or non-frontal pose (P).

3.2.2 CAS-PEAL

The CAS-PEAL database [25] includes 9031[10] frontal images (and several non-frontal images, which we do not use due to lack of protocol) from 1040 Chinese persons. Using these images six biased identification protocols are provided with the database. Each of the protocols tests a different image variation: facial *expression*, non-frontal *lighting*, *accessory*, different *background*, subject-camera *distance* and *aging*.

Unconventionally, the training set defined by the CAS-PEAL database consists of 1200 images that are a random subset of the images of the evaluation set. In each of the protocols all 1040 neutral and frontally illuminated images serve as model images; models are enrolled from one image per person only. For the probe sets the number of images and subjects differ between protocols, a complete list is given in [25].

3.2.3 Extended M2VTS database

XM2VTS [59] is a comparably small database of 295 subjects. We here use only the *darkened* protocol, which compares frontally illuminated images with non-frontally illuminated ones. The enrollment of a client model is performed with 3 images per client, whereas 4 probe files per identity are used to compute the scores. The training set consists of exactly the same images as used for model enrollment [59], making the protocol biased.

3.2.4 AR face database

The AR face database [55] contains 3312 images[11] from 76 male and 60 female clients taken in two sessions. Facial images in this database include three variations: facial expressions, strong controlled illumination and occlusions with sunglasses and scarfs.

We have created and published several unbiased verification protocols for this database, splitting up the identities into 50 training subjects (28 men and 22 women) and each 43 clients (24 male and 19 female) in the development and evaluation set. For model enrollment we use those two images per client that have neutral illumination, neutral expression and no occlusion. The protocols *expression*, *illumination*, *occlusion*, *occlusion_and_illumination* test the specific image variations that are defined in the database, i. e., probe images have either

[10] unlike the number 9032 incorrectly reported in [25]

[11] The website `http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html` reports more than 4000 images, but we could not reach the controller of the database to clarify the difference.

non-neutral facial expression, non-frontal illumination, partially occluded faces, or both occlusion and illumination.

3.2.5 Face recognition grand challenge

The *face recognition grand challenge* (FRGC) database in its version ver2.0 contains 36818 high resolution images of 466 clients that were collected in various sessions during four years. Additionally, the database also includes 3D image data, but these are not used in this paper. The database provides several biased protocols [70] (named *experiments* by the authors), three of which utilize 2D image data only: experiments 2.0.1, 2.0.2 and 2.0.4.

Experiments 2.0.1 and 2.0.2 compare only controlled images that were taken in a studio environment. While experiment 2.0.1 provides one image to enroll a client model, experiment 2.0.2 enrolls a client model from four images. Similarly, in experiment 2.0.1 a score is computed by comparing a client model with a single probe, whereas for experiment 2.0.2 four probe images per person are integrated to build a single score. Finally, experiment 2.0.4 uses the same client models as experiment 2.0.1, but probe images that were taken in a corridor or outdoors with unconstrained illumination conditions.

For each experiment the protocols define different *masks* (sub-protocols), which specify pairs of model/probe that should be taken to evaluate. In our experiments we use the most difficult *mask III*, throughout. The training set, which is identical for the three experiments, contains 12776 studio and corridor images from 266 clients. The clients of the training set form a subset of the test clients, making these protocols biased.

3.2.6 The good, the bad & the ugly

The good, the bad and the ugly (GBU) database [69] is built from 8638 high resolution frontal outdoor images of 782 clients. It defines the three unbiased protocols *Good*, *Bad* and *Ugly*, which specify image pairs that should be compared. Each protocol includes 1085 different images that are used to enroll client models — each model is enrolled from a single image and there exists several client models per identity. Likely, 1085 probe images are defined by each protocol and all models are compared with all probes to compute the final ROC curves. Additionally, four different training sets are present; we take the largest set *x8* in all our experiments on the GBU database, unlike [69], who used the *x2* training set to train the LRPCA algorithm.

3.3 Uncontrolled databases

Since we do not want to restrict the applications to use controlled image data we also investigate four challenging databases that contain images captured under completely uncontrolled conditions as they would appear in every day life.

3.3.1 BANCA

The first uncontrolled database we explore is BANCA [9]. Originally, it captures video and audio recordings of 52 persons for each 4 different languages to utter prompted sequences. Recordings were taken in 12 different sessions, where in each session every subject generated two videos, one true client access and one informed impostor access. From each of these videos, 5 images and one audio signal was extracted. However, only the English language was made available [9] and, hence, in this work we use only the images of the BANCA English database.

Several unbiased *open set* verification protocols are proposed with the database [9]. We here take only the most challenging protocol P, which enrolls client models on 5 *controlled* images, but probes the system with *controlled*, *degraded* and *adverse* images (for details see [9]). Two particularities of this database are that it is small, e. g., the training set consists of only 300 images and that the number of 2340 client and 3120 impostor scores is balanced.

3.3.2 Mobile biometry

The *mobile biometry* (MOBIO) database [56] consists of video data of 152 people taken with mobile devices like mobile phones or a laptop, we here use only the mobile phone data. For each client up to 12 different sessions were recorded. From each of these recordings one image was extracted by choosing a single frame after 1 second of video run time. Only two clients are skipped since the face is not visible in the video. These images differ in facial expression, pose, illumination conditions, and sometimes parts of the face are not captured by the device.

The MOBIO database provides two gender-specific unbiased evaluation protocols *female* and *male*, where exclusively female or male images are compared. In these protocols, 5 images per client are used to enroll a client model and all probe files are tested against all models of the same gender.

The training set consists of 9600 images from 13 females and 37 males. In our experiment we solely perform *gender-independent* training. The development set contains 18 female and 24 male clients, which are probed with 1890 or 2520 images, respectively. The evaluation set is a bit larger, it embraces 20 female and 38 male clients, using 2100 or 3990 probe files, respectively.

3.3.3 Surveillance camera face database

One of the most challenging image databases is the *surveillance camera* (SC) face database [28]. It contains images of 130 subjects taken by surveillance cameras. In total 5 different cameras took pictures in 3 different subject-camera-distances: *close*, *medium* and *far*. Since the cameras are anchored slightly above the head position of the clients, especially the *close* images are captured in a viewing angle slightly from above. Furthermore, with a face size of around 20 pixels the *far* images are smaller than the resolution used in our experiments in sec. 4 and, therefore, are scaled up during the image alignment, which is leading to blurred images. In any case, a client model is enrolled from one frontal mug shot image with decent resolution.

Additional to the *combined* protocol we defined three different protocols separated by the camera distance: *close*, *medium* and *far*. In each protocol probe images of 44 (development set) or 43 (evaluation set) subjects are compared to all client models. The images of the remaining 43 subjects are used for training.

3.3.4 Labeled faces in the wild

One of the most popular image databases is *the labeled faces in the wild* (LFW) database [41]. This database contains 13233 face images from 5749 celebrities, which were downloaded from the internet, labeled with the name of the celebrity that is shown in the image and cropped by a face detector. In this work we use the images aligned by the funneling algorithm [40]. The database itself does not provide the eye locations for the images, but we rely on the publicly available[12] annotations [30]. They consist of 9 facial feature points (mouth corners, eyes corners and nose) obtained by a facial feature detector [23]. The locations of the eye centers are estimated by computing the midpoint of the eye corners.

The particularity of the LFW database is that it specifies pairs of images, for which a score should be computed, equally distributed over client and impostor pairs. In our case we always chose the first image of the pair for model enrollment and the second image as probe. For the training sets LFW permits two alternatives: *image-restricted* defining specific image pairs that might be used for training, and *unrestricted* using all images of the training subjects. Here, we chose the *unrestricted* setup since some algorithms need to know the identity information of the training images, which is forbidden to be used in the *image-restricted* training setup.

The LFW database is split into two so-called *views*. Since *view 1* is considered to optimize algorithm configurations we only use *view 2* to report the final results. In *view 2* the subjects are split into 10 different subsets, so-called *folds*. Each fold contains 300 intrapersonal and 300 extrapersonal image pairs, for which the *classification success* is computed:

$$\mathrm{CS} = \frac{|\{s_{\mathrm{cli}} \mid s_{\mathrm{cli}} \geq \theta\}| + |\{s_{\mathrm{imp}} \mid s_{\mathrm{imp}} < \theta\}|}{|\{s_{\mathrm{cli}}\}| + |\{s_{\mathrm{imp}}\}|} \tag{38}$$

In our implementation of the protocol, for each of the 10 experiments 7 folds are used for training, while 2 folds build the *development set*, from which the threshold θ is estimated (cf. eq. (37)) and the last fold is employed to compute the classification success of this fold. Finally, the mean and the standard deviation of the classification successes over all 10 experiments is reported [41].

4 Experiments

To show the performance of the algorithms under various image conditions we execute a series of face recognition experiments, which are explained in more detail in this section. Since there has not been any exhaustive study about the impact of different image resolutions on the recognition performance we first try to find the optimal image resolution. Also, several image preprocessings are evaluated on different databases with controlled and uncontrolled illumination conditions.

To be as fair as possible we optimize the configurations of all of the algorithms taken from Bob [7] independently, see sec. 4.1 for details. We do not optimize the configurations of the LRPCA and LDA-IR algorithms since firstly these configurations have been optimized already — though to another database — and secondly the work of defining new local regions or new color transformations is out of the scope of this paper. We chose the BANCA database to perform the optimization experiments since the database is small, but still quite challenging and not only focused on frontal facial images.

[12] `http://lear.inrialpes.fr/people/guillaumin/data.php`

In the subsequent experiments we run all algorithms with the optimized configurations. To see, which algorithm is best suited to handle facial expressions, facial poses and partial occlusions of the face, in sec. 4.2 we execute the algorithms on some of the controlled databases. In a final set of experiments, which is presented in sec. 4.3, we run the algorithms on different controlled and uncontrolled databases and report the results using their default evaluation methods.

It should be noted that the goal of this study is to provide a replicable survey of a range of face recognition algorithms for research to built upon. It is **not** the goal of this study to demonstrate the superiority of a single best face recognition algorithm.

4.1 Optimizing algorithm configurations

4.1.1 Image resolutions

One important aspect of face recognition is the resolution of the facial image and its content. Interestingly, there are only few papers, e. g., [46, 106, 37] that pay attention to this aspect, but rather every researcher uses his or her own image resolution. To show the diversity of employed image resolutions in face recognition, in the online appendix we present a list of image resolutions, the captured facial areas and the image databases together with an algorithm name that are used in literature.

The first set of experiments that we conduct is to find out, which image resolution is best suited for face recognition. We execute all algorithms with configurations that we have set according to literature. We selected several different image resolutions, ranging from height $r_y = 20$ pixels to $r_y = 200$ pixels, always keeping an aspect ratio of $r_x\colon r_y = 4\colon 5$. Additionally, we always perform the identical image alignment (cf. sec. 2.1.1), such that the right and left eye positions are located at: $\vec{a}_r^* = (\frac{r_y}{4}, \frac{r_y}{5})$ and $\vec{a}_l^* = (\frac{3\,r_y}{4} + 1, \frac{r_y}{5})$, i. e., with inter-eye-distance $\frac{2}{5} r_y$. Only for the LRPCA algorithm we use unit aspect ratio and the eye locations defined by that algorithm, i. e., to assure that the local regions are still capturing the desired information. Note that we do not include LDA-IR in the image resolution evaluation since changing the parametrization of this algorithm in its original implementation is highly complex.

Since in this set of experiments we are only interested in the image resolution, those parameters of the algorithms that are sensitive to resolution are adapted. For example in the Gabor graphs algorithm the maximum frequency of the Gabor wavelets $k_{\max}$ and the node distance are adjusted. For all adaptations, the height $r_y^{\text{ref}} = 80$ pixels is used as the reference height, e. g., by computing $k_{\max} = \frac{\pi}{2} \frac{r_y}{r_y^{\text{ref}}}$ (cf. sec. 2.3.1).

