The CSU Face Identification Evaluation System: Its Purpose, Features and Structure

David S. Bolme, J. Ross Beveridge, Marcio Teixeira and Bruce A. Draper
Computer Science Department Colorado State University

Colorado State University

Abstract. The CSU Face Identification Evaluation System provides standard face recognition algorithms and standard statistical methods for comparing face recognition algorithms. The system includes standardized image pre-processing software, three distinct face recognition algorithms, analysis software to study algorithm performance, and Unix shell scripts to run standard experiments. All code is written in ANSI C. The preprocessing code replicates feature of pre-processing used in the FERET evaluations. The three algorithms provided are Principle Components Analysis (PCA), a.k.a Eigenfaces, a combined Principle Components Analysis and Linear Discriminant Analysis algorithm (PCA+LDA), and a Bayesian Intrapersonal/Extrapersonal Classifier (BIC). The PCA+LDA and BIC algorithms are based upon algorithms used in the FERET study contributed by the University of Maryland and MIT respectively. There are two analysis. The first takes as input a set of probe images, a set of gallery images, and similarity matrix produced by one of the three algorithms. It generates a Cumulative Match Curve of recognition rate versus recognition rank. The second analysis tool generates a sample probability distribution for recognition rate at recognition rank 1, 2, etc. It takes as input multiple images per subject, and uses Monte Carlo sampling in the space of possible probe and gallery choices. This procedure will, among other things, add standard error bars to a Cumulative Match Curve. The System is available through our website and we hope it will be used by others to rigorously compare novel face identification algorithms to standard algorithms using a common implementation and known comparison techniques.

1 Introduction

The The System was created to evaluate how well face identification systems perform. In addition to algorithms for face identification, the system includes software to support statistical analysis techniques that aid in evaluating the performance of face identification systems. The current system is designed with identification rather than verification in mind. The identification problem is: given a novel face, find the most similar images in a gallery of known people/images. The related verification problem is: given a novel image of specific person, confirm whether the person is or is not who they claim to be.

For simplicity sake, the CSU Face Identification and Evaluation System will henceforth be called the System. The System assumes, as did the earlier FERET evaluation[7], that a face recognition algorithm will first compute a similarity measure between images, and second perform a nearest neighbor match between novel and stored images. When this is true, the complete behavior of a face identification system can be captured in terms of a similarity matrix. The System will create these similarity matrices and provides analysis tools that utilize them generate cumulative match curves and recognition rate sample probability distributions. This document describes version 4.0 of the System that is available through our website[2].

2 System Overview

The System functionality can be split into four basic phases: image data preprocessing, algorithm training, algorithm testing and analysis of results: see Figure1. Preprocessing reduces unwanted image variation by aligning the face imagery, equalizing the pixel values, and normalizing the contrast and brightness. The three algorithms in this distribution have a training phase and a testing phase. The training phase reads training data and creates a subspace into which test images will be projected and matched. The testing phase reads the subspace information, projects images into this subspace, and generates a distance matrix. Typically the testing phase creates a distance matrix for the union of all images to be used either as probe images or gallery images in the analysis phase. The fourth phase performs analyzes on the distance matrices. This include computing recognition rates (csuAnalyzeRankCurve), conducting virtual experiments (csuAnalizePermute), or performing other statistical analysis on the data.

Sections 3 and 4 describe this functionality in greater detail. Before proceeding to discuss functionality further, there are four data structures commonly used to pass information between components of the System. These are imagery, image sets, algorithm training configurations, and distances matrices.

2.1 Imagery

The System was developed using frontal facial images from the FERET data set. Images are stored in an image file that contains pixel values in a binary floating point format (Big Endian / Sun byte order). The current system generates ".sfi" files. SFI stands for Single Float Image. Each image file contains a single line (record) ASCII header that contains the format specifier, "CSU_RASTER", followed by the column dimension, followed by the row dimension, followed by the number channels of data. The remainder of the file contains raw pixel values in row major order. Most images are single channel, but for multi-channel images the pixel value for channel two follows directly the pixel value for one, etc. The floating point portion of a single channel SFI file is identical to the NIST FERET image format. The only difference is our addition of a header. The The System also supports this NIST format and identifies such images with a ".nrm" suffix.

