Analyzing PCA-based Face Recognition Algorithms: Eigenvector Selection and Distance Measures

Wendy S. Yambor, Bruce A. Draper, and J. Ross Beveridge[1]

Computer Science Department
Colorado State University
Fort Collins, CO, U.S.A 80523
yambor,draper,ross@cs.colostate.edu

Abstract. This study examines the role of Eigenvector selection and Eigenspace distance measures on PCA-based face recognition systems. In particular, it builds on earlier results from the FERET face recognition evaluation studies, which created a large face database (1,196 subjects) and a baseline face recognition system for comparative evaluations. This study looks at using a combinations of traditional distance measures (City-block, Euclidean, Angle, Mahalanobis) in Eigenspace to improve performance in the matching stage of face recognition. A statistically significant improvement is observed for the Mahalanobis distance alone when compared to the other three alone. However, no combinations of these measures appear to perform better than Mahalanobis alone. This study also examines questions of how many EIgenvectors to select and according to what ordering criterion. It compares variations in performance due to different distance measures and numbers of Eigenvectors. Ordering Eigenvectors according to a like-image difference value rather than their Eigenvalues is also considered.

1 Introduction

Over the past few years, several face recognition systems have been proposed based on principal components analysis (PCA) [14, 8, 13, 15, 1, 10, 16, 6]. Although the details vary, these systems can all be described in terms of the same preprocessing and run-time steps. During preprocessing, they register a gallery of m training images to each other and unroll each image into a vector of n pixel values. Next, the mean image for the gallery is subtracted from each and the resulting "centered" images are placed in a gallery matrix M. Element $[i, j]$ of M is the ith pixel from the jth image.

A covariance matrix $\Omega = M\,M^T$ characterizes the distribution of the m images in $\Re^n$. A subset of the Eigenvectors of Ω are used as the basis vectors for a subspace in which to compare gallery and novel probe images. When sorted by decreasing Eigenvalue, the full set of unit length Eigenvectors represent an orthonormal basis where the first direction corresponds to the direction of maximum variance in the images, the second the next largest variance, etc. These

basis vectors are the Principle Components of the gallery images. At the time the Eigenspace is being computed, the centered gallery images are projected into this subspace. At run-time, recognition is accomplished by projecting a centered probe image into the subspace and the nearest gallery image to the probe image is selected as its match.

There are many differences in the systems referenced. Some systems assume that the images are registered prior to face recognition [15, 10, 11, 16]; among the rest, a variety of techniques are used to identify facial features and register them to each other. Different systems may use different distance measures when matching probe images to the nearest gallery image. Different systems select different numbers of Eigenvectors (usually those corresponding to the largest k Eigenvalues) in order to compress the data and to improve accuracy by eliminating Eigenvectors corresponding to noise rather than meaningful variation.

To help evaluate and compare individual steps of the face recognition process, Moon and Phillips created the FERET face database, and performed initial comparisons of some common distance measures for otherwise identical systems [10, 11, 9]. This paper extends their work, presenting further comparisons of distance measures over the FERET database and examining alternative way of selecting subsets of Eigenvectors.

2 The FERET database

For readers who are not familiar with it, the FERET database contains images of 1,196 individuals, with up to 5 different images captured for each individual. The images are separated into two sets: gallery images and probes images. Gallery images are images with known labels, while probe images are matched to gallery images for identification. The database is broken into four categories:

FB Two images were taken of an individual, one after the other. One image is of the individual with a neutral facial expression, while the other is of the individual with a different expression. One of the images is placed into the gallery file while the other is used as a probe. In this category, the gallery contains 1,196 images and the probe set has 1,195 images.

Duplicate I The only restriction of this category is that the gallery and probe images are different. The images could have been taken on the same day or a year apart. In this category, the gallery consists of the same 1,196 images as the FB gallery while the probe set contains 722 images.

fc Images in the probe set are taken with a different camera and under different lighting than the images in the gallery set. The gallery contains the same 1196 images as the FB & Duplicate I galleries, while the probe set contains 194 images.

Duplicate II Images in the probe set were taken at least 1 year after the images in the gallery. The gallery contains 864 images, while the probe set has 234 images.

This study uses FB, Duplicate I and Duplicate II images.

Table 1. Percent of probe images correctly recognized for combined classifiers: a) base distance measures and summed combinations on Duplicate I images, b) classifiers using bagging on Duplicate I and FB images.

