ACADEMIC PRESS

Available at www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Computer Vision and Image Understanding 91 (2003) 115–137

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

Recognizing faces with PCA and ICA

Bruce A. Draper,[a,*] Kyungim Baek,[b] Marian Stewart Bartlett,[c] and J. Ross Beveridge[a]

[a] *Department of Computer Science, Colorado State University, Ft. Collins, CO 80523, USA*
[b] *Department of Biomedical Engineering, Columbia University, New York, NY 10027, USA*
[c] *Institute for Neural Computation, University of California San Diego, La Jolla, CA 92037, USA*

Received 15 February 2002; accepted 11 February 2003

Abstract

This paper compares principal component analysis (PCA) and independent component analysis (ICA) in the context of a baseline face recognition system, a comparison motivated by contradictory claims in the literature. This paper shows how the relative performance of PCA and ICA depends on the task statement, the ICA architecture, the ICA algorithm, and (for PCA) the subspace distance metric. It then explores the space of PCA/ICA comparisons by systematically testing two ICA algorithms and two ICA architectures against PCA with four different distance measures on two tasks (facial identity and facial expression). In the process, this paper verifies the results of many of the previous comparisons in the literature, and relates them to each other and to this work. We are able to show that the FastICA algorithm configured according to ICA architecture II yields the highest performance for identifying faces, while the InfoMax algorithm configured according to ICA architecture II is better for recognizing facial actions. In both cases, PCA performs well but not as well as ICA.

1. Introduction

Over the last ten years, face recognition has become a specialized applications area within the larger field of computer vision. Sophisticated commercial systems

* Corresponding author. Fax: 1-970-491-2466.
E-mail addresses: draper@cs.colostate.edu (B.A. Draper), kb2107@columbia.edu (K. Baek), marni@salk.edu (M.S. Bartlett), ross@cs.colostate.edu (J.R. Beveridge).

1077-3142/03/$ - see front matter
doi:10.1016/S1077-3142(03)00077-8

perform face detection, image registration, and image matching, all in real time.[1] Although the details of most commercial systems are confidential, many of them perform image matching as a two-step process of subspace projection followed by classification in the space of compressed images. In a simple yet canonical scenario, face matching may be implemented as subspace projection followed by a nearest-neighbor classifier [34].

The sophistication of real-world commercial systems should not be underestimated. Many companies have developed innovative methods of face detection and registration. More importantly for this paper, they have enhanced their matching techniques, for example by pre-processing images, selecting and in some cases generating training data, generating spatially localized features, and optimizing classifiers for compressed subspaces. Sometimes the data being compressed are not face images at all, but differences of face images [33], Gabor jets [12], or other high-dimensional data computed from face images. Face recognition systems also employ a variety of techniques for selecting subspaces. As a result, it can be difficult to assign credit (or blame) to a particular component of a face recognition system, even when the details are not proprietary. The purpose of this paper is to compare the performance of two subspace projection techniques on face recognition tasks in the context of a simple baseline system. In particular, we compare principal component analysis (PCA) to independent component analysis (ICA), as implemented by the InfoMax [8] and FastICA [21] algorithms.

Why compare ICA to PCA? One reason is that the literature on the subject is contradictory. Bartlett, et al. [4,6], Liu and Wechsler [30], and Yuen and Lai [41] claim that ICA outperforms PCA for face recognition, while Baek et al. [1] claim that PCA outperforms ICA and Moghaddam [32] claims that there is no statistical difference in performance between the two. In a recent study with visible light and infra-red images, Socolinsky and Salinger report that ICA outperforms PCA on visible light images, but PCA outperforms ICA on LWIR images [37]. We also know of cases where researchers informally compared ICA to PCA while building a face recognition system, only to select PCA. The relative performance of the two techniques is therefore, an open question.

Part of the confusion stems from the number of factors that have to be controlled. The performance of PCA depends on the task statement, the subspace distance metric, and the number of subspace dimensions retained. The performance of ICA depends on the task, the algorithm used to approximate ICA, and the number of subspace dimensions retained. Even more confusingly, there are two very different applications of ICA to face recognition. ICA can be applied so as to treat images as random variables and pixels as observations, or to treat pixels as random variables and images as observations. In keeping with [2,6], we refer to these two alternatives as ICA architecture I and architecture II, respectively. There is therefore a space of possible PCA/ICA comparisons, depending on at least five factors. This

[1] For examples, see www.viisage.com, www.equinoxsensors.com, or www.identix.com/products/profaceit.html.

paper explores this space, in order to find the best technique for recognizing (1) subject identity and (2) facial actions in face images.

Another reason to explore the space of PCA/ICA comparisons is to provide data for the current debate over global versus local features. The basis vectors that define any subspace can be thought of as image features. Viewed this way, PCA and ICA architecture II produce global features, in the sense that every image feature is influenced by every pixel. (Equivalently, the basis vectors contain very few zeroes.) Depending on your preference, this makes them either susceptible to occlusions and local distortions, or sensitive to holistic properties. Alternatively, ICA architecture I produces spatially localized features that are only influenced by small parts of the image. It has been argued that this will produce better object recognition, since it implements recognition by parts [27]. If localized features are indeed superior, ICA architecture I should outperform PCA and ICA architecture II.

This paper will show empirically that the choice of subspace projection algorithm depends first and foremost on the nature of the task. Some tasks, such as facial identity recognition, are holistic and do best with global feature vectors. Other tasks, such as facial action recognition, are local and do better with localized feature vectors. For both types of tasks, ICA can outperform PCA, but only if the ICA architecture is selected with respect to the task type (ICA architecture I for localized tasks, ICA architecture II for holistic tasks). Furthermore, performance is optimized if the ICA algorithm is selected based on the architecture (InfoMax for architecture I, FastICA for architecture II). When PCA is used, the choice of subspace distance measure again depends on the task.

The rest of this paper is organized as follows. Section 2 provides the necessary background in terms of a brief introduction to face recognition (2.1), and the foundations of principal component analysis (2.2) and independent component analysis (2.3). Section 3 compares PCA with ICA architectures I and II on the task of facial identity recognition. Section 4 compares PCA and the ICA architectures on the task of recognizing facial expressions. Section 5 concludes with practical recommendations.

2. Background

2.1. Face recognition

Research in automatic face recognition dates back at least until the 1960s [13]. Most current face recognition techniques, however, date back only to the appearance-based recognition work of the late 1980s and 1990s. Kirby and Sirovich were among the first to apply principal component analysis (PCA) to face images, and showed that PCA is an optimal compression scheme that minimizes the mean squared error between the original images and their reconstructions for any given level of compression [26,36]. Turk and Pentland popularized the use of PCA for face recognition [40] (although see also [16]). They used PCA to compute a set of subspace basis vectors (which they called "eigenfaces") for a database of face images,

and projected the images in the database into the compressed subspace. New test images were then matched to images in the database by projecting them onto the basis vectors and finding the nearest compressed image in the subspace (eigenspace).

The initial success of eigenfaces popularized the idea of matching images in compressed subspaces. Researchers began to search for other subspaces that might improve performance. One alternative is Fisher's linear discriminant analysis (LDA, a.k.a. "fisherfaces") [38]. For any N-class classification problem, the goal of LDA is to find the N-1 basis vectors that maximize the interclass distances while minimizing the intraclass distances. At one level, PCA and LDA are very different: LDA is a supervised learning technique that relies on class labels, whereas PCA is an unsupervised technique. Nonetheless, in circumstances where class labels are available either technique can be used, and LDA has been compared to PCA in several studies [7,10,31,37].