The resulting EER on protocol P of the development set of the BANCA database are given in fig. 6. Interestingly, the results of most of the algorithms are very stable for any image resolution that is at least $\mathcal{R}_{\min} = (32, 40)$ pixels, which corresponds to an inter-eye-distance of 16 pixels. For resolutions smaller than $\mathcal{R}_{\min}$, configuration parameters that are adjusted to the image resolution partially do not make sense anymore (e. g.$k_{\max} > 2\pi$) and, therefore, the results degrade. LDA, ISV and Graphs require resolutions that are a bit higher, but also these algorithms settle around 100 pixels image height.

Since there is not much difference between the resolutions greater than $\mathcal{R}_{\min}$

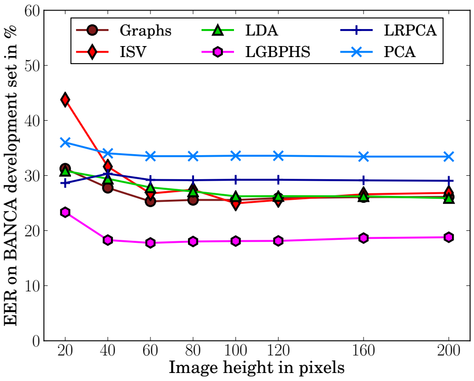


Figure 6: Image resolutions. *This figure displays the EER in % for the algorithms applied to different image resolutions.*

we choose to stick at the resolution $\mathcal{R} = (64, 80)$ as used in [34] for the rest of our experiments.

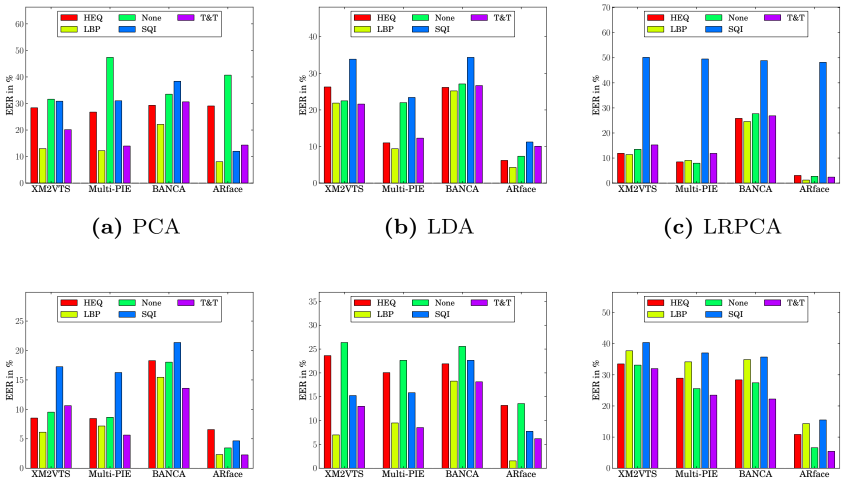
4.1.2 Image preprocessing

One severe issue in automatic face recognition is uncontrolled or strong illumination. Several image preprocessings that should reduce the impact of illumination in face recognition have been proposed (see sec. 2.1). Unfortunately, in literature there is no comprehensive analysis of image preprocessings for face recognition, but in general each researcher uses a single preferred preprocessing, if any.

We perform an evaluation of four different preprocessings. We also test if it might even be better not to have any photometric normalization at all. Since we test only gray level preprocessings we do not execute LDA-IR, which is based on color features. To evaluate the preprocessings we execute them on 3 databases with challenging controlled illumination conditions: the XM2VTS database (protocol *darkened*), the Multi-PIE database (protocol U) and the AR face database (protocol *illumination*). We also test the preprocessings on a database with uncontrolled illumination, for which we again select BANCA (protocol P).

The results of the preprocessing test can be observed in fig. 7. This figure is split up into the six evaluated face recognition algorithms. We can observe that every face recognition algorithm has its own preferred preprocessing. However, there is an overall trend for the LBP-based and the Tan & Triggs preprocessings, while histogram equalization and self quotient image do not perform as well and, obviously, neither taking no preprocessing at all.

Interestingly, the pixel based algorithms PCA (fig. 7(a)), LDA (fig. 7(b)) and LRPCA (fig. 7(c)) obtain lower EER values with the LBP-based preprocessing on all tested databases. For the Gabor wavelet based algorithms LGBPHS (fig. 7(d)) and Graphs (fig. 7(e)) the decision is not as clear. The LBP-based preprocessing performs best on two image databases with strong illumination conditions XM2VTS and AR face, but on BANCA and Multi-PIE the Tan & Triggs preprocessing works better. Though the selection is not easy



(a) PCA **(b)** LDA **(c)** LRPCA

(d) LGBPHS **(e)** Graphs **(f)** ISV

Figure 7: PREPROCESSING. *This figure shows the effect of different image preprocessing algorithms on the recognition performance, evaluated on four image databases.*

we choose Tan & Triggs for both LGBPHS and Graphs, considering that we run more experiments on databases with less strong illumination conditions and that this preprocessing was most commonly used in [54]. Finally, the ISV algorithm (fig. 7(f)) again has a clear preference for Tan & Triggs, which is stable across all tested databases.

4.1.3 Parameter optimization

After finding a suitable image resolution and the optimal image preprocessing for each algorithm we optimize their configurations independently. Due to the partially large number of configuration parameters to be optimized we performed optimization in several steps. Each step groups together configuration parameters that might influence each other. The exact EER's for all steps that are explained in more detail in this section are given in the online appendix. All experiments are executed on the development set of the BANCA database using protocol P.

PCA and LDA based face recognition algorithms For PCA and LDA algorithms only few parameters needs to be selected. In the first step we optimize the subspace dimensions of PCA and LDA, i. e., the number of eigenfaces (see eq. (6)) or Fisher faces we want to keep. Clearly, higher number of PCA dimensions results in lower EER and, hence, we keep 100% of PCA variance. The theoretical limit of LDA subspace dimensions is the number of training clients $C - 1$, which for BANCA is 29. For this LDA dimension, even lower PCA subspace dimensions seem to be sufficient, but still the lowest value is reached at 95% PCA variance. Interestingly, it is worth considering eigenvectors, which belong to zero-valued eigenvalues, the best performance is gained for LDA dimension 199. Nevertheless, since eigenvectors that belong to vanishing eigenvalues are unstable and this work is about reproducible research we stick to the theoretical limit of $C-1$ in the remaining experiments.

The second step optimizes the distance function in PCA and LDA subspace, as well as the scoring strategy when multiple features are stored in the model (cf. sec. 2.2.3). In both cases, we found the $d_{\cos}$ (cf. eq. (9)) similarity function to result in the best EER, though also d_{cor} performs well. The adopted scoring strategies, which gave the best results, differ between $d^{\min}$ for PCA and d^{avg} for LDA, see eqs. (11) and (10), respectively.

Gabor wavelet based face recognition algorithms For the Gabor wavelet based algorithms we do not optimize all configuration parameters of the Gabor wavelet transform. While testing the maximum frequency $k_{\max}$ and the size of the Gabor wavelet σ, we stick to the default values for k_{fac}, $\mu_{\max}$ and $\nu_{\max}$, i. e., keeping the number of Gabor wavelets to be 40.

The parameter optimization for the Gabor graphs algorithm is done in 3 steps. The first step evaluates the above mentioned Gabor wavelet parameters, as well as the employed Gabor jet similarity function. Clearly, the $k_{\max}$ and σ parameters have an influence on how similarity functions perform. For high σ values, e. g., S_C does not perform well, whereas for low σ it is competitive. The overall best configuration uses S_{n+C} with $\sigma = 2\pi$ and $k_{\max} = \pi$. The next two steps try to discover, which node distance and which scoring strategy work best. Apparently, the EER is stable for a wide range of inter-node distances and starts increasing only at 12 pixels. To save memory and computation time in our following experiments we choose the grid distance $g = 6$, though $g = 1$ performs slightly better. Experimentally, the best strategy here is to use the scoring strategy from eq. (28) exploiting the max operator.

The parameter optimization of the LGBPHS algorithm is also divided into 3 steps. The first step evaluates size and overlap of the blocks, from which the local LGBP histograms are extracted. Interestingly, the EER difference between the tested configurations is quite small. We choose the best performing configuration: 4×4 blocks with no overlap. In the second test the Gabor wavelet configuration is optimized. Gabor phases seem to be useful in very few scenarios, but these include the best one. Hence, for LGBPHS we use the optimal $\sigma = \sqrt{2}\pi$ and $k_{\max} = \pi$ and we include histograms of Gabor phases as well. Finally, we try different variants of LBP and different histogram similarity measures. The optimal solution turned out to be the comparison of histograms of non-uniform $\mathrm{LBP}_{8,2}$ patterns with the histogram intersection measure.

Generative face recognition algorithm For the ISV algorithm again 3 configuration parameter optimization steps are executed. First, block size and overlap are evaluated. Larger block sizes and, apparently, the highest possible overlap of blocks perform better, we decide to select block size 10 and overlap 9. In the second test we investigate the number of DCT components to keep and find 45 to be the optimal number. Afterward, we test different numbers of Gaussians and different ISV subspace dimensions. Higher numbers of Gaussians improve the EER, but results are comparably stable with respect to the ISV subspace dimension. The best result was obtained with 768 Gaussians and an ISV subspace of dimension 200.

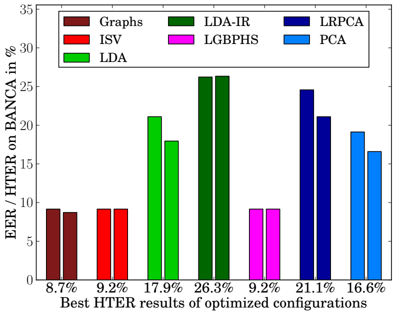


Figure 8: Parameter optimization IV. *This figure summarize the final results of the parameter optimization step of all used face recognition algorithms. Both the equal error rate on the development set and the half total error rate on the evaluation set of the BANCA database are displayed. The HTER results are also given as numbers.*

Summary To summarize the parameter optimization fig. 8 shows the EER on the development set and the HTER on the evaluation set of the BANCA database after optimizing the PCA, LDA, Graphs, LGBPHS and ISV algorithm configurations. The plot is augmented with the results of the LRPCA and the LDA-IR algorithms, which use configurations optimized to another database — the GBU database. Clearly, ISV, LGBPHS and Graphs perform approximately equally well, while the equal error rates of PCA and LDA are around twice as high. Apparently, LRPCA and LDA-IR perform even worse than their “native” counterparts since, obviously, the database matters, to which the configurations are optimized.

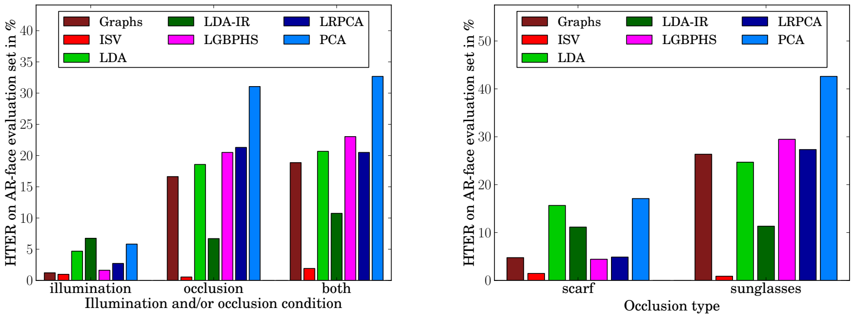
4.2 Face variations

After optimizing configurations of all algorithms we test the algorithms against several variations that influence recognition. For illustration purposes in this section we display the results as graphs and only report the numbers as obtained on the evaluation sets. Exact numbers for the experiments on both development and evaluation set can be found in the online appendix.

4.2.1 Partial occlusions

One aspect of automatic face recognition, especially in surveillance based applications, is the partial occlusion of faces. Two prominent occlusions are sunglasses as they are worn during summer and scarfs covering the lower part of the face during winter. One database that tests exactly these two types of occlusions is the AR face database with its protocol *occlusion*.