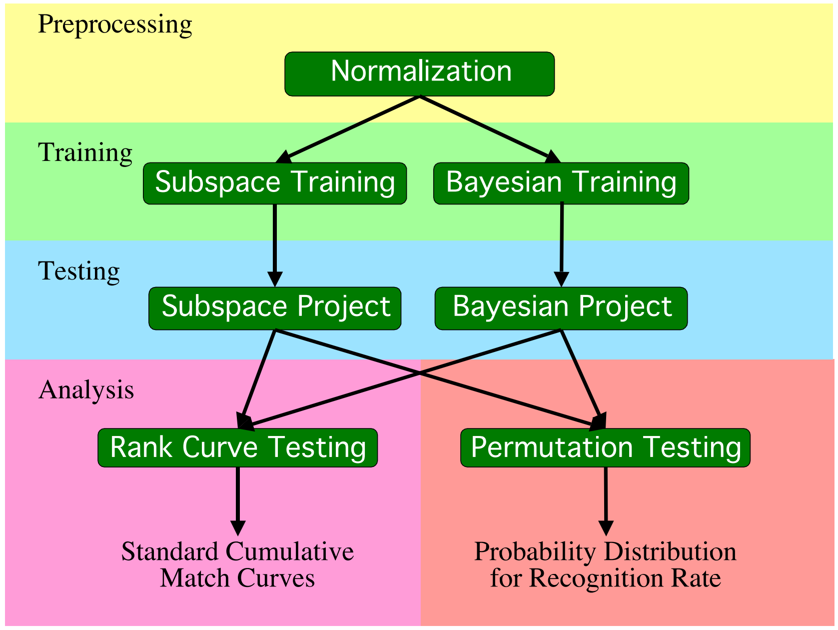

Fig. 1. Overview of execution flow for the csuSubspace system, which includes a standard PCA identification algorithm and also a PCA+LDA identification algorithm.

2.2 Image Sets

It is impossible to run experiments without first identifying sets of images to be used to train and test algorithms. This distribution includes many common image lists, including the training images, gallery images, and four standard probe sets used in the FERET evaluations. While image lists are always ASCII files enumerating filenames of image files, they are used to represent training image sets, test image sets, probe image sets and gallery image sets. When running experiments, it is important to keep track of how distinct lists are being used. The actual FERET training, gallery and probe set lists are available at: http://www.cs.colostate.edu/evalfacerec/data.html

2.3 Training Configuration Files

These files contain subspace basis vectors, associated eigenvalues, along with algorithms specific meta-data such as an ASCII copy of the command line used to generate the training data. The training files are a combination of binary and ASCII data: an ASCII header followed by binary data. Specifically, the basis vectors are stored as 64 bit floating point values. These files are inputs to the testing algorithms and carry all the necessary information generated by the algorithm training phase.

2.4 Distance Matrices

Each algorithm produces a distance matrix for all of the images in the testing list. All algorithms assume that smaller distances are a closer match. In many cases the base metric will yield a similarity score, where higher scores indicate more similarity. When this is the case the similarity values are negated to produce a “distance like” metric. Some examples of this are the Correlation and MahAngle distance metrics in csuSubpace, and the Bayesian and Maximum Likelihood Metric in the csuBayesian code.

3 Preprocessing

Preprocessing is conducted using the executable csuPreprocessNormalize. The executable performs five steps in converting a PGM FERET image to a normalized image. The five steps are summarized in Figure2 and an sample normalized image is shown. The eye coordinates are required for geometric normalization. These are available for the FERET images from NIST and are included in the System.

Sample Normalized Image

1. Integer to float conversion - Converts 256 gray levels into floating point equivalents.
2. Geometric normalization -- Lines up human chosen eye coordinates.
3. Masking -- Crops the image using an elliptical mask and image borders such that only the face from forehead to chin and cheek to cheek is visible.
4. Histogram equalization -- Equalizes the histogram of the unmasked part of the image.
5. Pixel normalization -- scales the pixel values to have a mean of zero and a standard deviation of one.