One characteristic of both PCA and LDA is that they produce spatially global feature vectors. In other words, the basis vectors produced by PCA and LDA are non-zero for almost all dimensions, implying that a change to a single input pixel will alter every dimension of its subspace projection. There is also a lot of interest in techniques that create spatially localized feature vectors, in the hopes that they might be less susceptible to occlusion and would implement recognition by parts. The most common method for generating spatially localized features is to apply independent component analysis (ICA) to produce basis vectors that are statistically independent (not just linearly decorrelated, as with PCA) [2]. Non-negative matrix factorization (NMF) is another method for generating localized feature vectors [27].

ICA can also be used to create feature vectors that uniformly distribute data samples in subspace [4,5]. This conceptually very different use of ICA produces feature vectors that are not spatially localized. Instead, it produces feature vectors that draw fine distinctions between similar images in order to spread the samples in subspace. Keeping with the terminology introduced in [2,5], we refer to use of ICA to produce spatially independent basis vectors as architecture I, and the use of ICA to produce statistically independent compressed images as architecture II.

Gabor jets represent a substantially different class of subspace projection techniques. Unlike PCA, LDA, or ICA, the Gabor basis vectors are specified a-priori (although their spatial positions might be trained). Depending on the size and spacing of the jets, they also compress images far less than the other techniques; in fact, in extreme cases they may expand rather than contract the data. Nonetheless, there is evidence that subspaces defined by Gabor jets, although large, may be good for recognizing faces [12,23] and facial expressions [17].

Finally, there are techniques that apply mixtures of local linear subspaces. For example, Kambhatla and Leen mix local PCA subspaces to compress face data [24], and Frey et al. [19] apply a mixture of factor analyzers to recognize faces. Although they have not yet been applied to face recognition, Tipping and Bishop provide an EM algorithm for optimizing mixture models of PCA subspaces [39], and Lee et al. provide a similar algorithm for optimizing mixtures of ICA subspaces [28].

2.2. PCA

PCA is probably the most widely used subspace projection technique for face recognition. PCA basis vectors are computed from a set of training images **I**. As a first step, the average image in **I** is computed and subtracted from the training images, creating a set of data samples

$$i_1, i_2, \ldots, i_n \in \mathbf{I} - \bar{\mathbf{I}}.$$

These data samples are then arrayed in a matrix **X**, with one column per sample image

$$\mathbf{X} = \left[\begin{bmatrix} \vdots \\ i_1 \\ \vdots \end{bmatrix} \cdots \begin{bmatrix} \vdots \\ i_n \\ \vdots \end{bmatrix} \right]$$

$\mathbf{X}\mathbf{X}^{\mathbf{T}}$ is then the sample covariance matrix for the training images, and the principal components of the covariance matrix are computed by solving

$$\mathbf{R}^{\mathrm{T}}(\mathbf{X}\mathbf{X}^{\mathrm{T}})\mathbf{R} = \Lambda,$$

where Λ is the diagonal matrix of eigenvalues and **R** is the matrix of orthonormal eigenvectors. Geometrically, **R** is a rotation matrix[2] that rotates the original coordinate system onto the eigenvectors, where the eigenvector associated with the largest eigenvalue is the axis of maximum variance, the eigenvector associated with the second largest eigenvalue is the orthogonal axis with the second largest variance, etc. Typically, only the N eigenvectors associated with the largest eigenvalues are used to define the subspace, where N is the desired subspace dimensionality.

There are three related arguments for matching images in the subspace of N eigenvectors. The first is compression. It is computationally more efficient to compare images in subspaces with significantly reduced dimensions. For example, image vectors with 65,536 pixels (256×256) might be projected into a subspace with only 100–300 dimensions. The second argument assumes that the data samples are drawn from a normal distribution. In this case, axes of large variance probably correspond to signal, while axes of small variance are probably noise. Eliminating these axes therefore improves the accuracy of matching. The third argument depends on a common pre-processing step, in which the mean value is subtracted from every image and the images are scaled to form unit vectors. This projects the images into a subspace where Euclidean distance is inversely proportional to correlation between the source images. As a result, nearest neighbor matching in eigenspace becomes an efficient approximation to image correlation.

[2] This slightly liberal definition of rotation also includes reflection.

2.3. ICA

While PCA decorrelates the input data using second-order statistics and thereby generates compressed data with minimum mean-squared reprojection error, ICA minimizes both second-order and higher-order dependencies in the input. It is intimately related to the *blind source separation* (BSS) problem, where the goal is to decompose an observed signal into a linear combination of unknown independent signals. Let $\mathbf{s}$ be the vector of unknown source signals and $\mathbf{x}$ be the vector of observed mixtures. If $\mathbf{A}$ is the unknown mixing matrix, then the mixing model is written as

$$\mathbf{x} = \mathbf{A}\mathbf{s}.$$

It is assumed that the source signals are independent of each other and the mixing matrix $\mathbf{A}$ is invertible. Based on these assumptions and the observed mixtures, ICA algorithms try to find the mixing matrix $\mathbf{A}$ or the separating matrix $\mathbf{W}$ such that

$$\mathbf{u} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{A}\mathbf{s}$$

is an estimation of the independent source signals [14] (Fig. 1).

ICA can be viewed as a generalization of PCA. As previously discussed, PCA decorrelates the training data so that the sample covariance of the training data is zero. Whiteness is a stronger constraint that requires both decorrelation and unit variance. The whitening transform can be determined as $\mathbf{D}^{-1/2}\mathbf{R}^{\mathbf{T}}$, where $\mathbf{D}$ is the diagonal matrix of the eigenvalues and $\mathbf{R}$ is the matrix of orthogonal eigenvectors of the sample covariance matrix. Applying whitening to observed mixtures, however, results in the source signal only up to an orthogonal transformation. ICA goes one step further so that it transforms the whitened data into a set of statistically independent signals [22].

Signals are statistically independent when

$$\mathbf{f}_{\mathbf{u}}(\mathbf{u}) = \Pi_i \mathbf{f}_{\mathbf{u}_i}(\mathbf{u}_i),$$

where $\mathbf{f}_{\mathbf{u}}$ is the probability density function of $\mathbf{u}$. (It is equivalent to say that the vectors $\mathbf{u}$ are uniformly distributed.) Unfortunately, there may not be any matrix $\mathbf{W}$ that fully satisfies the independence condition, and there is no closed form expression

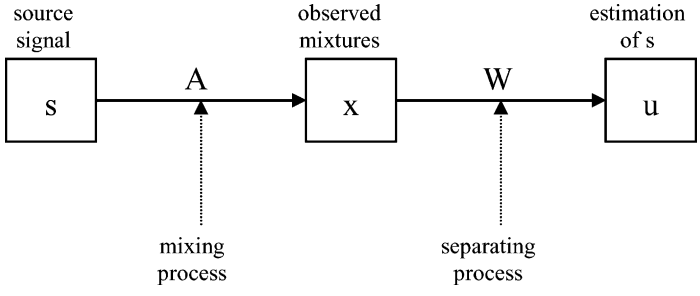


Fig. 1. Blind source separation model.

to find **W**. Instead, there are several algorithms that iteratively approximate **W** so as to indirectly maximize independence.