Fig. 9(a) contain the results of the occlusion experiments. As a baseline for this database we selected the protocol *illumination*, on which all algorithms perform nicely, only LDA-IR seems to have slight problems with illumination. When occlusions come into play, except for ISV all algorithms suffer a severe drop in performance, independent of whether there is additional non-frontal illumination.



(a) Effect of illumination and occlusion

(b) Effect of different occlusion types

Figure 9: PARTIAL OCCLUSIONS. *This figure shows the effect of partial occlusions of the face on the different face recognition algorithms.*

Having a closer look by separating between the two occlusion types (cf. fig. 9(b)), scarfs and sunglasses seem to have different impacts. While people wearing a scarf, which covers approximately half of the face, can still reasonably well be recognized, sunglasses completely break down the face recognition systems — except for ISV — up to chance level (PCA in fig. 9(b) is close to 50% HTER). These results suggest that the eye region contain most discriminative information, which corresponds to the findings of [66].

4.2.2 Facial expressions

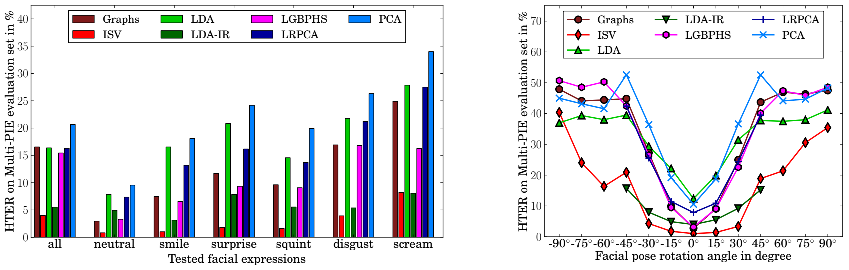
Another aspect an automatic face recognition system must deal with is facial expression. To test the algorithms against various facial expressions we selected the protocol E of the Multi-PIE database.

The results of the expression experiment are shown in fig. 10(a). Interestingly, it can be observed that most algorithms can not handle facial expressions satisfactorily. While neutral faces are recognized quite well by all algorithms, other expressions influence most of the algorithms severely. One exception is ISV since it seems to be stable against mild facial expressions and is still very good in presence of extreme expressions like *disgust* and *scream*. Also LDA-IR handles facial expressions well. On the other hand, all other pixel based algorithms, i. e., PCA, LDA and LRPCA are already unable to cope with mild expressions like *smile* or *surprise*.

4.2.3 Face poses

To use automatic face recognition algorithms in surveillance applications the issue of non-frontal face pose needs to be solved. To test how the algorithms perform on non-frontal images we execute them on protocol P of the Multi-PIE database. Similar to all other protocols we evaluate in this paper, the model enrollment is done using frontal images, while probe images are taken from left profile to right profile in steps of 15°.

The image alignment step uses the hand-labeled eye positions as long as both eyes are visible in the image, i. e., for images with a rotation less or equal to



(a) Facial expressions **(b)** Face poses

Figure 10: Facial expressions and poses. *This figure shows the effect of facial expressions and face pose on the different face recognition algorithms.*

±45°. In the profile and near-profile cases we align images according to the eye and mouth positions, both of which are hand-labeled in the Multi-PIE database. The final positions are $\vec{a}_e^* = (25, 16)^\top$ and $\vec{a}_m^* = (25, 52)^\top$ for eye and mouth in the left profile images, and $\vec{a}_e^* = (38, 16)^\top$ and $\vec{a}_m^* = (38, 52)^\top$ in the right profile. We choose these positions to assure the face including the nose tip to be inside the image, while keeping most of the background outside.

In fig. 10(b) verification performance is plotted for each of the tested poses independently. It can be observed that close-to-frontal poses up to ±15° can be handled by most algorithms, the performance order of the algorithms is similar to what we obtained before. On the other hand, none of the algorithms can handle profile faces, i. e., rotations bigger than ±60°. For rotations higher than ±30° the algorithms that do not need any training, i. e., LGBPHS and Graphs, as well as PCA, LDA and LRPCA have a verification performance around chance level. The only two algorithms that can handle rotations between ±30° and ±60° better are ISV and LDA-IR. Apparently, the color information used in the LDA-IR algorithm is more stable than other features when facial pose is present. In ISV it seems that the underlying statistical model, which treats the extracted local features independently, is robust against moderate pose.

To obtain the results shown in fig. 10(b) the algorithms are trained on all poses. In a similar test, we evaluate if a pose-specific training, i. e., taking the training images of frontal and desired pose only improves performance. The results, which are detailed in the online appendix, show that most algorithms reduce error rates, but often not significantly.

Please note that the results of the LRPCA and LDA-IR algorithms in fig. 10(b) are biased because their training sets do not contain any profile images, i. e., with a rotation higher than ±45°. Since the local regions defined in LRPCA are not visible in near-profile images this algorithm is only evaluated on images with both eyes visible. Similarly, we run LDA-IR experiments only on near-frontal faces since we could not provide the eye and mouth positions, which are required for profile image alignment, to the LDA-IR algorithm.

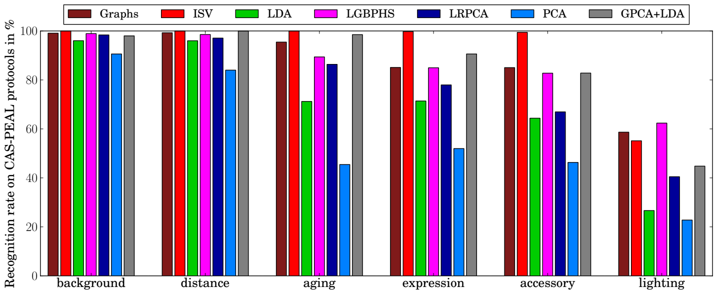


Figure 11: CAS-PEAL. *This figure displays the recognition rates obtained by the tested face recognition algorithms and the GPCA+LDA baseline of [25] on the various protocols of the CAS-PEAL database.*

4.3 Tests on other databases

As the last set of experiments we execute the face recognition algorithms on other publicly available face databases. We evaluate the results using the evaluation methods as proposed by the databases and provide performance figures accordingly. Specifically, we here include those databases that do not provide separate development and evaluation sets. Detailed information on exact values are moved to the online appendix.

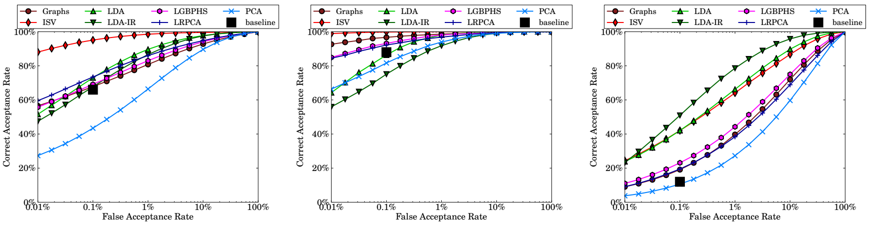
4.3.1 CAS-PEAL

The first database we evaluate our algorithms on is CAS-PEAL, whose default evaluation protocols compute recognition rates. The results of this evaluation are given in fig. 11, where we also include results of the *Gabor feature based PCA+LDA* (GPCA+LDA)[13] algorithm [25]. Since the images provided by the CAS-PEAL database are gray-scale only the LDA-IR algorithm cannot be run.

The results shown in fig. 11 confirm our previous findings. For the simple variations of *background* and *distance*, all systems work close to perfect, except for PCA and LDA. *Aging* seems to be a problem for most of the algorithms, only Graphs, GPCA+LDA and ISV can deal with aging properly. When varying facial *expressions* and *accessories*, nearly all systems drop performance, only ISV seems to be stable against these variations.

The most severe problem in face recognition is the change of illumination. In protocol *lighting* of the CAS-PEAL database not only the directions of light sources are varied, but also the light type changes from ambient light in enrollment images to fluorescent or incandescent light in probe images. This explains the dramatic drop of performance of all systems. Notably, here ISV is no longer the best algorithm, but it is outperformed by LGBPHS and Graphs, both of which rely on Gabor wavelet responses.

[13] In [24] the CAS-PEAL database organizers propose to use LGBPHS, which seems to work better than GPCA+LDA, but they do not provide exact numbers for the experimental results.



(a) 2.0.1 **(b)** 2.0.2 **(c)** 2.0.4

Figure 12: FRGC. *This figure shows ROC curves for experiments 2.0.1, 2.0.2 and 2.0.4 of the FRGC database using mask III, throughout. The FRCG baseline performance of 66%, 88% and 12% CAR at FAR=0.1%, respectively, is marked.*

4.3.2 Face recognition grand challenge

The FRGC database comes with 3 different biased verification protocols. The ROC curves for the algorithms executed on these experiments are presented in fig. 12. The baseline results reported by [71] are also present in the plot as a single marker at 0.1% FAR.

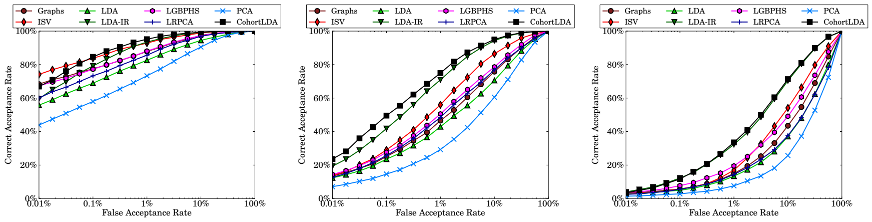
Experiment 2.0.1 compares controlled studio portrait images with each other, using one image for model enrollment and one image for probing. In this scenario, ISV outrivals all other algorithms by far, which all work approximately equally well — besides PCA. Interestingly, the normal LDA outperforms the LDA-IR algorithm in this experiment. This suggests that, extracting color information does not seem to be beneficial when images are taken in controlled environments.

Experiment 2.0.2 tests how well multiple images per person can improve verification performance. The results in fig. 12(b) show that those algorithms that combine different model images in a more sophisticated way, i. e., ISV and Graphs gain a lot in performance, while the other algorithms cannot exploit multiple images that well.

Finally, experiment 2.0.4 uses probe images with uncontrolled illumination. Fig. 12(c) illustrates that the algorithms LDA-IR, LDA and ISV work nicely on this experiment, while all other algorithms perform poorly. Apparently, these are the 3 algorithms that can make use of the biased nature of the protocols during training.

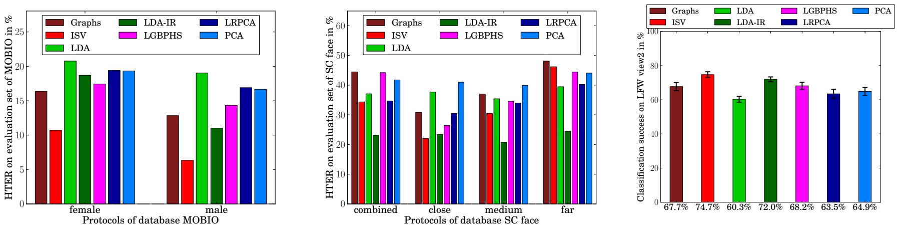
4.3.3 The good, the bad and the ugly

The GBU database provides 3 unbiased verification protocols with increasing difficulty: *Good*, *Bad* and *Ugly*. The ROC curves for all tested algorithms are given in fig. 13. Additionally, we also executed the CohortLDA algorithm, which is LDA-IR including the cohort normalization as originally proposed by the CSU toolkit, as a baseline system. Clearly, most of the algorithms perform comparably well on protocol *Good*, the best two systems are CohortLDA and ISV. For protocols *Bad* and *Ugly*, LDA-IR and CohortLDA are outnumbering all other algorithms, for *Ugly*, ISV even looses its third position to LGBPHS. Interestingly, besides the worst algorithms PCA and LRPCA, also LDA is not working properly in any of the protocols. ISV and LDA performing comparably



(a) Good **(b)** Bad **(c)** Ugly

Figure 13: GBU. *This figure shows ROC curves with logarithmic FAR axis for the Good, the Bad and the Ugly protocols of the GBU database.*



(a) MOBIO **(b)** SC face **(c)** LFW

Figure 14: MOBIO and SC face. *This figure shows HTER's on the evaluation set for all protocols of the databases MOBIO, SC face and LFW.*

bad in the protocols *Bad* and *Ugly* might be due to the fact that the protocols are unbiased and, hence, identities in the training and evaluation set differ.