Fig. 2. Image Normalization

Our csuPreprocessNormalize code accomplishes many of the same tasks performed by code originally written at NIST called “facetonorm”. However, it is not identical to the NIST version. For example, histogram equalization is done only within the unmasked portions of the face. Our code is more robust and we recommend using it in place of the NIST version.

4 Algorithms

Version 4.0 of the System comes with three face identification algorithms. These algorithms where chosen because they are well known and had high scores on the FERET

Phase 3 test. The algorithms are intended to perform as a test platform for evaluation techniques and to serve as a common baseline for algorithm comparisons.

4.1 Principle Components Analysis (csuSubspace/PCA)

The first algorithm released by CSU was based on Principle Components Analysis (PCA)[5]. This system is based on a linear transformation in feature space. Feature vectors for the PCA algorithm are formed by concatenating the pixel values from the images. These raw feature vectors are very large (~20,000 values) and are highly correlated. PCA rotates feature vectors from this large, highly correlated subspace to a small subspace which has no sample covariance between features.

PCA has two useful properties when used in face recognition. The first is that it can be used to reduce the dimensionality of the feature vectors. This dimensionality reduction can be performed in either a lossy or lossless manor. When applied in a lossy manor, basis vectors are truncated from the front or back of the transformation matrix. It is assumed that these vectors correspond to not useful information such as lighting variations (when dropped from the front) or noise (when dropped from the back). If none of the basis vectors are dropped it is called a lossless transformation and it should be possible to get perfect reconstruction for the training data based on the compressed feature vectors.

The second useful property is that PCA eliminates all of the statistical covariance in the transformed feature vectors. This means that the covariance matrix for the transformed (training) feature vectors will always be diagonal. This property is exploited for some distance measures such as L1, MahAngle, and Bayesian based classifiers.

Training PCA training is performed by the csuSubspaceTraining executable. The PCA is the default mode (it can also perform LDA training). The PCA basis is computed by the snapshot method using a Jacobi eigensolver from the Intel CV library. The basis vectors can be eliminated from the subspace using the cutOff and dropNVectors command line options. These methods are described in detail in[9]. The training program outputs a training file that contains a description of the training parameters, the mean of the training image, the eigenvalues or fisher values, and a basis for the subspace.

Distance Metrics The csuSubspaceProject code is used to generate distance files. It requires a list of images and a subspace training file. The code projects the feature vectors onto the basis. It then computes the distance between pairs of images in the list. The output is a set of distance files containing the distance from each image to all other images in the list. The distance metrics include city block (L1), Euclidean (L2), Correlation, Covariance, versus Angle (PCA only), and LDA Soft (LDA only). We have published a study comparing PCA to PCA+LDA using these different distance metrics[1]

4.2 Linear Discriminant Analysis (csuSubspace/PCA+LDA)

The second algorithm is Linear Discriminant Analysis (PCA+LDA) based upon that written by Zhao and Chellapa[10]. The algorithm is based on Fischer's Linear Discriminants. LDA training attempts to produce a linear transformations that emphasize differences between classes while reducing differences within classes. The goal is to form a subspace that is linearly separable between classes.

When used in the Face Identification and Evaluation System each human subject forms a class. LDA training requires training data that has multiple images per subject. LDA training is performed by first using PCA to reduces the dimensionality of the feature vectors. After this LDA is performed on the training data to further reduces the dimensionality in such a way that class distinguishing features are preserved. A final transformation matrix is produced by multiplying the PCA and LDA basis vectors to produce a full raw to LDA space transformation matrix.

The final output of the LDA training is the same as PCA. The algorithm produces a set of LDA basis vectors. These basis vectors produce a transformation of the feature vectors. Like the PCA algorithm, distance metrics can be used on the LDA feature vectors.