Since it is difficult to maximize the independence condition above directly, all common ICA algorithms recast the problem to iteratively optimize a smooth function whose global optima occurs when the output vectors **u** are independent. For example, InfoMax relies on the observation that independence is maximized when the entropy **H(u)** is maximized, where:

$$\mathbf{H}(\mathbf{u}) \equiv -\int \mathbf{f_u}(\mathbf{u}) \log \mathbf{f_u}(\mathbf{u})\, d\mathbf{u}.$$

InfoMax performs gradient ascent on the elements w_{ij} so as to maximize **H(u)** [8]. (It gets its name from the observation that maximizing **H(u)** also maximizes the mutual information **I(u,x)** between the input and output vectors.) The *JADE* algorithm minimizes the kurtosis of $\mathbf{f_u}(\mathbf{u})$ through a joint diagonalization of the fourth-order cumulants, since minimizing kurtosis will also maximize statistical independence. *FastICA* is arguably the most general, maximizing

$$J(y) \approx c[E\{G(y)\} - E\{G(v)\}]^2,$$

where G is a non-quadratic function, v is a gaussian random variable, and c is any positive constant, since it can be shown that maximizing any function of this form will also maximize independence [21].

InfoMax, JADE, and FastICA all maximize functions with the same global optima [14,21]. As a result, all three algorithms should converge to the same solution for any given data set. In practice, the different formulations of the independence constraint are designed to enable different approximation techniques, and the algorithms find different solutions because of differences among these techniques. Limited empirical studies suggest that the differences in performance between the algorithms are minor and depend on the data set. For example, Zibulevsky and Pearlmutter test all three algorithms on a simulated blind-source separation problem, and report only small differences in the relative error rate: 7.1% for InfoMax, 8.6% for FastICA, and 8.8% for JADE [42], On the other hand, Karvanen et al. [25] report on another simulated blind-source separation problem where JADE slightly outperforms FastICA, with InfoMax performing significantly worse. Ziehe et al. [43] report no significant difference between FastICA and JADE at separating noise from signal in MEG data. In studies using images, Moghaddam [32] and Lee et al. [28] report qualitatively similar results for JADE and FastICA, but do not publish numbers.

2.3.1. Architecture I: statistically independent basis images

Regardless of which algorithm is used to compute ICA, there are two fundamentally different ways to apply ICA to face recognition. In architecture I, the input face images in **X** are considered to be a linear mixture of statistically independent basis images **S** combined by an unknown mixing matrix **A**. The ICA algorithm learns the weight matrix **W**, which is used to recover a set of

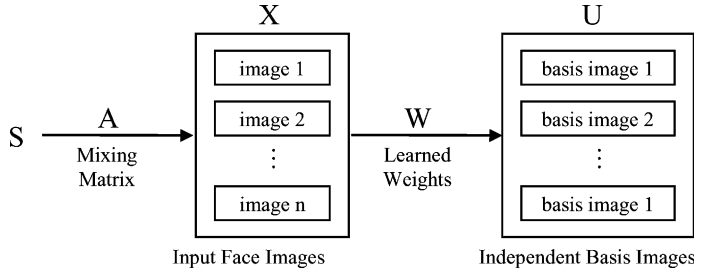


Fig. 2. Finding statistically independent basis images.

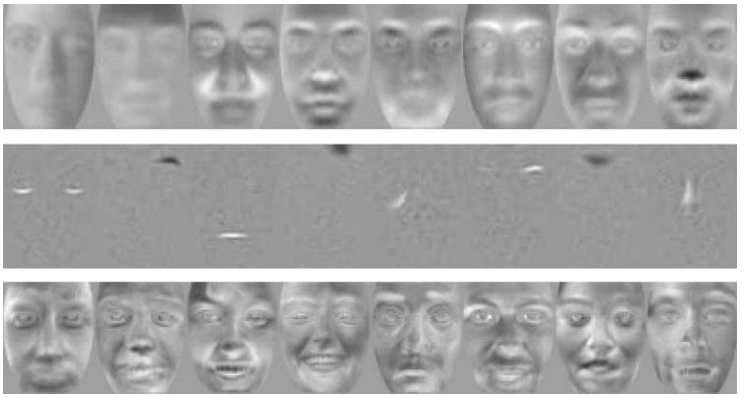


Fig. 3. Eight feature vectors for each technique. The top row contains the eight eigenvectors with highest eigenvalues for PCA. The second row shows eight localized feature vectors for ICA architecture I. The third row shows eight (non-localized) ICA feature vectors for ICA architecture II.

independent basis images in the rows of **U** (Fig. 2). In this architecture, the face images are variables and the pixel values provide observations for the variables. The source separation, therefore, is performed in face space. Projecting the input images onto the learned weight vectors produces the independent basis images. The compressed representation of a face image is a vector of coefficients used for linearly combining the independent basis images to generate the image. The middle row of Fig. 3 shows eight basis images produced in this architecture. They are spatially localized, unlike the PCA basis images (top row) and those produced by ICA architecture II (bottom row).

In [4,5], Bartlett and colleagues first apply PCA to project the data into a subspace of dimension m to control the number of independent components produced by ICA. The InfoMax algorithm is then applied to the eigenvectors to minimize the statistical dependence among the resulting basis images. This use of PCA as a pre-processor in a two-step process allows ICA to create subspaces of size m for any m. In [30], it is also argued that pre-applying PCA enhances ICA performance by (1) discarding small trailing eigenvalues before whitening and (2) reducing computational complexity

by minimizing pair-wise dependencies. PCA decorrelates the input data; the remaining higher-order dependencies are separated by ICA.

To describe the mathematical basis for architecture I, let $\mathbf{R}$ be a p by m matrix containing the first m eigenvectors of a set of n face images, as in Section 2.2. Let p be the number of pixels in a training image. The rows of the input matrix to ICA are variables and the columns are observations, therefore, ICA is performed on $\mathbf{R}^{\mathbf{T}}$. The m independent basis images in the rows of $\mathbf{U}$ are computed as $\mathbf{U} = \mathbf{W} * \mathbf{R}^{\mathbf{T}}$. Then, the n by m ICA coefficients matrix $\mathbf{B}$ for the linear combination of independent basis images in $\mathbf{U}$ is computed as follows:

Let C be the n by m matrix of PCA coefficients. Then,

$$C = \mathbf{X} \ * \ \mathbf{R} \text{ and } \mathbf{X} = C * \mathbf{R}^{\mathbf{T}}.$$

From $\mathbf{U} = \mathbf{W} * \mathbf{R}^{\mathbf{T}}$ and the assumption that $\mathbf{W}$ is invertible we get

$$\mathbf{R}^{\mathbf{T}} = \mathbf{W}^{-1} * \mathbf{U}.$$

Therefore,

$$\mathbf{X} = (C * \mathbf{W}^{-1}) \ * \ \mathbf{U} = \mathbf{B} \ * \ \mathbf{U}.$$

Each row of $\mathbf{B}$ contains the coefficients for linearly combining the basis images to comprise the face image in the corresponding row of $\mathbf{X}$. Also, $\mathbf{X}$ is the reconstruction of the original data with minimum squared error as in PCA.

2.3.2. Architecture II: statistically independent coefficients

While the basis images obtained in architecture I are statistically independent, the coefficients that represent input images in the subspace defined by the basis images are not. The goal of ICA in architecture II is to find statistically independent coefficients for input data. In this architecture, the input is transposed from architecture I, that is, the pixels are variables and the images are observation. The source separation is performed on the pixels, and each row of the learned weight matrix $\mathbf{W}$ is an image. $\mathbf{A}$, the inverse matrix of $\mathbf{W}$, contains the basis images in its columns. The statistically independent source coefficients in $\mathbf{S}$ that comprise the input images are recovered in the columns of $\mathbf{U}$ (Fig. 4). This architecture was used in [9] to find image filters that produced statistically independent outputs from natural scenes.