4.3.4 Mobile biometry

Fig. 14(a) shows the HTER computed on the evaluation set of the MOBIO database for both protocols *female* and *male*. It seems that female clients are more difficult to verify. This finding comply with other face verification experiments performed on this database [97, 32]. As before ISV is able to outperform all other algorithms on both protocols, followed by Graphs and LGBPHS. Again, PCA, LRPCA and LDA obtain the highest HTER. Apparently, LDA performs even worse than PCA, which might be explained by the small number of clients in the training set. Interestingly, LDA-IR is the second best algorithm for *male*, but not for *female*.

4.3.5 Surveillance camera face database

Probably most difficult is the SC face database. The HTER results for the four different protocols are depicted in fig. 14(b). Disastrously, only two algorithms are able to beat the 30% mark and no algorithm is better than 20% HTER. The best algorithm on the SC face database is LDA-IR. Especially in the *far* condition the original images are very small and it seems that, besides color information, no useful features can be extracted from these images any more, even when the images are up-scaled to the default image resolution of 64 × 80 pixels. For images of this small resolution, more advanced techniques as the simple bi-linear interpolation, e. g., as proposed by [4] need to be used to enhance the content of the images.

4.3.6 Labeled faces in the wild

The last database, on which we test our algorithms, is the *labeled faces in the wild* (LFW) database. Fig. 14(c) displays the average classification rates as well as the standard deviations over the 10 different folds of *view 2* as required by the LFW protocol [41]. With 74.7% average classification accuracy ISV performs best on this database, tightly followed by LDA-IR. Since the LFW protocol is unbiased one can clearly see that the LDA algorithm is performing worse than PCA, in this case LDA is once more the worst algorithm of them all. Also, the local region split of LRPCA does not seem to be helpful. Due to uncontrolled head poses and facial expressions the defined local regions might not have captured the local features they aim at.

5 Discussion

After executing all these experiments and showing the identification and verification performances of the algorithms under several conditions and for various image databases, we want to discuss other properties of the algorithms. Additionally, in this section we try to highlight the contribution of this paper and discuss what we might have missed.

5.1 Properties of the algorithms

5.1.1 Algorithm execution time

Tab. 1(a) contains the execution times that were measured in a test run on the BANCA database. Particularly, the extractor, projector and enroller training is executed using 300 training files, while feature extraction and projection are performed on 6020 images. During enrollment 52 client models are generated, each using the features of 5 images. Finally, in the scoring step 5460 scores are computed. In any case, we do not take into account the time for accessing the data on hard disk, but we only measure the real execution time of the algorithms. Hence, the actual processing time might increase due to hard disk or network latencies.

The algorithm configuration in the experiments is identical to the one after optimizing the configurations. Apparently, the execution time of the algorithms differ a lot. For the simple pixel-based algorithms PCA, LDA, LRPCA and LDA-IR the projector training is done in a couple of seconds, while the feature projection takes most of the time, here around one minute. Enrollment is almost instantaneous since it just needs to store all features, and the scoring is also very fast. Note that the LRPCA and LDA-IR algorithms are solely implemented in Python, whereas PCA and LDA used the mixed C++/Python implementation as given in Bob [7].

The extraction of Gabor graphs and the enrollment of the client models is nearly as fast as the feature extraction/projection in the PCA based meth-

Algorithm	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
Extraction	1.9 m	9.7 m	—	—	8.5 h	—	—
Projector training	—	1.8 h	1.9 s	5.7 s	—	8.8 s	0.8 s
Projection	—	4.8 h	4.5 s	1.4 m	—	1.4 m	45.7 s
Enrollment	0.7 s	1.4 m	0.8 s	1.2 s	2.4 m	1.6 s	1.1 s
Scoring	21.7 s	11.6 s	2.8 s	3.7 s	23.3 s	9.9 s	3.1 s
total	2.3 m	6.7 h	10.0 s	1.6 m	8.6 h	1.7 m	50.8 s

(a) Execution Time

Algorithm	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
Projector size	—	55 MB	1.2 MB	2.3 MB	—	77 MB	12 MB
Feature size	81 kB	1.3 MB	40 kB	1.8 kB	≈3 MB	27.3 kB	40 kB
Projected size	—	800 kB	232 B	—	—	—	2.3 kB
Model size	406 kB	264 kB	1.1 kB	9 kB	≈9 MB	400 kB	11.7 kB

(b) Memory Requirements

Table 1: Time and memory properties. *This table gives an overview of the execution time that specific parts of the algorithms need and the size of the produced elements in memory. The machining times are measured on a 3.4 GHz Intel i7 processor with 16 GB of RAM, executing both development and evaluation set of the BANCA database.*

ods. Only the scoring takes more time since computing the similarity measures require a higher computational effort.

The LGBPHS feature extraction needs a huge amount of time. This can be explained by the fact that the features themselves are huge and, hence, we chose a compressed format to store the histograms. This decreases the size of the LGBPHS feature vector (though tab. 1(b) shows that LGBPHS features still are longest), but increase the extraction and also the enrollment time a lot. Also the scoring time is affected.

The longest training and projection time is needed by ISV. During training the distribution of the mixture of Gaussians and the U subspace of the ISV algorithm are estimated — both procedures rely on computationally intensive iterative processes. Furthermore, the long projection time can be explained by its complexity, where sufficient statistics of the samples given the Gaussian mixture model are first computed, before being used to estimate session offsets. Finally, the scoring time is comparably short since most of the time consuming estimations are cached in the projection and enrollment steps.

5.1.2 Memory requirements

Tab. 1(b) displays the memory requirements of the objects produced during the execution of the algorithms. All elements are stored in double precision, i. e., with 8 bytes for each number.

For PCA-related algorithms, these objects are the projection matrix and the sizes of the extracted features and the enrolled models. For LRPCA and LDA-IR these values are estimates since the format, which is stored, is unknown.[14]

[14] We just use the `pickle` module of Python to store the LRPCA and LDA-IR data. Tab. 1(b) shows the resulting file size on disk.

Depending on the complexity of the algorithms, the size of the features and models differ slightly. In any case, the projection matrix needs to be stored to be able to use these technologies in a real word application, which might be problematic on devices with limited memory. Notably, the lowest memory consumption is achieved by the LDA algorithm, which, e.g., is able to use a projected feature vector of only 29 dimensions, i. e., 232 Bytes.

The size of the Gabor graphs is also relatively small. An enrolled model that stores 5 feature vectors is approximately of the same size of the LRPCA model. For LGBPHS, the feature and model sizes are much higher. Please note that the sizes of the LGBPHS feature vectors and client models differ slightly because we use a compressed format to store the histograms. Still, feature vectors and models of this size makes it difficult to use this algorithm in a real world application, at least with the configuration that we optimized.

Finally, the size of the ISV projector and the projected features are comparably high, while the client model is relatively small. This is an advantage over having large models since in a face recognition application, usually many models are stored, but only few probe images need to be processed at a time.

5.1.3 Algorithm complexity

From the researchers view point it is also important to analyze the complexity of face recognition algorithms. Even though algorithms may be very good in terms of recognition or verification performance, if they are hard to implement — and there is no open source implementation of this algorithm available — only few researchers will build upon these algorithms.

The definitely most simple algorithm is PCA. Utilizing linear algebra packages like `LAPACK` in C++ or `scipy` in python, a working PCA-based face recognition system can be implemented in a couple of hours. The feature extraction and projection are very simple and easy to understand. Also the comparison of model and probes is using simple distance functions, which can be implemented quickly or taken from existing libraries. Once having PCA, LDA can be coded in a similar amount of time. Also the implementation of LRPCA and LDA-IR can be done quickly, but it might be cumbersome to define local regions or color layers.

The LGBPHS algorithm is relatively straightforward to implement. The probably most difficult part is the Gabor wavelet transform. Unfortunately, there has been no open source code[15] available to perform the Gabor wavelet transform in a fast and explicit manner. Since 2012, Bob [7] contains open source code to execute a Gabor wavelet transform, which is optimized and standardized. The subsequent LBP feature extraction, the image block separation, the histogram collection and the histogram comparisons are again simple.

The Gabor graphs algorithm as we use it in our experiments is more complicated to implement. The Gabor graph extraction is, besides the already mentioned Gabor wavelet transform, easy to be done. The more difficult part of the algorithm is the comparison between model and probe features. While traditional and simple implementations use only the absolute values of the Gabor wavelet responses, the more advanced methods that include Gabor phases are much more difficult to understand and to implement.

[15] Some people published MatLab code that performs Gabor wavelet transform, though.

Of the tested algorithms ISV is definitely most complex. While all other algorithms in this paper are discriminative, ISV is a generative algorithm. Researchers, who are not familiar with generative algorithms, might have larger problems to understand the specific details required for implementation. The feature extraction step, i. e., cutting the image into blocks and extracting a bag of DCT block features from them is the easiest part of the algorithm. An efficient implementation of Gaussian mixture modeling can be difficult and error prone. Usually, the GMM training includes two steps: a k-means-computation and a density estimation of the mixture of Gaussians, which is sophisticated. The subspace of the ISV model also relies on an iterative procedure. This method alternately estimates the latent variables associated to each training sample given the current estimate of the ISV subspace, and this subspace given the current values of the latent variables.

Unfortunately, the most successful algorithm is also the most complex and complicated one. Obviously, we cannot help researchers in understanding the algorithm, but we do provide the source code for the algorithm (and, of course, also for all other algorithms) so that at least the error-prone implementation of the algorithm does not need to be re-done.

5.2 About this evaluation

Of course, an evaluative survey of face recognition algorithms as we provide with this paper cannot cover the full range of all possible face recognition algorithms including all their variations, and we might have omitted some aspects of face recognition. We know that this paper does not answer the question: What is the best face recognition algorithm? Nonetheless, we hope to provide some insights about advantages and drawbacks of the algorithms that we tested and also some hints, which algorithms are well suited under different circumstances.

5.2.1 What we missed

Though we could not test all face recognition algorithms we tried to find a good compromise, which algorithms to test and which to leave out, and we are sorry if we do not evaluate the algorithm of your choice. Also, we executed algorithms only like they are reported in literature. Theoretically, we could have tried PCA or LDA on DCT features or ISV modeling of Gabor features, etc., the range of possible tests is unlimited.

One face recognition approach that we left out completely is the face recognition from geometrical features [47, 17]. As [44] pointed out, all of these algorithms are outdated and most of them work well only with hand-labeled feature points. Using automatically detected node positions the recognition performance of geometrically based algorithms drop dramatically [31]. Additionally, no open source implementation of these algorithms are available. Since we do not want to implement outdated algorithms with low recognition accuracy we do not perform any experiments with geometrical features.

Another aspect of face recognition is score normalization. For example, *ZT-norm* [8] has been shown lately [97] to be very effective and able to improve face verification drastically. *Score calibration* [13] can not improve face recognition in the first place, but it can make scores from different systems comparable and, thus, simplify fusion of these systems. In this work, we do not perform any score normalization or calibration, and no fusion system is studied.

We tried to make the comparison of the face recognition systems as fair as possible. We optimized the configurations of most algorithms to a certain image database. Only the LRPCA and LDA-IR algorithms were optimized to another database [69, 53] and we did not touch this configurations in our experiments. This biases the algorithms towards different image variations, but still we think we could show the trends of the algorithms. Also, the optimization was done in several steps using discrete sets of configuration parameters. A joint optimization strategy with continuous parameters could have resulted in a slightly better performance on BANCA.

We intentionally optimized the configurations on one database and kept the configurations stable during all subsequent tests. Therefore, the results on the other databases are not optimal. For example, in FRGC the PCA algorithm could not reach the baseline performance, which was computed with an eigenface algorithm as well. Also, the Graphs algorithm can obtain better results on the CAS-PEAL protocol *lighting* as we showed in [33] when the configuration is optimized towards that database, or ISV can work better on SC face [96].