Training Like PCA, LDA training is performed by the csuSubspaceTraining executable. This algorithm is enabled using the -lda option. PCA is first performed on the training data to determine an optimal basis for the image space. The training images are projected onto the PCA subspace to reduce their dimensionality before LDA is performed. Computationally LDA follows the method outlined by [3]. A detailed description of the implementation can be found in[4]. The subspace generated using the -lda option is the composition of the PCA followed by the LDA projection matrices. LDA generates one fewer basis vectors than there are training classes, i.e. training subjects.

Distance Metrics The csuSubspaceProject generate distance files for PCA+LDA. Please see the PCA distance metrics section for more information.

4.3 Bayesian Intrapersonal Classifier (csuBayesian/BIC)

The third algorithm in the CSU distribution is based on an algorithm developed by Moghaddam and Pentland[6]. There are two variants of this algorithm, a *maximum a posteriori* (MAP) and *maximum likelihood* (ML) classifier. This algorithm is interesting for several reasons, including the fact that it examines the difference image between two photos as a basis for determining whether the two photos are of the same subject. Difference images which originate from two photos of different subjects are said to be *extrapersonal* whereas images which originate from two photos of the same subject are said to be *Intrapersonal*.

The key assumption in Moghaddam and Pentland's work is that the particular difference images belonging to the Intrapersonal and extrapersonal difference images originate from distinct and localized Gaussian distributions within the space of all possible difference images.

The actual parameters for these distributions are not known, so the algorithm begins by extracting from the training data, using statistical methods, the parameters that define the Gaussian distributions corresponding to the Intrapersonal and extrapersonal difference images. This training stage, called density estimation, is performed through Principle Components Analysis (PCA). This stage estimates the statistical properties of two subspaces: one for difference images that belong to the Intrapersonal class and another for difference images that belong to the extrapersonal class. During the testing phase, the classifier takes each image of unknown class membership and uses the estimates of the the probability distributions as a means of identification.

Training In the current CSU implementation, the extrapersonal and intrapersonal difference images for training are generated using the "csuMakeDiffs" program and subsequently the parameters of the two subspaces are estimated by running PCA ("csuSubspaceTrain"). This is done independently for the intrapersonal and extrapersonal difference images. Unlike in our earlier distribution, the current distribution does *not* include a separate and independent program for training the Bayesian classifier.

Distance Metrics The "csuBayesianProject" code is used to generate distance files. It requires a list of images and two subspace training files (one for the extrapersonal difference images and another for the intrapersonal difference images). The code projects the feature vectors onto each of the two sets of basis vectors and then computes the probability that each feature vector came from one or the other subspace. The output is a set of distance files containing the similarity from each image to all other images. The similarities may be computed using the maximum a posteriori (MAP) or the maximum likelihood (ML) methods. From a practical standpoint, the ML method uses information derived only from the intrapersonal images, while the MAP method uses information derived from both distributions.

5 Standardizing Algorithm Analysis

The primary motivation in developing the System is to analyze the performance of different algorithms. Many publications in the face identification domain compare recognition rates between different algorithms. However, different implementations of even such apparently simple algorithms as a PCA face identification algorithm can produce different outcomes, particularly if image preprocessing, training and choice of distance metric is not controlled. For example, in[1], the comparison of different PCA distance metrics showed that using PCA with L2 distance can create a false impression that an alternative algorithm is doing well: our studies as well as FERET, show PCA should be used with Mahalanobis Angle as the preferred distance metric.

It is therefore important that Our System provides not only standardized algorithms, but also standardized preprocessing, standardized scripts that include details such as choice of distance metric, and finally standardized ways of analyzing results. In this section we will explain in more detail the two analysis tools included in the System.