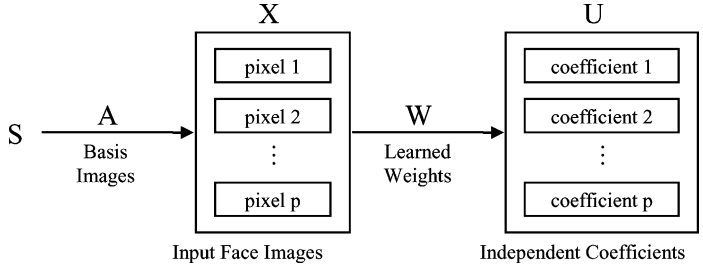


Fig. 4. Finding statistically independent coefficients.

The eight basis images shown in the bottom row of Fig. 3 show more global properties than the basis images produced in architecture I (middle row).

In this work, ICA is performed on the PCA coefficients rather than directly on the input images to reduce the dimensionality as in [4,5]. Following the same notation described above, the statistically independent coefficients are computed as $\mathbf{U} = \mathbf{W} * \mathbf{C}^{\mathbf{T}}$ and the actual basis images shown in Fig. 3 are obtained from the columns of $\mathbf{R} * \mathbf{A}$.

3. Recognizing facial identities

As the preceding discussion should make clear, comparing PCA to ICA in the context of face recognition is not a simple task. First one must refine the task: is the goal to recognize individuals, facial actions, gender or age? After defining the task, there are choices with regard to the ICA architecture, the ICA algorithm, the number of subspace dimensions, and the data set. We will show later that the choice of subspace distance metric is also significant, at least for PCA. This explains much of the confusion in the literature, as researchers have reported on different experiments, each of which can be thought of as one point in the space of PCA/ICA comparisons. In the paper, we try to fill out most of the space of PCA/ICA comparisons, relating previous studies to each other and giving a better picture of the relative performance of each algorithm. The rest of this section compares PCA to ICA on the task of recognizing facial identity; Section 4 compares PCA to ICA on the task of recognizing facial expressions.

3.1. The baseline system

Following the lead of Moon and Phillips [34], we compare PCA and ICA in the context of a baseline face recognition system. The baseline system matches novel probe images against a gallery of stored images. It does not do face detection or image registration; rather, it assumes that all images (probe and gallery) have been preregistered. For every probe image, there is one and only one gallery image of the same subject, so trials are binomial. If the first image retrieved from the gallery is of the same subject as the probe, then the trial is a success, otherwise it is a failure. The images are from the FERET face image data set [35], and have been reduced to 60×50 pixels. Examples are shown in Fig. 5.

Fig. 5. Sample images from the FERET data set.

The baseline system performs histogram normalization as a pre-processing step on all images (probe and gallery) using a routine provided by NIST [35]. During training, 500 training images are randomly selected from the gallery. This set of training images is provided as input to a subspace projection technique to form a set of basis vectors. The gallery images are then projected into these subspaces. During testing, probe images are projected into the same subspaces, and nearest neighbor retrieval is used to match compressed probe images to compressed gallery images.

3.2. The FERET face data

For consistency with other studies, we use the FERET face images, including the data partitions and terminology from [35]. As described there, the FERET gallery contains 1196 face images. There are four sets of probe images that are compared to this gallery: the *fb* probe set contains 1195 images of subjects taken at the same time as the gallery images. The only difference is that the subjects were told to assume a different facial expression than in the gallery image. The *duplicate I* probe set contains 722 images of subjects taken between one minute and 1031 days after the gallery image was taken. The *duplicate II* probe set is a subset of the duplicate I set, containing 234 images taken at least 18 months after the gallery image. Finally, the *fc* probe set contains 194 images of subjects under significantly different lighting.[3]

3.3. Comparison: InfoMax vs PCA

To compare PCA to ICA on the FERET facial identity task, we need to select an ICA algorithm, an ICA architecture, the number of subspace dimensions, and a subspace distance metric. Our first comparison uses InfoMax[4] to implement ICA. In keeping with a heuristic developed during the original FERET studies [34], we keep 200 subspace dimensions, or 40% of the maximum possible number of non-zero eigenvalues (given a training set of 500 images). This corresponds to keeping an average of 96% of the total variance in the eigenvalues for this data set.

Holding the ICA algorithm and the number of subspace dimensions constant, we test both ICA architectures and four possible distance metrics for PCA subspaces: the L1 (city-block) distance metric, the L2 (Euclidean distance) metric, cosine distance, and the Mahalanobis (L2 with each dimension scaled by the square root of its eigenvalue) distance metric. Since ICA basis vectors are not mutually orthogonal, the cosine distance measure is often used to retrieve images in the ICA subspaces (e.g. [2]). We report ICA results using both the cosine and L2 distance measures. Although not systematically presented here, we also tested ICA with the L1 and Mahalanobis distance measures. For ICA architecture II, the cosine measure clearly

[3] The image lists for the four probe sets and the gallery can be found at www.cs.colostate.edu/evalfacerec.

[4] With a block size of 50 and a learning rale starting at 0.001 and annealed over 1600 iterations to 0.0001. For architecture II, the learning rate began at 0.008.

outperforms all other measures. For ICA architecture I, cosine clearly outperforms L1; there is no significant difference between cosine and Mahalanobis.

Table 1 shows the recognition rate for PCA, ICA architecture I, and ICA architecture II, broken down according to probe set and distance metric. The most striking feature of Table 1 is that ICA architecture II with cosine distance measure always has the highest recognition rate. PCA is second, with both the L1 and Mahalanobis distance measures performing well. The performance of ICA architecture I and PCA with L2 and cosine distance measures are very close, and neither is competitive with architecture II/Cosine or PCA/L1 or PCA/Mahalanobis.

Are the findings that (1) ICA architecture II outperforms PCA/L1 and (2) PCA/L1 outperforms all other options statistically significant? To answer this question we must refine it. *Are these results significant with respect to the probe set size?* This question asks whether more probe images would have altered the results, and can be answered using McNemar's significance test for paired binomial values. By this measure, ICA architecture II outperforms PCA/L1 at a significance level (p-val) of .97, and PCA/L1 outperforms the other options at significance levels above .99.

Another question is: *are these results significant with respect to the choice of training set?* In other words, if we chose another set of 500 training images at random, would the results be the same? Beveridge et al. present a sophisticated resampling technique for answering this question with regard to the choice of gallery [11]. However, we are primarily interested in the variation induced by the choice of training set (not gallery), and it is not feasible to run InfoMax thousands of times. Instead, we compared PCA/L1 to ICA architectures I and II on ten randomly selected sets of 500 training images, evaluating each on all four probe sets as shown in Table 2. Significantly, the relative ranking of the algorithms never changes. Across ten trials and four probe sets, InfoMax architecture II (shown in the graph with $\times$'s, and labeled IM-2) always outperforms PCA/L1 (shown with squares and labeled PCA(L1)), which always outperforms InfoMax architecture I (IM-1; shown with triangles). We therefore believe that the relative rankings do not depend on the random choice of training images.