5.2.2 What we achieved

Nevertheless, the contribution of this paper is — to our knowledge — unique. We perform the first extensive and extensible evaluative survey of face recognition algorithms that is completely based on open source software[16] and freely available tools and packages and no additional commercial software needs to be bought to run the experiments. All experiments can be rerun and all results (including the figures and tables from this paper) can be regenerated by other researchers by invoking just a short sequence of commands, which are documented in the software package. Using these commands we execute several state-of-the-art face recognition algorithms, optimize their configurations and test them on various image databases using their default evaluation protocols, which contained different image variations like illumination, expression, pose and occlusion.

Since the implementation of the evaluation protocols is time consuming and error prone, many researchers rely on results generated on small image databases using their own protocols, which makes their results incomparable to the results of other researchers [89, 84]. In the source code that we provide [7, 34] evaluation protocols for several image databases are already implemented, and changing the database or the protocol is as easy as changing one command line parameter. Additionally, the same software package also allows to prototype new ideas, test combinations of these ideas with existing code, run face recognition experiments and evaluate the results of these experiments. Since the evaluation is always identical, results are always directly comparable.

With this software package we want to encourage researchers to run face recognition experiments in a comparable way. Using Python and the *Python Package Index* (PyPI) it is easily possible for researchers to provide their source code for interested people to regenerate their results. A nice side effect of publishing source code together with scientific paper lies in the fact [93] that

[16] The software will be released as soon as this paper is accepted for publication. Please send a message to manuel.guenther@idiap.ch for a sneak preview of the library.

papers with source code are cited on average 5 times more than papers without. The software package that we distribute with this paper is one example of how to provide source code and reproducible results to other researchers.

5.2.3 What we found

We have tested several state-of-the-art face recognition algorithms on several image databases and with different image variations. In most of the tests we have found that:

1. ISV, the generative approach that models a face as the distribution of facial features, outperforms the other algorithms, sometimes by far, as long as the illumination conditions are not too difficult. Unfortunately, this algorithm needs quite a long time for the (offline) training and model enrollment, and also for the (online) feature extraction.

2. Color information, as used by LDA-IR, can be very helpful, especially when the texture itself is degraded due to low resolution, difficult facial expressions, occlusions or pose.

3. Image preprocessing plays an important role, and each face recognition algorithm has its own preferred preprocessing. Sometimes, the best preprocessing even changes from database to database. Interestingly, algorithms work with many image resolution — as far as it exceeds a lower limit of approximately 16 pixels inter-eye-distance.

4. In general, raw pixel values are not well suited as features for face recognition. During the preprocessing tests, the pixel-based algorithms PCA, LDA and LRPCA clearly preferred the LBP-based preprocessing, which can be seen as a kind of feature extraction. Though these algorithms usually can be trained fast and also the score computation is efficient a different kind of feature might help to improve their recognition capabilities.

5. Gabor wavelet based algorithms in general perform better than pixel based algorithms. Especially, images with strong or uncontrolled illumination conditions are handled better by algorithms using Gabor wavelets. Furthermore, a proper use of Gabor phases improves the performance of these algorithms. In this study, we used two state-of-the-art methods that do not include any training. We assume that these methods can be improved by incorporating (learning) knowledge from the training set.

6. None of the algorithms is able to handle non-frontal pose, even if they have seen all poses during training. The direct comparison of features from different poses seems not to be possible with the discriminative algorithms, and even the generative algorithm still has a lot of problems. Hence, we believe that different kinds of methods need to be implemented, e. g., [63] showed a promising approach to that problem.

7. When multiple files are available for model enrollment or probing, the ISV algorithm, which directly uses multiple images, and the Graphs algorithm, which used a local scoring strategy, are able to exploit these data better

than the other algorithm that use only simple scoring strategies like taking the minimum distance or maximum similarity. In general, incorporating several images in the verification process improves performance, leading to the use of videos for enrollment and probing.

8. Biased evaluation protocols, where identities or even images are shared between training set and development/evaluation sets, favor the algorithms that make use of the identity information during training, like LDA and ISV (see also [53] on the effect of biased protocols on the LDA-IR algorithm). In unbiased protocols these algorithms perform much worse, sometimes LDA is even outperformed by the simple PCA algorithm.

6 Conclusion

In this paper we presented the first evaluative and reproducible study of open source face recognition algorithms. First, we briefly described different kinds of state-of-the-art and popular face recognition algorithms including several image preprocessing strategies. We selected a representative set of algorithms with different approaches and executed a list of face recognition experiments on them. Most of the algorithms were taken from the open source software library Bob [7], while two algorithms stem from the Colorado State University toolkit [69, 53].

The first evaluation that we performed was assessing, which image resolution is required for the different algorithms to run properly. After selecting a proper image resolution, we evaluated the performance of the algorithms under several different image preprocessing strategies on some difficult image databases and selected the most appropriate preprocessing for each face recognition algorithm. Subsequently, we optimized the configurations of most algorithms to the BANCA database, leaving the already optimized configurations of the CSU algorithms untouched. We tested the algorithm performance with regard to different image variations like facial expressions, partial occlusions and non-frontal poses. Afterward, we selected a number of challenging databases and ran the algorithms on them. Hence, this paper provides reference implementations and reference results for many image databases. Finally, we discussed a number of attributes of the algorithms that might limit their usability in real world applications.

A short summary of the evaluation could be that there is not a single algorithm that works best in all cases and for all applications. Nevertheless, there are some favorites. The simple PCA algorithm performed worst in basically every test we did. LDA works quite well on biased protocols, but in unbiased protocols it reaches the same level of performance as PCA. On most tests, the local region PCA algorithm performed only slightly better than PCA. Gabor wavelet based algorithms are well suited in difficult illumination conditions and were average in the other tests we performed. Still there is room for improvement of these algorithms since the ones we have tested in this paper did not make use of the training set. The only algorithm in our test that used color information, i. e., LDA-IR works very well under several circumstances, especially when the image conditions are rather poor and algorithms cannot rely on facial features any more. The generative algorithm ISV performed best in most of the tests, but has the drawback of a very long execution time and high memory usage and cannot be used, e. g., in mobile devices with limited capacities and real-time demands.

Though we have tested only a small subset of pixel based algorithms, the results suggest that PCA and LDA algorithms that are based on raw pixel values are limited and not appropriate anymore to new challenging problems characterized by large and uncontrolled databases such as labeled faces in the wild, MOBIO, the SC face database and others, for which new research directions need to be developed.

One important aspect of this evaluation is that we provide the source code for each of the experiments, including all image database interfaces, all preprocessings, all feature extractors, all recognition algorithms and all evaluation scripts. Therefore, all experiments can be rerun and all figures can be recreated by anybody that has access to the raw image data. Additionally, we want to encourage other researchers to use our source code to run their own face recognition experiments since the software is designed to be easy to handle, easy to extend and to produce comparable results. We furthermore want to animate researchers to publish the source code of their algorithms.

Acknowlegdments

This evaluation has received funding from the European Community's FP7 under grant agreements 238803 (BBfor2: `bbfor2.net`) and 284989 (BEAT: `beat-eu.org`).
This work is based on open source software provided by the Idiap Research Institute and the Colorado State University. The authors want to thank all contributors of the software for their great work.

References

[1] A.F. Abate, M. Nappi, D. Riccio, and G. Sabatino. 2d and 3d face recognition: A survey. *Pattern Recognition Letters*, 28(14):1885–1906, 2007.

[2] T. Ahonen, A. Hadid, and M. Pietikainen. Face recognition with local binary patterns. In *ECCV*, pages 469–481. Springer, 2004.

[3] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 28(12):2037–2041, 2006.

[4] N. Al-Hassan, S.A. Jassim, and H. Sellahewa. Construction of dictionaries to reconstruct high-resolution images for face recognition. In *ICB*. Springer, 2013.

[5] T. Ali, L.J. Spreeuwers, and R.N.J. Veldhuis. Forensic face recognition: A survey. In *Face Recognition: Methods, Applications and Technology*, Computer Science, Technology and Applications, pages 187–200. Nova Publishers, 2012.

[6] M. Aly. Face recognition using SIFT features. Technical report, California Institute of Technology, 2006.

[7] 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 *ACM-MM*, pages 1449–1452. ACM press, 2012.

[8] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing*, 10(1):42–54, 2000.

[9] E. Bailly-Baillière et al. The BANCA database and evaluation protocol. In *AVBPA*, volume 2688 of *LNCS*, pages 625–638. SPIE, 2003.

[10] M.S. Bartlett, J.R. Movellan, and T.J. Sejnowski. Face recognition by independent component analysis. *Transactions on Neural Networks*, 13(6):1450–1464, 2002.

[11] D.M. Blackburn, M. Bone, and P.J. Phillips. *Face Recognition Vendor Test 2000: Evaluation Report*. Storming Media, 2001.

[12] D.S. Bolme, J.R. Beveridge, M. Teixeira, and B.A. Draper. The CSU face identification evaluation system: its purpose, features, and structure. In *ICVS*, pages 304–313. Springer, 2003.

[13] N. Brümmer. *Measuring, refining and calibrating speaker and language information extracted from speech*. PhD thesis, Stellenbosch University, Netherlands, 2010.

[14] F. Cardinaux, C. Sanderson, and S. Bengio. User authentication via adapted statistical models of face images. *Transactions on Signal Processing*, 54(1):361–373, 2006.

[15] F. Cardinaux, C. Sanderson, and S. Marcel. Comparison of MLP and GMM classifiers for face verification on xm2vts. In *AVBPA*, volume 2688 of *LNCS*, pages 911–920. Springer, 2003.

[16] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546. IEEE Computer Society, 2005.

[17] I.J. Cox, J. Ghosn, and P.N. Yianilos. Feature-based face recognition using mixture-distance. In *CVPR*, pages 209–216. IEEE Computer Society, 1996.

[18] J.G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[19] K. Delac, M. Grgic, and S. Grgic. Statistics in face recognition: analyzing probability distributions of PCA, ICA and LDA performance results. In *ISPA*, LNCS, pages 289–294. Springer, 2005.

[20] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, 39(1):1–38, 1977.

[21] L. El Shafey, C. McCool, R. Wallace, and S. Marcel. A scalable formulation of probabilistic linear discriminant analysis: Applied to face recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1788–1794, 2013.

[22] K. Etemad and R. Chellappa. Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A*, 14:1724–1733, 1997.

[23] M. Everingham, J. Sivic, and A. Zisserman. “Hello! My name is... Buffy” – automatic naming of characters in TV video. In *BMVC*, pages 92.1–92.10. BMVA, 2006.

[24] W. Gao, B. Cao, S. Shan, X. Chen, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. *Systems, Man and Cybernetics, Part A: Systems and Humans*, 38:149–161, 2008.

[25] W. Gao, B. Cao, S. Shan, D. Zhou, X. Zhang, and D. Zhao. The CAS-PEAL large-scale Chinese face database and baseline evaluations. Technical report, Joint Research & Development Laboratory for Face Recognition, Chinese Academy of Sciences, 2004.

[26] J.-L. Gauvain and C.-H. Lee. Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains. *Transactions on Speech and Audio Processing*, 2(2):291–298, 1994.

[27] C. Geng and X. Jiang. Face recognition using SIFT features. In *ICIP*, pages 3313–3316. IEEE Signal Processing Society, 2009.

[28] M. Grgic, K. Delac, and S. Grgic. SCface–surveillance cameras face database. *Multimedia Tools and Applications*, 51(3):863–879, 2011.

[29] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 2010.

[30] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, pages 498–505. IEEE, 2009.

[31] M. Günther. *Statistical Gabor Graph Based Techniques for the Detection, Recognition, Classification, and Visualization of Human Faces*. PhD thesis, Institut für Neuroinformatik, Technische Universität Ilmenau, Germany, 2011.

[32] M. Günther et al. The 2013 face recognition evaluation in mobile environment. In *ICB*, 2013.

[33] M. Günther, D. Haufe, and R.P. Würtz. Face recognition with disparity corrected Gabor phase differences. In *ICANN*, volume 7552 of *LNCS*, pages 411–418. Springer, 2012.