5.1 Rank Curve Generation

Rank curve analysis was used in the FERET evaluations as one basis for algorithm comparison. It provides a method of analyzing recognition rates of an algorithm f recognition rank. Although this analysis is simple it can provide interesting information not apparent in a rank one recognition rate. Figures 3 and 4 show the rank curves generated for the standard FA and FC FERET probe sets. The System comes with a set of Unix Scripts that run a portion of the original FERET evaluation for the three algorithms including all four of standard probe sets. This experiment is not identical to that done in FERET. Differences include new algorithm implementations, new image preprocessing code and perhaps most importantly different training image sets. Keeping those caveats in mind, the script will preprocess the FERET imagery, train the algorithms, run the algorithms to generate distance matrices, and finally build cumulative match curves for the standard set of FERET gallery images and each of the four standard FERET probe image sets. This script is intended as a baseline, or point of departure, for people wanted understand what was done in the FERET evaluation and wanting to adapt it to their own purposes.

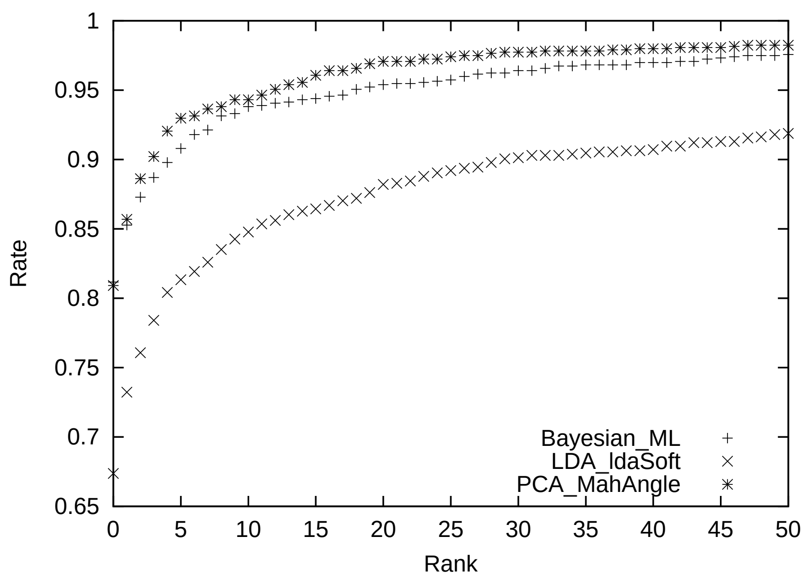

Fig. 3. Comparison of the three algorithms on the FB probe set.

5.2 Permuting Probe and Gallery Image Choices

A weakness of comparing recognition rates in cumulative match curves is they lack standard error bars. The question of what one really wants standard error bars to represent, and thus how to compute them, can become more involved that at first it might

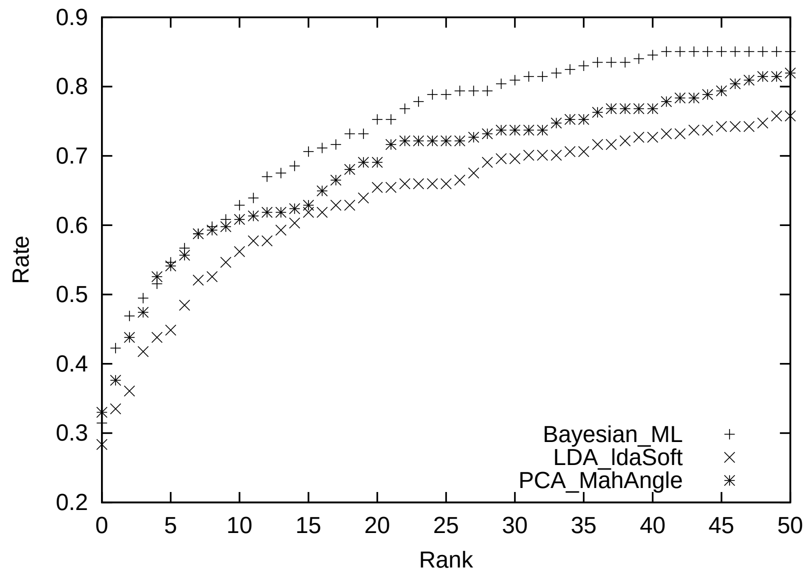

Fig. 4. Comparison of the three algorithms on the FC probe set.