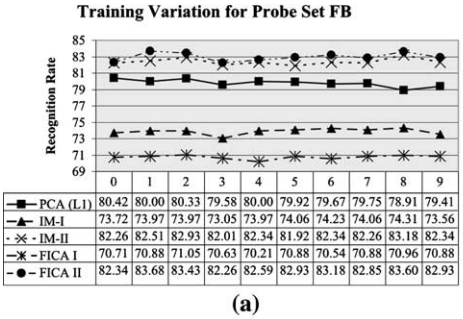
A third important question is: *do the results depend on the number of sub space dimensions?* Fig. 6 shows the performance results for PCA with all three distance

Table 1
Recognition rates for PCA and both architectures of ICA on the FERET face data set

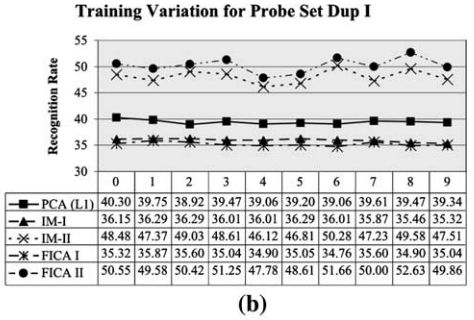
Probe st	ICA (InfoMax) Arch. 1		ICA (InfoMax) Arch. II		PCA			
	Cosine	L2	Cosine	L2	L1	L2	Cosine	Mahalanobis
fafb (1195)	73.72%	75.90%	82.26%	74.90%	80.42%	72.80%	70.71%	75.23%
fafc (194)	5.67%	5.15%	51.03%	35.57%	20.62%	4.64%	4.64%	39.69%
dup I (722)	36.15%	32.96%	48.48%	37.81%	40.30%	33.24%	35.32%	39.34%
dup II (234)	14.53%	14.53%	32.48%	25.64%	22.22%	14.53%	15.38%	24.36%
Total (2345)	50.62%	50.70%	64.31%	55.31%	57.31%	49.17%	48.83%	56.16%

The task is to match the identity of the probe image.

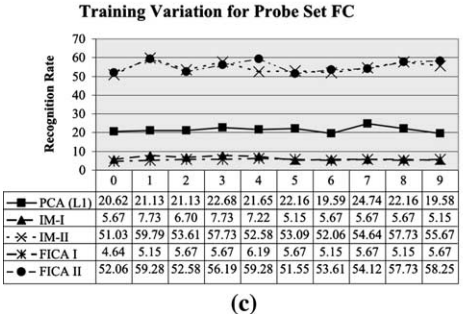
Table 2
Trails of PCA/L1, InfoMax Arch I, and InfoMax Arch II on ten sets of randomly selected training images



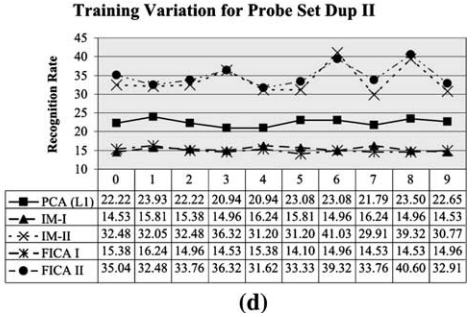
(a)



(b)



(c)



(d)

metrics and both architectures of ICA as the number of subspace dimensions varies from 50 to 200.[5] The relative ordering of the subspace projection techniques does not depend on the number of subspace dimensions, since for most techniques the lines never cross. The one exception is PCA with the Mahalonobis distance metric, which performs almost as well as ICA architecture II with small numbers of subspace dimensions, but whose performance on probe sets fb and dupI drops off relative to ICA architecture II and PCA with L1 as the number of subspace dimensions increases.

We conclude that for the task of facial identity recognition, ICA architecture II outperforms PCA with the L1 distance metric, which outperforms the other combinations, at least on the standard FERET probe sets. For small subspaces, the Mahalanobis metric can be used inplace of L1. Interestingly, this directly contradicts the claim made by Lee and Seung [27] (and echoed in [15,20,29]) that spatially localized feature vectors are better than overlapping (non-spatially localized) feature vectors for face recognition, since spatially localized feature vectors performed much worse than global features in our study.

[5] This data was collected using training set #1.

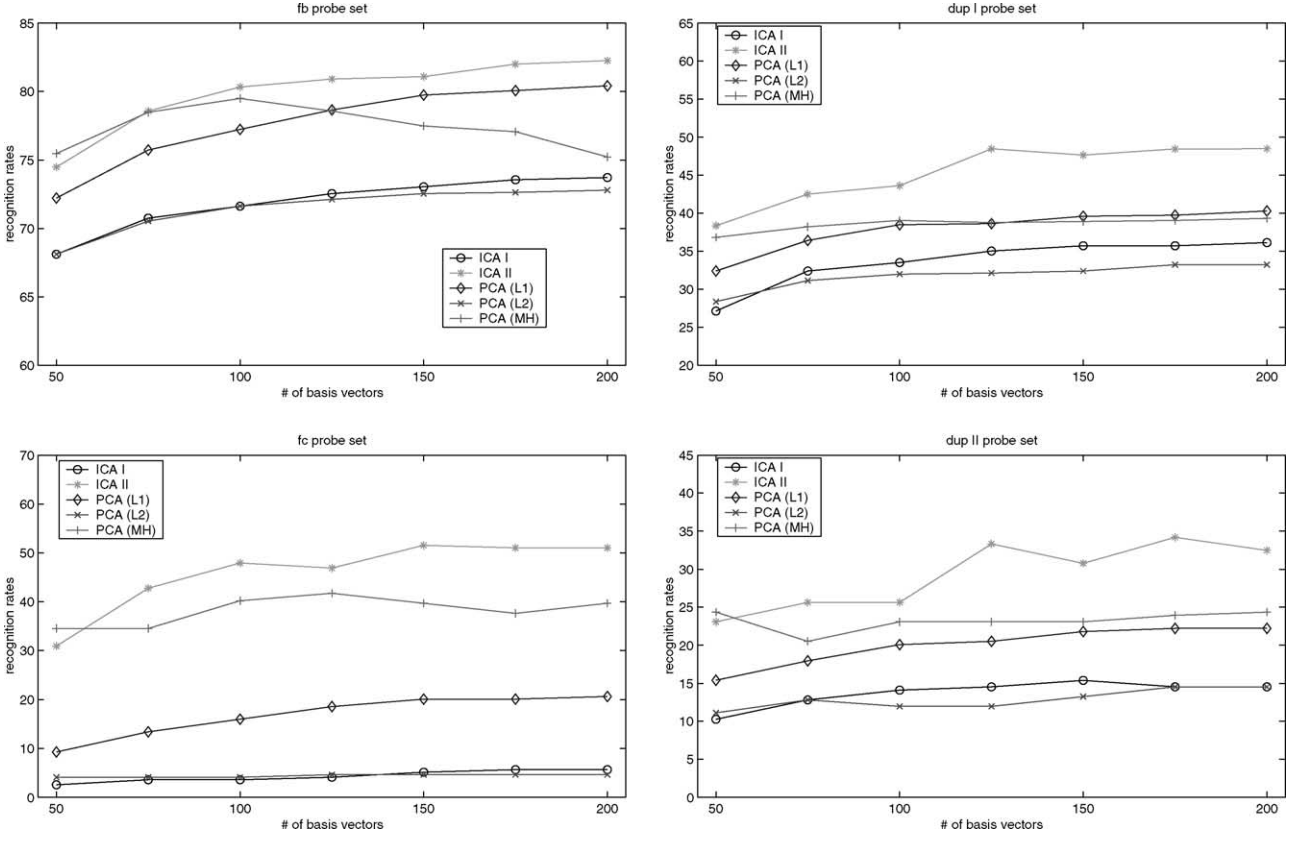


Fig. 6. Recognition rates for ICA architecture I (-○-), ICA architecture II (-∗-), and PCA with the L1 (-◇-), L2 (-×-) and Mahalanobis (-+-) distance measures as a function of the number of subspace dimensions. Each graph corresponds to a particular probe set (fb, dup I, dup II, or fc). Recognition rates were measured for subspace dimensionalities starting at 50 and increasing by 25 dimension up to a total of 200.