[34] M. Günther, R. Wallace, and S. Marcel. An open source framework for standardized comparisons of face recognition algorithms. In *ECCV. Workshops and Demonstrations*, volume 7585 of *LNCS*, pages 547–556. Springer, 2012.

[35] M. Günther and R.P. Würtz. Face detection and recognition using maximum likelihood classifiers on Gabor graphs. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(3):433–461, 2009.

[36] A. Hadid, M. Pietikäinen, and T. Ahonen. A discriminative feature space for detecting and recognizing faces. In *CVPR*, pages 797–804. Springer, 2004.

[37] D. Haufe. Einfluss der Bildauflösung auf Gesichtserkennung durch Graphenvergleich. BSc thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, 2008.

[38] G. Heusch, Y. Rodriguez, and S. Marcel. Local binary patterns as an image preprocessing for face authentication. In *FG*, pages 9–14. IEEE Computer Society, 2006.

[39] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen. Local binary patterns and its application to facial image analysis : A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 41(6):765–781, 2011.

[40] G.B. Huang, V. Jain, and E.G. Learned-Miller. Unsupervised joint alignment of complex images. In *ICCV*, pages 1–8. IEEE, 2007.

[41] G.B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.

[42] R. Huang, V. Pavlovic, and D.N. Metaxas. A hybrid face recognition method using markov random fields. In *ICPR*, pages 157–160. IEEE, 2004.

[43] International Civil Aviation Organization (ICAO). Machine Readable Travel Documents (doc 9303). `http://www.icao.int/Security/mrtd/Pages/default.aspx`, 2006. Part 1, Vol. 2.

[44] R. Jafri and H.R. Arabnia. A survey of face recognition techniques. *Journal of Information Processing Systems*, 5(2):41–68, 2009.

[45] A.K. Jain and S.Z. Li. *Handbook of Face Recognition*. Springer, 2005.

[46] D. González Jiménez, M. Bicego, J.W.H. Tangelder, B.A.M. Schouten, O.O. Ambekar, J. Alba-Castro, E. Grosso, and M. Tistarelli. Distance measures for Gabor jets-based face authentication: A comparative evaluation. In *ICB*, pages 474–483. Springer, 2007.

[47] T. Kanade. *Picture Processing System by Computer Complex and Recognition of Human Faces*. PhD thesis, Kyoto University, Japan, 1973.

[48] J. Kittler, Y. Li, and J. Matas. On matching scores for LDA-based face verification. In *BMVC*, pages 5.1–5.10. BMVA, 2000.

[49] C. Liu. Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance. *Transactions on Pattern Analysis and Machine Intelligence*, 28:725–737, 2006.

[50] Q. Liu, H. Lu, and S. Ma. Improving kernel Fisher discriminant analysis for face recognition. *Transactions on Circuits and Systems for Video Technology*, 14(1):42–49, 2004.

[51] W. Liu, Y. Wang, S.Z. Li, and T. Tan. Null space approach of Fisher discriminant analysis for face recognition. In *ECCV, Biometric Authentication Workshop*, pages 32–44. Springer, 2004.

[52] S. Lucey and T. Chen. A GMM parts based face representation for improved verification through relevance adaptation. In *CVPR*, pages 855–861. IEEE Computer Society, 2004.

[53] Y.M. Lui, D.S. Bolme, P.J. Phillips, J.R. Beveridge, and B.A. Draper. Preliminary studies on the good, the bad, and the ugly face recognition challenge problem. In *CVPR Workshops*, pages 9–16. IEEE Computer Society, 2012.

[54] S. Marcel et al. On the results of the first mobile biometry (MOBIO) face and speaker verification evaluation. In *Recognizing patterns in signals, speech, images, and videos*, ICPR, pages 210–225. IEEE, 2010.

[55] A. Martínez and R. Benavente. The AR face database. Technical Report 24, Computer Vision Center, 1998.

[56] C. McCool et al. Bi-modal person recognition on a mobile phone: using mobile phone data. In *ICME Workshop on Hot Topics in Mobile Multimedia*, pages 635–640. IEEE Computer Society, 2012.

[57] C. McCool and S. Marcel. Parts-based face verification using local frequency bands. In *ICB*, pages 259–268. Springer, 2009.

[58] C. McCool, R. Wallace, M. McLaren, L. El Shafey, and S. Marcel. Session variability modelling for face authentication. *IET Biometrics*, 2013.

[59] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *AVBPA*, pages 72–77. LNCS, 1999.

[60] B. Moghaddam. Principal manifolds and probabilistic subspaces for visual recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 24:780–788, 2002.

[61] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV*, pages 786–793. IEEE, 1995.

[62] B. Moghaddam, W. Wahid, and A. Pentland. Beyond eigenfaces: Probabilistic matching for face recognition. In *FG*, pages 30–35. IEEE Computer Society, 1998.

[63] M.K. Müller, M. Tremer, C. Bodenstein, and R.P. Würtz. Learning invariant face recognition from examples. *Neural Networks*, 41:137–146, 2013.

[64] H.V. Nguyen, L. Bai, and L. Shen. Local Gabor binary pattern whitened PCA: A novel approach for face recognition from single image per person. In *Advances in Biometrics*, volume 5558 of *LNCS*, pages 269–278. Springer, 2009.

[65] V.D.M. Nhat and S. Lee. An improvement on PCA algorithm for face recognition. In *ISNN*, LNCS, pages 1016–1021. Springer, 2005.

[66] O. Ocegueda, S.K. Shah, and I.A. Kakadiaris. Which parts of the face give out your identity? In *CVPR*, pages 641–648. IEEE Computer Society, 2011.

[67] T. Ojala, M. Pietikäinen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996.

[68] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[69] P.J. Phillips, J.R. Beveridge, B.A. Draper, G. Givens, A.J. O'Toole, D.S. Bolme, J. Dunlop, Yui Man Lui, H. Sahibzada, and S. Weimer. An introduction to the good, the bad, & the ugly face recognition challenge problem. In *FG*, pages 346–353. IEEE Computer Society, 2011.

[70] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, Jin Chang, K. Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. In *CVPR*, volume 1, pages 947–954. IEEE Computer Society, 2005.

[71] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, and W. Worek. Preliminary face recognition grand challenge results. In *FG*, pages 15–24. IEEE Computer Society, 2006.

[72] P.J. Phillips, P. Grother, R. Micheals, D. Blackburn, E. Tabassi, and M. Bone. Face recognition vendor test 2002: Evaluation report. Technical report, NIST, 2003.

[73] P.J. Phillips, P. Rauss, and S.Z. Der. FERET (face recognition technology) recognition algorithm development and test results. Technical report, Army Research Lab, 1996.

[74] P.J. Phillips, T. Scruggs, A.J. O'Toole, P.J. Flynn, K.W. Bowyer, C.L. Schott, and M. Sharpe. FRVT 2006 and ICE 2006 large-scale results. Technical report, NIST, 2007.

[75] S.J.D. Prince and J.H. Elder. Probabilistic linear discriminant analysis for inferences about identity. In *ICCV*, pages 1–8. IEEE, 2007.

[76] K. Ramírez-Gutiérrez, D. Cruz-Pérez, and H. Pérez-Meana. Face recognition and verification using histogram equalization. In *ACS*, pages 85–89. WSEAS, 2010.

[77] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[78] Y. Rodriguez and S. Marcel. Face authentication using adapted local binary pattern histograms. In *ECCV*, pages 321–332. Springer, 2006.

[79] H.A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *CVPR*, pages 38–44. Springer, 1998.

[80] M. Sadeghi and J. Kittler. Data fusion in face verification. In *COST Workshop, Biometrics on the Internet: Fundamentals, Advances and Applications*, pages 61–66, 2004.

[81] F.S. Samaria and A.C. Hartert. Parameterisation of a stochastic model for human face identification. In *WACV*, pages 138–142. IEEE Computer Society, 1994.

[82] C. Sanderson and K.K. Paliwal. Fast features for face authentication under illumination direction changes. *Pattern Recognition Letters*, 24(14):2409–2419, 2003.

[83] Á. Serrano, I. Martín de Diego, C. Conde, and E. Cabello. Recent advances in face biometrics with Gabor wavelets: A review. *Pattern Recognition Letters*, 31(5):372–381, 2010.

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

[85] L. Shen, L. Bai, and M.C. Fairhurst. Gabor wavelets and general discriminant analysis for face identification and verification. *Image and Vision Computing*, 25(5):553–563, 2010.

[86] H. Shin, S.-D. Kim, and H.-C. Choi. Generalized elastic graph matching for face recognition. *Pattern Recognition Letters*, 28(9):1077–1082, 2007.

[87] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America A*, 4(3):519–524, 1987.

[88] B. Stratmann. Einfluss der Graphenstruktur auf die Leistung eines Gesichtserkennungssystems. BSc thesis, Institut für Neuroinformatik, Ruhr-Universität Bochum, Germany, 2010.

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

[90] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Transactions on Image Processing*, 19(6):1635–1650, 2010.

[91] M.L. Teixeira. The Bayesian intrapersonal/extrapersonal classifier. Master's thesis, Colorado State University, 2003.

[92] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[93] P. Vandewalle. Code sharing is associated with research impact in image processing. *Computing in Science and Engineering*, 14(4):42–47, 2012.

[94] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137–154, 2002.

[95] R.J. Vogt and S. Sridharan. Explicit modelling of session variability for speaker verification. *Computer Speech & Language*, 22(1):17–38, 2008.

[96] R. Wallace, M. McLaren, C. McCool, and S. Marcel. Inter-session variability modelling and joint factor analysis for face authentication. In *IJCB*, pages 1–8. IEEE, 2011.

[97] R. Wallace, M. McLaren, C. McCool, and S. Marcel. Cross-pollination of normalization techniques from speaker to face authentication using Gaussian mixture models. *Transactions on Information Forensics and Security*, 7(2):553–562, 2012.

[98] H. Wang, S.Z. Li, and Y. Wang. Face recognition under varying lighting conditions using self quotient image. In *FG*, pages 819–824. IEEE Computer Society, 2004.

[99] L. Wiskott, J.-M. Fellous, N. Krüger, and C. v.d. Malsburg. Face recognition by elastic bunch graph matching. *Transactions on Pattern Analysis and Machine Intelligence*, 19:775–779, 1997.

[100] L. Wiskott, J.-M. Fellous, N. Krüger, and C. v.d. Malsburg. Face recognition by elastic bunch graph matching. In *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, chapter 11, pages 355–396. CRC, 1999.

[101] Y. Wong, S. Chen, S. Mau, C. Sanderson, and B.C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *CVPR Workshops*, pages 74–81. IEEE Computer Society, 2011.

[102] R.P. Würtz. *Multilayer Dynamic Link Networks for Establishing Image Point Correspondences and Visual Object Recognition.* PhD thesis, Fakultät für Physik und Astronomie, Ruhr-Universität Bochum, Germany, 1994.

[103] W.S. Yambor. Analysis of PCA-based and Fisher discriminant-based image recognition algorithms. Master's thesis, Colorado State University, 2000.

[104] J. Yang, D. Zhang, A.F. Frangi, and J. Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *Transactions on Pattern Recognition and Machine Intelligence*, 26(1):131–137, 2004.

[105] P. Yang, S. Shan, W. Gao, S.Z. Li, and D. Zhang. Face recognition using Ada-boosted Gabor features. In *FG*, pages 356–361. IEEE Computer Society, 2004.

[106] B. Zhang, S. Shan, X. Chen, and W. Gao. Histogram of gabor phase patterns (HGPP): A novel object representation approach for face recognition. *Transactions on Image Processing*, 16(1):57–68, 2007.

[107] L. Zhang, R. Chu, S. Xiang, S. Liao, and S.Z. Li. Face detection based on multi-block lbp representation. In *ICB*, pages 11–18. Springer, 2007.

[108] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In *ICCV*, volume 1, pages 786–791. IEEE Computer Society, 2005.