appear. We have developed a Monte Carlo based method that is described fully in [1]. An alternative means of computing error bars has been developed by Ross Micheals and Terry Boult[8]. Our csuPermute code performs virtual experiments using the distance files. It does this by taking random permutations of the probe and gallery sets and then performs nearest neighbor classification. It then generates a sample probability distribution for recognition rate under the assumption that probe and gallery images are interchangeable for subjects. Figure 5 shows an example comparing the PCA and BIC algorithms on a set of 640 FERET images of 120 subjects. Observe that average performance of the PCA algorithm is higher than BIC, but that relative to standard error bars derived from the sample distributions, the difference does not appear significant relative to changes in the choice of probe and gallery images.

References

1. J. Ross Beveridge, Kai She, Bruce Draper, and Geof H. Givens. A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 535 – 542, December 2001.
2. Ross Beveridge. Evaluation of face recognition algorithms web site. `http:\\cs.colostate.edu\evalfacerec`.
3. Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. John Wiley & Sons, second edition edition, 2001.
4. J. Ross Beveridge. The Geometry of LDA and PCA Classifiers Illustrated with 3D Examples. Technical Report CS-01-101, Computer Science, Colorado State University, 2001.

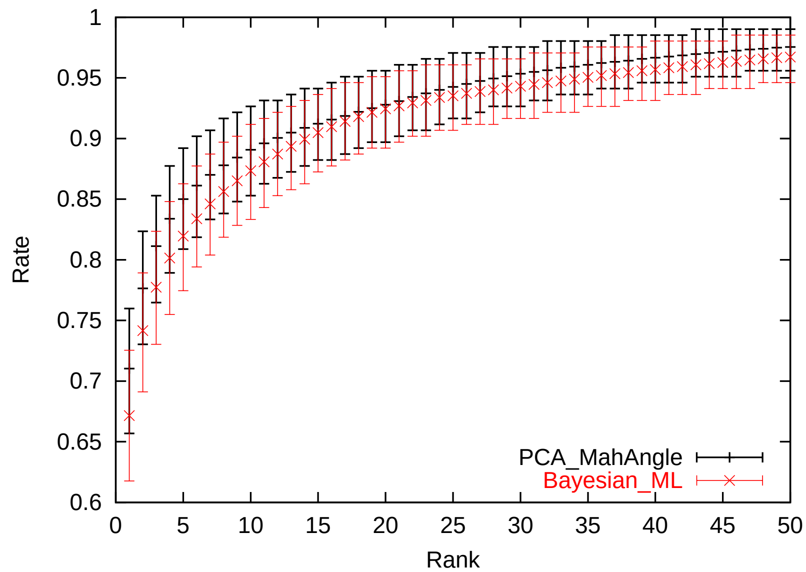

Fig. 5. This figure compares our PCA and Bayesian algorithms. The rank curves include 95% error bars that were estimated by csuPermute. The error bars show that there is no significant difference between PCA and Maximum Likelihood when trained on the FERET training set.

5. M. A. Turk and A. P. Pentland. Face Recognition Using Eigenfaces. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages 586 – 591, June 1991.
6. B. Moghaddam, C. Nastar, and A. Pentland. A bayesian similarity measure for direct image matching. *ICPR*, B:350–358, 1996.
7. P.J. Phillips, H.J. Moon, S.A. Rizvi, and P.J. Rauss. The FERET Evaluation Methodology for Face-Recognition Algorithms. *T-PAMI*, 22(10):1090–1104, October 2000.
8. Ross J. Micheals and Terry Boult. Efficient evaluation of classification and recognition systems. In *IEEE Computer Vision and Pattern Recognition 2001*, page (to appear), December 2001.
9. Wendy S. Yambor. Analysis of pca-based and fisher discriminant-based image recognition algorithms. Master's thesis, Colorado State University, 2000.
10. W. Zhao, R. Chellappa, and A. Krishnaswamy. Discriminant analysis of principal components for face recognition. In *In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, Face Recognition: From Theory to Applications*, pages 73–85, 1998.