3.4. Comparisons to previous results

Does Table 1 contradict the previous results in the literature? In many cases no, if the ICA architecture and distance measures are taken into account. For example, Baek et al. found that PCA with the L1 distance measure outperformed ICA architecture I. This is consistent with Table 1. Liu and Wechsler [30] compared ICA architecture II to PCA with L2, and found that ICA was better. Again this agrees with Table 1. Similarly, Bartlett et al. [4,5] report that ICA architecture II outperforms PCA with L2, as predicted by Table 1.

Three papers compared ICA architecture I to PCA with the L2 distance metric, and found that ICA outperformed PCA [4,6,41] when cosines were used as the ICA subspace distance measure. Our study also finds that ICA architecture I is slightly better than PCA/L2, but it is not clear if this result is statistically significant. We did not pursue this question further, since architecture I is suboptimal for face recognition with ICA, and the L2 distance metric is suboptimal for face recognition with PCA.

The one previous result that contradicts ours is the study by Moghaddam [32] which found no statistically significant difference between ICA architecture II and PCA with the L2 distance metric. According to Table 1, ICA architecture II should have won this comparison, and the result should not have been in doubt. From the paper, however, it appears that Moghaddam may have used the L2 distance metric for ICA as well as for PCA. As shown in Table 1, this significantly lowers the performance of ICA architecture II relative to the performance using cosines, and may explain part of the discrepancy.

3.5. InfoMax vs. FastICA

When comparing ICA to PCA above, one of the fixed factors was the ICA algorithm. Would the results have been any different with another ICA algorithm, such as FastICA or JADE? Unfortunately, it is not feasible to run JADE on problems with more than 50 dimensions. We did, however, test the performance of FastICA[6] on both architectures, for the same ten randomly selected training sets. The results are shown back in Table 2, where the FastICA implementation of architecture I is labeled FICA-1 and marked graphically with a star, and the FastICA implementation of architecture II is labeled FICA-2 and marked with a circle. As shown there, the differences in performance between the two algorithms are relatively small, and depend on the architecture: FastICA performs better in architecture II, but InfoMax performs better in architecture I.

The choice of ICA algorithms therefore does not affect the ranking of PCA relative to the two ICA architectures. ICA architecture II was already better than PCA using InfoMax; switching to FastICA makes it better still. Similarly, PCA was already better than InfoMax in ICA architecture I; switching to FastICA makes the

[6] The FastICA implementation is due to Hyvärinen et al., and can be found at www.cis.hut.fi/projects/ica/fastica.

performance of ICA architecture I drop even lower. The relative rankings stay the same, no matter which ICA algorithm is used. The choice of ICA algorithm does affect performance, however. Users should use FastICA with architecture II, and InfoMax with architecture I, at least in this problem domain.

4. Recognizing facial actions

4.1. Methodology

The comparisons in Section 3 established an ordering for subspace projection techniques in the task of facial identity recognition. This section, however, will show that this order is highly task dependent. In fact, the order is reversed when recognizing facial actions.

Expressions are formed by moving localized groups of face muscles. We hypothesized, therefore, that localized features would be better for recognizing facial actions (and therefore emotions) than spatially overlapping features. We tested this hypothesis using the same baseline system as described in Section 3.1. This time, however, the images are of subjects performing specific facial actions. Ekman described a set of 46 human facial actions [18], and Ekman and Hager collected data for 20 subjects performing a total of 80 subject/action pairs. A temporal sequence of five images was taken of each subject performing each action, where the first image shows only small movements at the beginning of action. The movements become successively more pronounced in images 2 through 5. All five images are then subtracted from the picture of the same subject with a neutral expression, prior to the start of the facial action. The resulting data is composed of difference images, as shown in Fig. 7. (This data is further described in [3]). For the experiments in this paper, only the upper halves of the faces were used.

One difference between Ekman and Hagar's facial action data set and the FERET gallery is that the facial action data set contains many examples of each facial action. As a result, the facial action gallery can be divided into categories (one per action),

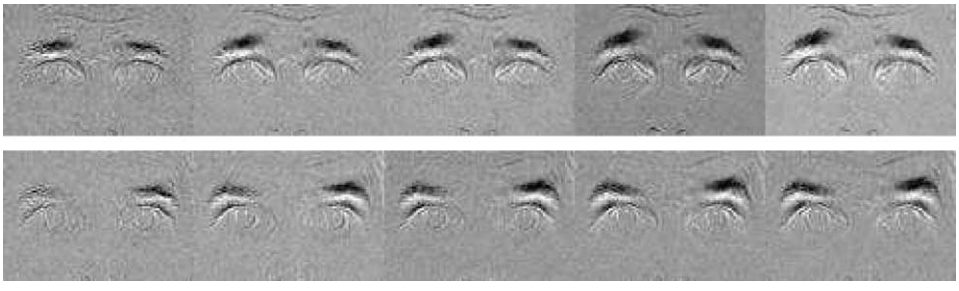


Fig. 7. Sequences of difference images for Action Unit 1 and Action Unit 2. The frames are arranged temporally left to right, with the left most frame being the initial stage of the action, and the right most frame being its most extreme form.

and the class discriminability r can be used to order the ICA basis vectors[7] [3]. To compute r, each training image is assigned a facial action class label. Then, for each feature, the between-class variability V_{between}, and within-class variability, V_{within} of the corresponding coefficients are computed by:

$$V_{\text{between}} = \sum_i (M_i - M)^2,$$

$$V_{\text{within}} = \sum_i \sum_j (b_{ij} - M_i)^2,$$

where M is the overall mean of coefficients across the training images, M_i is the mean for class i, b_{ij} is coefficient of jth training image in class i, and r is the ratio of V_{between} to V_{within}, i.e., $r = V_{\text{between}}/V_{\text{within}}$. This allows us to rank the ICA features in terms of discriminability and to plot recognition rate as a function of the number of subspace dimensions. We create the same plot for PCA, ordering its features according to class discriminability (Fig. 8, left) and according to the eigenvalues (Fig. 8, right).

For testing purposes, the subjects were partitioned into four sets. The four sets were designed to keep the total number of actions as even as possible, given that not all subjects performed all actions. (For anyone who wishes to replicate our experiments, the exact partition by subject index is given in Appendix A.) Otherwise, the methodology and parameters were the same as described in Section 3.1, except that in this experiment a trial was a success if the retrieved gallery image was performing the same facial action as the probe image. We performed the experiments twice, once restricting the probe set to only the third image in each temporal action sequence (as in [3]), and once allowing all five images to be used as probes. Once again, ICA architecture I, ICA architecture II, and PCA with the L1, L2 and Mahalanobis distance metrics were compared.