[109] W. Zhang, S. Shan, L. Qing, X. Chen, and W. Gao. Are Gabor phases really useless for face recognition? *Pattern Analysis & Applications*, 12:301–307, 2009.

[110] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

[111] W. Zhao, A. Krishnaswamy, R. Chellappa, D.L. Swets, and J. Weng. Discriminant analysis of principal components for face recognition. In *Face Recognition: From Theory to Applications*, pages 73–85. Springer, 1998.

A Additional figures

A.1 Face recognition tool chain

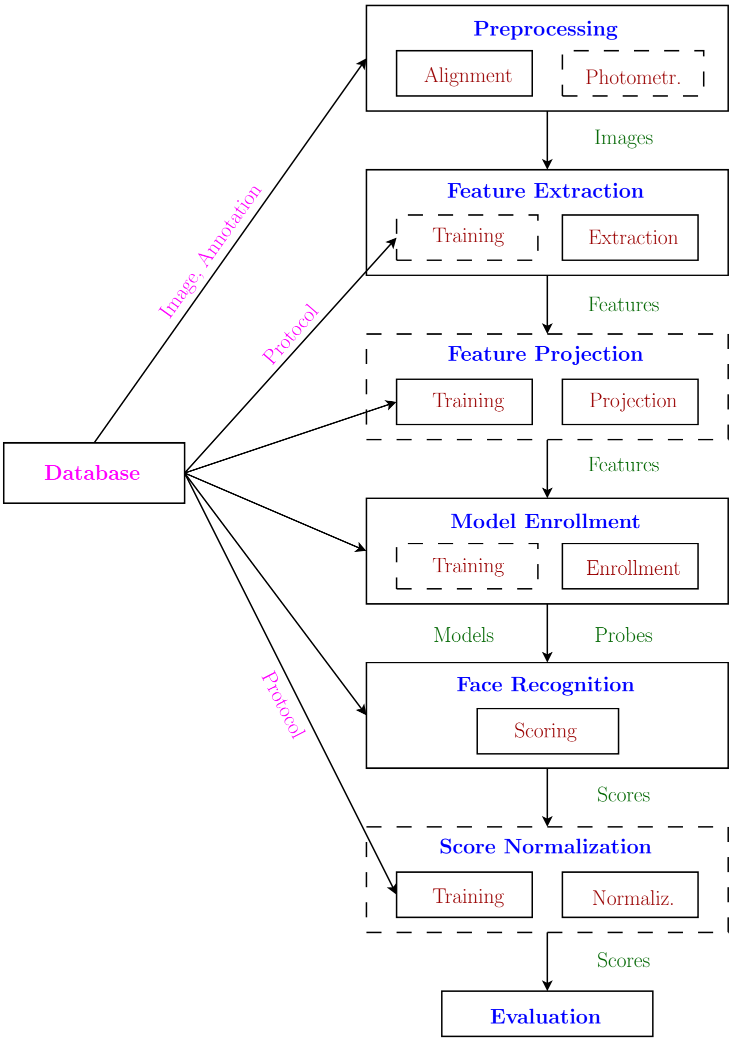
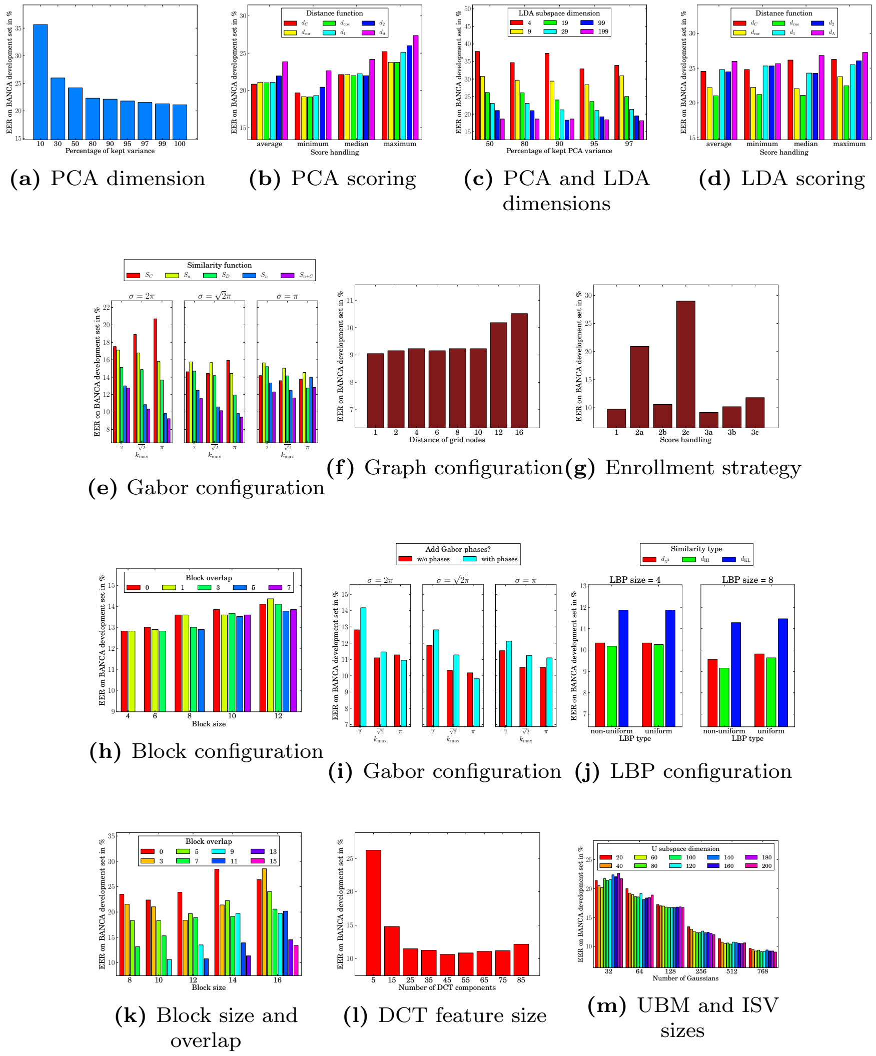


Figure 15: Face recognition. *This figure shows the generic execution order of face recognition algorithms. Boxes with dashed lines indicate optional steps, while solid lines show certain steps.*

A.2 Configuration optimization



(a) PCA dimension

(b) PCA scoring

(c) PCA and LDA dimensions

(d) LDA scoring

(e) Gabor configuration

(f) Graph configuration

(g) Enrollment strategy

(h) Block configuration

(i) Gabor configuration

(j) LBP configuration

(k) Block size and overlap

(l) DCT feature size

(m) UBM and ISV sizes

Figure 16: CONFIGURATION OPTIMIZATION. *This figure displays configuration optimization results of PCA ((a) – (b)), LDA ((c) – (d)), Graphs ((e) – (g)), LGBPHS ((h) – (j)) and ISV ((k) – (m)).*

B Image Resolutions used in Literature

image size; area (database)	**algorithm**	**references**
90 pixel between eyes	ISO standard (Page 20)	[43]
64×64; inner face (CAS-PEAL)	PCA, PCA+LDA, GPCA+LDA, LGBPHS	[24]
92×112; whole face (AT & T)	Precropped images in the AT & T database	[81]
250×250; whole face (LFW)	Precropped images in the LFW database	[41]
256×384; whole face (FERET)	Original uncropped images of FERET	[73]
21×12 (FERET)	PCA, ICA, BIC; LDA, Kernel-PCA	[60, 45]
50×60 (FERET)	PCA, PCA+LDA, PCA+ICA	[19, 10]
128×128; face + hair	PCA	[87]
128×128 (GBU)	Local Region PCA	[69]
25×30 (AT & T)	LDA	[22]
57×61 (XM2VTS)	LDA	[48]
64×49	LDA	[80]
42×48, 84×96 (FERET)	PCA+LDA	[111]
65×75 (GBU)	CohortLDA	[53]
128×128; inner face (FERET, FRGC)	Kernel Fisher Analysis	[49]
130×150 (FERET)	Bayesian Intrapersonal Extrapersonal Classifier	[91]
19×19 (MoBo)	Local Binary Patterns	[36]
68×84 (XM2VTS, BANCA)	Adapted LBP Histograms	[78]
80×88 (FERET)	LGBPHS	[108]
120×120; inner face (FRGC1, Yale, CMU-PIE)	Local Ternary Patterns	[90]
130×150, 128×128 (FERET)	Local Binary Patterns	[2, 3]
55×51, 150×115, **220×200** (BANCA)	Gabor jets	[46]
12×15 — 600×800; face + hair (CAS-PEAL)	Gabor graphs	[37]
64×64, 88×88, **128×128** (FERET, CAS-PEAL)	Histogram of Gabor Phase Patters	[106]
64×80, 32×40 (XM2VTS)	DCT-GMM and DCT-MLP	[15]
128×128 (FERET, BANCA)	Gabor wavelets, General Discriminant Analysis	[85]
128×128; face + hair (FERET)	EBGM	[99]
168×224 + **300×400**; face + hair (CAS-PEAL, FRGC)	Gabor graphs, Gabor phases	[35, 33]
68×68 (BANCA)	Local Frequency Bands	[57]
64×56; eyes+nose (VidTIMIT, Weizman)	DCT Features	[82]
91×114 (BANCA)	DCT-GMM	[52]
56×46 (AT & T, AR face, FERET)	Energy based models	[16]
92×112 (Yale, FERET, AT & T)	Markov Random Fields	[42]
92×112 (AT & T); 243×320 (Yale)	SIFT	[6]
50×57 (AT & T); 60×85 (AR face)	SIFT	[27]

Table 2: Image Resolutions. *This table lists an assortment of image resolutions and facial areas as used for face recognition by other researchers.*

C Exact numbers

	Graphs	**ISV**	**LDA**	**LDA-IR**	**LGBPHS**	**LRPCA**	**PCA**
-90°	49.21 %	34.81 %	39.84 %	n/a	44.14 %	n/a	48.13 %
-75°	49.93 %	24.22 %	36.33 %	n/a	46.20 %	n/a	42.55 %
-60°	44.44 %	12.53 %	36.72 %	n/a	48.41 %	n/a	44.14 %
-45°	41.02 %	20.70 %	41.41 %	12.90 %	43.75 %	40.18 %	54.30 %
-30°	24.61 %	3.91 %	29.33 %	6.66 %	22.66 %	24.69 %	35.55 %
-15°	8.59 %	0.39 %	17.58 %	3.87 %	8.63 %	10.94 %	16.41 %
+0°	1.95 %	0.04 %	10.16 %	2.31 %	1.97 %	5.86 %	7.81 %
+15°	7.81 %	0.39 %	18.66 %	3.57 %	7.42 %	9.38 %	14.45 %
+30°	22.21 %	2.40 %	25.00 %	6.25 %	21.88 %	23.78 %	34.73 %
+45°	41.02 %	17.97 %	37.89 %	12.08 %	37.89 %	38.67 %	51.52 %
+60°	43.75 %	16.80 %	37.12 %	n/a	47.70 %	n/a	42.67 %
+75°	46.46 %	26.57 %	36.33 %	n/a	47.66 %	n/a	43.70 %
+90°	47.25 %	38.68 %	35.86 %	n/a	45.32 %	n/a	45.31 %

(a) EER on all poses

	Graphs	**ISV**	**LDA**	**LDA-IR**	**LGBPHS**	**LRPCA**	**PCA**
-90°	47.89 %	40.38 %	36.98 %	n/a	50.69 %	n/a	44.98 %
-75°	44.10 %	24.02 %	39.35 %	n/a	48.53 %	n/a	43.17 %
-60°	44.42 %	16.36 %	38.00 %	n/a	50.29 %	n/a	41.56 %
-45°	44.80 %	20.93 %	39.50 %	15.72 %	42.47 %	42.42 %	52.59 %
-30°	27.28 %	4.28 %	29.43 %	7.96 %	26.52 %	25.52 %	36.41 %
-15°	9.91 %	1.77 %	22.18 %	4.89 %	9.48 %	11.45 %	19.18 %
+0°	2.75 %	1.01 %	12.37 %	3.97 %	3.16 %	7.85 %	10.52 %
+15°	9.10 %	1.40 %	19.87 %	5.52 %	9.08 %	10.89 %	18.77 %
+30°	24.97 %	3.34 %	31.46 %	9.22 %	22.56 %	24.43 %	36.62 %
+45°	43.70 %	18.88 %	37.76 %	15.21 %	40.08 %	39.50 %	52.54 %
+60°	46.93 %	21.41 %	37.48 %	n/a	47.37 %	n/a	44.10 %
+75°	46.32 %	30.55 %	37.99 %	n/a	45.89 %	n/a	44.67 %
+90°	47.49 %	35.44 %	41.17 %	n/a	48.52 %	n/a	48.28 %