4.2. Results and discussion

The results of this experiment are complex, in part because the recognition rate does not increase monotonically with the number of subspace dimensions. Fig. 8 shows the recognition rate as a function of the number of dimensions for each of the five techniques, averaged across four test sets. For every technique, the maximum recognition rate occurs at a different number of subspace dimensions on every test set, suggesting that it may not be possible to tune the techniques by changing the number of subspace dimensions. Table 3 therefore presents the recognition rates for every technique using 110 basis vectors. Appendix B gives the same results when only the third images from each temporal sequence are used as probes, as in [3,17] (the results are essentially equivalent).

[7] We did not order ICA basis vectors by relevance for the face recognition task because the FERET data set contains only two head-on images of most subjects, and no more than four head-on images of any subject.

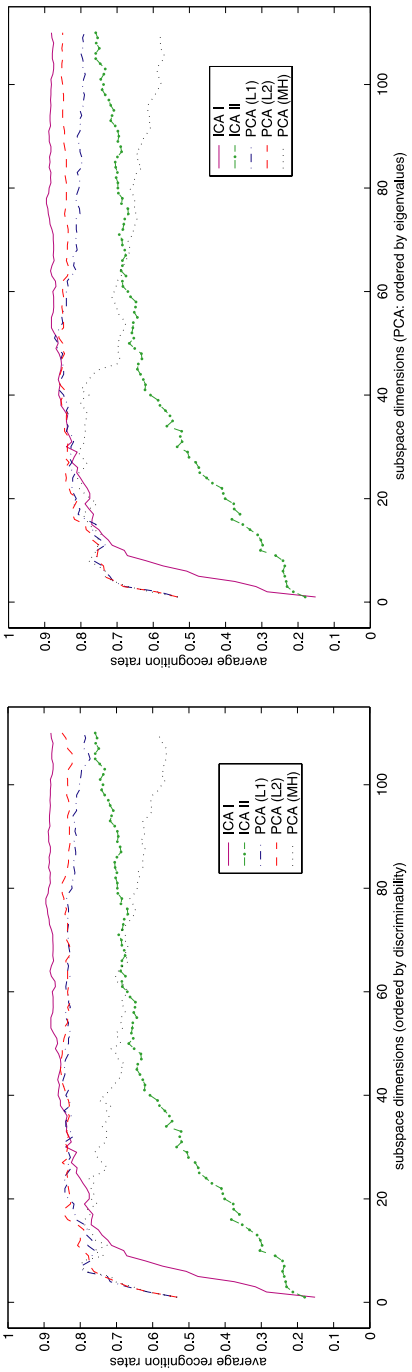


Fig. 8. Recognition rates vs. subspace dimensions. On the left, both ICA and PCA components are ordered by the class discriminability while PCA components are ordered according to the eigenvalues in the right plot. ICA architecture I is (—), ICA architecture II is (–•–), PCA with L1 is (— · —), PCA with L2 is (– – –), PCA with Mahalanobis is (· · · · · ·).

Table 3
Average recognition rates for facial actions using ICA and PCA

Test set	110 Basis vctors				
	ICA I	ICA II	PCA(L1)	PCA(L2)	Mahalanobis
1	87.37%	74.73%	83.16%	83.16%	52.63%
2	95.79%	84.21%	84.21%	92.63%	70.53%
3	82.50%	69.00%	80.00%	83.00%	53.00%
4	87.27%	75.91%	70.00%	81.82%	58.18%
Avg.	88.12%	75.87%	79.00%	85.00%	58.50%

The images were divided into four sets according to Appendix B and evaluated using 4-fold cross validation. Techniques were evaluated by testing from 1 to 110 subspace dimensions and taking the average.

When recognizing facial actions, it is ICA architecture I that outperforms the other four techniques. PCA/L2 is second best, followed by PCA/L1, ICA architecture II, and PCA/Mahalanobis. This is consistent with the hypothesis that spatially localized basis vectors outperform spatially overlapping basis vectors for recognizing facial actions, and underscores the point that the analysis technique must be selected based on the recognition task. It is also consistent with the data in [3,17].

5. Conclusion

Comparisons between PCA and ICA are complex, because differences in tasks, architectures, ICA algorithms, and distance metrics must be taken into account. This is the first paper to explore the space of PCA/ICA comparisons, by systematically testing two ICA architectures, two ICA algorithms, and three PCA distance measures on two tasks (facial identity and facial expression). In the process, we were able to verify the results of previous comparisons in the literature, and to relate them to each other and to this work.

In general, we find that the most important factor is the nature of the task. Some tasks, like facial identity recognition, are holistic and require spatially overlapping feature vectors. Other tasks, like facial action recognition, are more localized, and perform better with spatially disjoint feature vectors.

For the facial identity task, we find that ICA architecture II provides the best results, followed by PCA with the L1 or Mahalanobis distance metrics. The difference between PCA and ICA in this context is not large, but it is significant. ICA architecture I and PCA with the L2 or cosine metrics are poorer choices for this task. If ICA architecture II is used, we recommend using the FastICA algorithm, although the difference between FastICA and InfoMax is not large. For the more localized task of recognizing facial actions, the recommendation is reversed: we found the best results using InfoMax to implement ICA architecture I. PCA with the L2 metric was the second most effective technique.

As with any comparative study, there are limitations. We did not test the effects of registration errors or image pre-processing schemes. We tested only two ICA

algorithms. We evaluated PCA and ICA according to only one criterion, recognition rate, even though other criteria such as computational cost may also apply. Most significantly, although we measured the effects of different ICA algorithms and PCA distance metrics, we cannot explain these differences in terms of the underlying data distributions. As a result, it is difficult to predict the best technique for a novel domain. We consider understanding this relationship to be future work both for the authors and for others in the field of appearance-based object recognition.

Acknowledgments

We would like to thank Paul Ekman and Joe Hager of UCSF who collected the facial expression data set. This work was supported in part by DARPA under contract DABT63-00-1-0007 the Office of Naval Research under contract ONR N0014-02-1-0616, and the National Science Foundation under Grant NSF IIS 0220141.

Appendix A. Partition of facial action data set

As discussed in Section 4.1, not all 20 subjects were capable of performing all six facial actions in isolation. As a result, there are 80 subject/action pairs, not 120. Therefore, when dividing subjects into four partitions, we had to ensure that all four partitions had roughly the same number of subject/action pairs, and that every action was equally represented in each partition. We partitioned subjects as follows:

Subject partition for experiments in Section 4. Each row corresponds to a facial action, each column to a set of subjects. Table entries correspond to the number of subject/action pairs in a partition for the corresponding facial action

	Partition (subject #)			
AU #	1 (0,2,4,5,14)	2 (1,7,8,9,16)	3 (3,10,11,12,13)	4 (6,17,18,19,20)
1	2	2	2	3
2	2	3	2	3
4	5	4	4	5
5	5	5	5	5
6	1	1	2	1
7	4	4	5	5
Total	19 actions	19 actions	20 actions	22 actions

Appendix B. More recognition results for facial action sequences

Previous studies with Ekman and Hager's facial action data set have classified the third image in each sequence [3,17]. In so doing, they avoided testing on the smaller

facial actions in the first image and the largest facial actions in the fifth image. For this main body of this paper, we used images from all five stages as probe images. To compare our results to previous papers, however, we also tested just the third image from each sequence. The results are shown in the table below. Readers can verify that there is little significant difference between this table and Table 3.

Performance of ICA architecture 1 (ICA I), ICA architecture II (ICA II), and PCA with the L1, L2 and Mahalanobis distance functions (PCA(L1), PCA(L2), and PCA(Mh), respectively), when classifying just the third frame of each action unit sequence, as in [3,17]. The results are qualitatively equivalent to those in Table 3

Test set	110 Basis vectors				
	ICA I	ICA II	PCA(L1)	PCA(L2)	PCA (Mh)
1	94.73%	86.84%	84.21%	84.21%	52.63%
2	100.0%	94.74%	84.21%	94.74%	73.68%
3	85.00%	70.00%	85.00%	85.00%	55.00%
4	86.36%	75.00%	72.73%	81.82%	59.09%
Avg	91.25%	81.25%	81.25%	86.25%	60.00%

References

[1] K. Baek, B.A. Draper, J.R. Beveridge, K. She, PCA vs ICA: A comparison on the FERET data set, presented at Joint Conference on Information Sciences, Durham, NC, 2002.

[2] M.S. Bartlett, Face Image Analysis by Unsupervised Learning, Kluwer Academic, Dordrecht, 2001.

[3] M.S. Bartlett, G. Donato, J.R. Movellan, J.C. Hager, P. Ekman, T.J. Sejnowski, Image representations for facial expression coding, in: Advances in Neural information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000, pp. 886–892.

[4] M.S. Bartlett, H.M. Lades, T.J. Sejnowski, Independent component representations for face recognition, presented at SPIE Symposium on Electronic Imaging: Science and Technology, Conference on Human Vision and Electronic Imaging III, San Jose, CA, 1998.

[5] M.S. Bartlett, J.R. Movellan, T.J. Sejnowski, Face recognition by independent component analysis, IEEE Transaction on Neural Networks 13 (2002) 1450–1464.

[6] M.S. Bartlett, T.J. Sejnowski, Viewpoint invariant face recognition using independent component analysis and attractor networks, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Neural Information Processing Systems – Natural and Synthetic, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 817–823.

[7] P. Belhumeur, J. Hespanha, D. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transaction on Pattern Analysis and Machine Intelligence 19 (1997) 711–720.

[8] A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution, Neural Computation 7 (1995) 1129–1159.

[9] J.A. Bell, T.J. Sejnowski, The 'Independent Components' of natural scenes are edge filters, Vision Research 37 (1997) 3327–3338.

[10] J.R. Beveridge, The Geometry of LDA and PCA Classifiers Illustrated with 3D Examples, Colorado State University, web page 2001.

[11] J.R. Beveridge, K. She, B.A. Draper, G.H. Givens, A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition, presented at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.

[12] I. Biederman, P. Kalocsai, Neurocomputational bases of object and face recognition, Philosophical Transactions of the Royal Society: Biological Sciences 352 (1997) 1203–1219.

[13] W.W. Bledsoe, The model method in facial recognition, Panoramic Research, Inc., Palo Alto, CA PRI:15, August 1966.
[14] J.-F. Cardoso, Infomax and maximum likelihood for source separation, IEEE Letters on Signal Processing 4 (1997) 112–114.
[15] X. Chen, L. Gu, S.Z. Li, H.-J. Zhang, Learning Representative Local Features for Face Detection, presented at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.
[16] G.W. Cottrell and M.K. Fleming, Face recognition using unsupervised feature extraction, presented at International Neural Network Conference, Dordrecht, 1990.
[17] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, T. Sejnowski, Classifying facial actions, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (1999) 974–989.
[18] P. Ekman, W. Friesen, Facial Action Coding System: A Technique for the Measurement of Facial Movement, Consulting Psychologists Press, Palo Alto, CA, 1978.
[19] B.J. Frey, A. Colmenarez, T.S. Huang, Mixtures of Local Linear Subspaces for Face Recognition, presented at IEEE Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, 1998.
[20] D. Guillamet, M. Bressan, J. Vitrià, A Weighted Non-negative Matrix Factorization for Local – Representations, presented at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.
[21] A. Hyvärinen, The fixed-point algorithm and maximum likelihood estimation for independent component analysis, Neural Processing Letters 10 (1999) 1–5.
[22] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, Wiley, New York, 2001.
[23] P. Kalocsai, H. Neven, J. Steffens, Statistical Analysis of Gabor-filter Representation, presented at IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998.
[24] N. Kambhatla, T.K. Leen, Dimension reduction by local PCA, Neural Computation 9 (1997) 1493–1516.
[25] J. Karvanen, J. Eriksson, V. Koivunen, Maximum Likelihood Estimation of ICA-model for Wide Class of Source Distributions, presented at Neural Networks in Signal Processing, Sydney, 2000.
[26] M. Kirby, L. Sirovich, Application of the Karhunen-Loeve procedure for the characterization of human faces, IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (1990) 103–107.
[27] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791.
[28] T.-W. Lee, T. Wachtler, T.J. Sejnowski, Color oppency is an efficient representation of spectral properties in natural scenes, Vision Research 42 (2002) 2095–2103.
[29] S.Z. Li, X. Hou, H. Zhang, Q. Cheng, Learning Spatially Localized, Parts-Based Representation, presented at IEEE Conference on Computer Vision and Pattern Recognition, Kauai, HI, 2001.
[30] C. Liu and H. Wechsler, Comparative Assessment of Independent Component Analysis (ICA) for Face Recognition, presented at International Conference on Audio and Video Based Biometric Person Authentication, Washington, DC, 1999.
[31] A.M. Martinez, A.C. Kak, PCA versus LDA, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (2001) 228–233.
[32] B. Moghaddam, Principal Manifolds and Bayesian Subspaces for Visual Recognition, presented at International Conference on Computer Vision, Corfu, Greece, 1999.
[33] B. Moghaddam, A. Pentland, Beyond Eigenfaces: Probabilistic Matching for Face Recognition, presented at International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998.
[34] H. Moon, J. Phillips, Analysis of PCA-based face recognition algorithms, in: K. Boyer, J. Phillips (Eds.), Empirical Evaluation Techniques in Computer Vision, IEEE Computer Society Press, Los Alamitos, CA, 1998.
[35] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET evaluation methodology for face-recognition algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 1090–1104.
[36] L. Sirovich, M. Kirby, A low-dimensional procedure for the characterization of human faces, Journal of the Optical Society of America 4 (1987) 519–524.

[37] D. Socolinsky and A. Selinger, A Comparative Analysis of Face Recognition Performance with Visible and Thermal Infrared Imagery, presented at International Conference on Pattern Recognition, Quebec City, 2002.
[38] D. Swets, J. Weng, Using discriminant eigenfeatures for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 831–836.
[39] M.E. Tipping, C.M. Bishop, Mixtures of probabilistic principal component analysers, Neural Computation 11 (1999) 443–482.
[40] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1991) 71–86.
[41] P.C. Yuen, J.H. Lai, Independent Component Analysis of Face Images, presented at IEEE Workshop on Biologically Motivated Computer Vision, Seoul, 2000.
[42] M. Zibulevsky, B.A. Pearlmutter, Blind Separation of Sources with Sparse Representation in a Given Signal Dictionary, presented at International Workshop on Independent Component Analysis and Blind Source Separation, Helsinki, 2000.
[43] A. Ziehe, G. Nolte, T. Sander, K.-R. Müller, G. Curio, A Comparison of ICA-based Artifact Reduction Methods for MEG, presented at 12th International Conference on Biomagnetism, Espoo, Finland, 2000.