(b) HTER on all poses

Table 3: Poses I. *This table reports the exact numbers for the plot in fig. 10(b), for development and evaluation set.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
-90°	49.21 %	32.81 %	37.50 %	n/a	44.14 %	n/a	47.27 %
-75°	49.93 %	18.75 %	35.49 %	n/a	46.20 %	n/a	43.85 %
-60°	44.44 %	9.38 %	32.08 %	n/a	48.41 %	n/a	45.67 %
-45°	41.02 %	11.33 %	33.18 %	10.16 %	43.75 %	35.58 %	54.71 %
-30°	24.61 %	1.95 %	20.31 %	5.48 %	22.66 %	22.27 %	35.86 %
-15°	8.59 %	0.39 %	9.38 %	1.88 %	8.63 %	7.81 %	14.48 %
+0°	1.95 %	0.02 %	n/a	3.12 %	1.97 %	4.33 %	8.18 %
+15°	7.81 %	0.39 %	7.42 %	3.21 %	7.42 %	5.49 %	16.80 %
+30°	22.21 %	2.39 %	19.92 %	3.48 %	21.88 %	21.48 %	37.54 %
+45°	41.02 %	12.11 %	30.47 %	8.62 %	37.89 %	35.12 %	53.91 %
+60°	43.75 %	13.27 %	31.57 %	n/a	47.70 %	n/a	43.00 %
+75°	46.46 %	22.71 %	32.42 %	n/a	47.66 %	n/a	44.90 %
+90°	47.25 %	33.20 %	37.11 %	n/a	45.32 %	n/a	45.31 %

(a) EER on tested poses

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
-90°	47.89 %	32.86 %	37.40 %	n/a	50.69 %	n/a	44.46 %
-75°	44.10 %	21.04 %	35.54 %	n/a	48.53 %	n/a	43.99 %
-60°	44.42 %	10.84 %	30.28 %	n/a	50.29 %	n/a	43.77 %
-45°	44.80 %	14.87 %	35.16 %	12.69 %	42.47 %	38.21 %	53.27 %
-30°	27.28 %	2.97 %	22.54 %	7.96 %	26.52 %	21.52 %	38.82 %
-15°	9.91 %	0.73 %	9.28 %	4.33 %	9.48 %	9.45 %	16.84 %
+0°	2.75 %	0.80 %	n/a	3.73 %	3.16 %	6.45 %	10.77 %
+15°	9.10 %	0.44 %	9.16 %	4.61 %	9.08 %	8.81 %	17.62 %
+30°	24.97 %	2.95 %	19.70 %	8.69 %	22.56 %	20.66 %	38.41 %
+45°	43.70 %	12.15 %	30.92 %	10.66 %	40.08 %	36.60 %	52.73 %
+60°	46.93 %	15.91 %	36.42 %	n/a	47.37 %	n/a	44.48 %
+75°	46.32 %	22.89 %	34.92 %	n/a	45.89 %	n/a	43.73 %
+90°	47.49 %	30.06 %	37.32 %	n/a	48.52 %	n/a	45.81 %

(b) HTER on tested poses

Table 4: Poses II. *This table reports exact numbers for an experiment that trains the algorithms only on the tested pose.*

	Graphs	**ISV**	**LDA**	**LDA-IR**	**LGBPHS**	**LRPCA**	**PCA**
all	15.94 %	3.49 %	16.84 %	4.20 %	14.76 %	17.19 %	19.62 %
neutral	2.34 %	0.03 %	5.86 %	1.99 %	2.28 %	6.25 %	7.41 %
smile	6.25 %	1.56 %	15.62 %	4.69 %	7.81 %	10.94 %	15.62 %
surprise	6.25 %	1.56 %	18.45 %	3.12 %	3.48 %	15.49 %	20.31 %
squint	9.38 %	0.00 %	17.55 %	3.40 %	9.45 %	18.43 %	21.88 %
disgust	20.31 %	4.69 %	27.74 %	4.69 %	21.89 %	26.46 %	32.81 %
scream	28.12 %	9.38 %	31.25 %	9.38 %	20.31 %	31.25 %	34.56 %

(a) EER

	Graphs	**ISV**	**LDA**	**LDA-IR**	**LGBPHS**	**LRPCA**	**PCA**
all	16.54 %	3.99 %	16.37 %	5.52 %	15.43 %	16.29 %	20.66 %
neutral	2.96 %	0.81 %	7.85 %	4.94 %	3.30 %	7.36 %	9.56 %
smile	7.45 %	1.01 %	16.54 %	3.14 %	6.56 %	13.19 %	18.08 %
surprise	11.68 %	1.80 %	20.83 %	7.84 %	9.34 %	16.17 %	24.17 %
squint	9.62 %	1.60 %	14.57 %	5.54 %	9.07 %	13.70 %	19.92 %
disgust	16.91 %	3.92 %	21.73 %	5.37 %	16.80 %	21.21 %	26.31 %
scream	24.89 %	8.21 %	27.86 %	8.05 %	16.25 %	27.50 %	34.00 %

(b) HTER

Table 5: EXPRESSIONS. *This table reports the exact numbers for the plot in fig. 10a, for development and evaluation set.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
illumination	0.38 %	0.02 %	4.65 %	6.59 %	0.10 %	1.16 %	5.03 %
occlusion	22.09 %	0.46 %	21.44 %	8.67 %	24.27 %	21.51 %	33.72 %
both	23.55 %	1.16 %	21.80 %	14.24 %	26.45 %	21.51 %	34.01 %
scarf	4.65 %	1.25 %	14.55 %	15.70 %	5.23 %	7.05 %	18.02 %
sunglasses	26.28 %	1.16 %	26.12 %	12.21 %	27.96 %	27.88 %	45.27 %

(a) EER

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
illumination	1.22 %	0.98 %	4.69 %	6.76 %	1.63 %	2.72 %	5.82 %
occlusion	16.63 %	0.55 %	18.58 %	6.70 %	20.50 %	21.30 %	31.06 %
both	18.87 %	1.92 %	20.68 %	10.75 %	23.04 %	20.50 %	32.68 %
scarf	4.76 %	1.46 %	15.66 %	11.14 %	4.43 %	4.88 %	17.10 %
sunglasses	26.36 %	0.88 %	24.70 %	11.33 %	29.47 %	27.33 %	42.61 %

(b) HTER

Table 6: Occlusions. *This table reports the exact numbers for both plots in fig. 9, for development and evaluation set.*

	Graphs	ISV	LDA	LGBPHS	LRPCA	PCA	GPCA+LDA
background	99.10 %	100.00 %	96.02 %	98.92 %	98.37 %	90.60 %	98.00 %
distance	99.27 %	100.00 %	96.00 %	98.55 %	97.09 %	84.00 %	100.00 %
aging	95.45 %	100.00 %	71.21 %	89.39 %	86.36 %	45.45 %	98.50 %
expression	85.10 %	99.75 %	71.40 %	84.97 %	77.96 %	51.97 %	90.60 %
accessory	85.03 %	99.43 %	64.38 %	82.76 %	66.96 %	46.30 %	82.80 %
lighting	58.67 %	55.15 %	26.66 %	62.37 %	40.48 %	22.78 %	44.80 %

Table 7: CAS-PEAL. *This table shows the recognition rates of the different algorithms on all of the CAS-PEAL protocols.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA	CohortLDA
Good	77.46 %	83.74 %	68.88 %	79.19 %	77.22 %	73.28 %	57.96 %	84.59 %
Bad	25.08 %	29.03 %	23.54 %	41.80 %	27.60 %	25.63 %	14.62 %	49.53 %
Ugly	5.70 %	5.55 %	4.97 %	12.25 %	7.70 %	5.52 %	2.79 %	11.68 %

(a) GBU

Table 8: GBU. *This table reports the CAR at* $FAR = 0.1\%$ *for fig. 13.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA	baseline
2.0.1	67.77 %	95.11 %	72.96 %	67.67 %	69.14 %	73.44 %	43.42 %	66.00 %
2.0.2	96.70 %	99.82 %	86.32 %	75.12 %	93.14 %	92.21 %	81.67 %	88.00 %
2.0.4	19.11 %	42.05 %	42.05 %	50.92 %	23.10 %	19.45 %	10.65 %	12.00 %

(a) FRGC

Table 9: FRGC. *This table reports the CAR at* $FAR = 0.1\%$ *for fig. 12.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
female	10.00 %	4.50 %	17.19 %	12.55 %	10.90 %	12.12 %	17.79 %
male	10.00 %	3.37 %	15.12 %	8.98 %	10.12 %	16.35 %	14.44 %

(a) EER

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
female	16.37 %	10.71 %	20.77 %	18.69 %	17.43 %	19.40 %	19.33 %
male	12.83 %	6.34 %	19.04 %	11.01 %	14.33 %	16.90 %	16.66 %

(b) HTER

Table 10: MOBIO. *This table reports the exact numbers for both plots in fig. 14a.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
combined	44.70 %	32.42 %	30.75 %	19.68 %	44.09 %	31.97 %	39.85 %
close	31.34 %	20.91 %	29.55 %	20.45 %	27.72 %	29.44 %	38.18 %
medium	40.45 %	28.18 %	28.62 %	17.72 %	36.82 %	28.73 %	35.45 %
far	47.36 %	40.80 %	36.36 %	20.91 %	45.09 %	36.82 %	45.94 %

(a) EER

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
combined	44.45 %	34.34 %	37.06 %	23.14 %	44.20 %	34.67 %	41.75 %
close	30.79 %	22.02 %	37.66 %	23.38 %	26.41 %	30.45 %	41.00 %
medium	37.02 %	30.46 %	35.39 %	20.76 %	34.57 %	33.98 %	39.92 %
far	48.11 %	46.18 %	39.46 %	24.43 %	44.39 %	40.22 %	44.06 %

(b) HTER

Table 11: SC FACE. *This table reports the exact numbers for both plots in fig. 14b.*

	Graphs	ISV	LDA	LDA-IR	LGBPHS	LRPCA	PCA
fold1	67.00 %	75.20 %	62.20 %	71.80 %	67.70 %	64.70 %	62.80 %
fold2	71.20 %	77.50 %	59.00 %	71.20 %	72.00 %	61.00 %	63.30 %
fold3	68.20 %	72.50 %	63.00 %	74.00 %	68.70 %	65.70 %	65.20 %
fold4	62.70 %	73.50 %	59.00 %	71.20 %	63.70 %	65.20 %	67.00 %
fold5	70.00 %	73.80 %	58.00 %	70.30 %	68.20 %	60.20 %	61.20 %
fold6	70.20 %	73.20 %	60.00 %	70.00 %	68.00 %	64.00 %	68.80 %
fold7	66.20 %	73.70 %	60.70 %	73.30 %	68.30 %	63.00 %	65.30 %
fold8	65.70 %	74.80 %	58.50 %	71.50 %	67.50 %	59.20 %	62.30 %
fold9	67.30 %	75.70 %	59.80 %	74.20 %	67.20 %	63.30 %	65.00 %
fold10	68.80 %	77.50 %	62.70 %	72.50 %	70.50 %	68.70 %	67.70 %
mean	67.72 %	74.73 %	60.28 %	72.00 %	68.17 %	63.48 %	64.87 %
std	2.39	1.65	1.69	1.38	2.06	2.69	2.35

Table 12: LFW. *This table reports the classification success rates for all folds of the LFW database, see fig. 14c for the summary plot.*