Classifier	Dup. I
L1	35
L2	33
Angle	34
Mahalanobis	42
S(L1, L2)	35
S(L1, Angle)	39
S(L1, Mahalanobis)	43
S(L2, Angle)	33
S(L2, Mahalanobis)	42
S(Angle, Mahalanobis)	42
S(L1, L2, Angle)	35
S(L1, L2, Mahalanobis)	42
S(L1, Angle, Mahalanobis)	43
S(L2, Angle, Mahalanobis)	42
S(L1, L2, Angle, Mahalanobis)	42

(a)

Classifier	Dup. I	FB
L1	35	77
L2	33	72
Ang	34	70
Mah	42	74
Bagging	37	75
Bagging, Best 5	38	78
Bagging, Weighted	38	77

(b)

3 Distance Measures

In [9], Moon and Phillips look at the effect of four traditional distance measures in the context of face recognition: city-block (L1 norm), squared Euclidean distance (L2 norm), angle, and the Mahalanobis distance. The appendix in [9] formally defines these distance measures and for completeness these definitions are repeated here in Appendix A. There was one minor problem encountered with the definition of Mahalanobis distance given in [9] and this is discussed in Appendix A.

This paper presents further evaluations of traditional distance measures in the context of face recognition. In particular, we considered the hypothesis that some combination of the four standard distance measures (L1, L2, angle and Mahalanobis) might outperform the individual distance measures. To this end, we test both simple combinations of the distance measures, and "bagging" the results of two or more measures using a voting scheme [2, 4, 7].

3.1 Adding Distance Measures

The simplest mechanism for combining distance measures is to add them. In other words, the distance between two images is defined as the sum S of the distances according to two or more traditional measures:

$$S(a_1, \ldots, a_h) \;=\; a_1 + \ldots + a_h \tag{1}$$

Using S, all combinations of the base metrics (L1, L2, angle, Mahalanobis) were used to select the nearest gallery image to each probe image. The percentage of

images correctly recognized using each combination is shown in Table 1a, along with the recognition rates for the base measures themselves.

Of the four base distance measures, there appears to be a significant improvement with Mahalanobis distance. On the surface, 42% seems much better than 33%, 34% or 35%. The best performance for any combined measure was 43% for the S(L1, Angle, Mahalanobis) combination. While higher, this does not appear significant. The question of when a difference is significant will be taken up more formally in Section 3.4.

Interestingly, the performance of the combined measures were never less than the performance of their components evaluated separately. For example, the performance of S(L1, L2) is 35%; this is better than the performance of L2 (33%) and the same as L1 (35%).

3.2 Distance Measure Aggregation

The experiment above tested only a simple summation of distance measures; one can imagine many weighting schemes for combining distance measures that might outperform simple summation. Rather than search the space of possible distance measure combinations, however, we took a cue from recent work in machine learning that suggests the best way to combine multiple estimators is to apply each estimator independently and combine the results by voting [2, 4, 7].

For face recognition, this implies that each distance measure is allowed to vote for the image that it believes is the closest match to a probe. The image with the most votes is chosen as the matching gallery image. Voting was performed three different ways.

Bagging Each classifier is given one vote as explained above.

Bagging, Best of 5 Each classifier votes for the five gallery images that most closely match the probe image.

Bagging, Weighted Classifiers cast five votes for the closest gallery image, four votes for the second closest gallery image, and so on, casting just one vote for the fifth closest image.

Table 1b shows the performance of voting for the Duplicate I and FB probe sets. On the Duplicate I data, Mahalanobis distance alone does better than any of the bagged classifiers: 42% versus 37% and 38%. On the simpler FB probe set, the best performance for a separate classifier is 77% (for L1) and the best performance for the bagged classifiers is 78%: not an apparently significant improvement. In the next section we explore one possible explanation for this lack of improvement when using bagging.

3.3 Correlating Distance Metrics

As described in [2], the failure of voting to improve performance suggests that the four distance measures share the same bias. To test this theory, we correlated the distances calculated by the four measures over the Duplicate I probe set. Since

Table 2. Correlation of classifiers on the duplicate I probe set.

	L1-L2	L1-Angle	L1-Mah.	L2-Angle	L2-Mah.	Angle-Mah.
Rank Correlation	0.46	0.39	0.39	0.62	0.50	0.58

Table 3. Images that were poorly identified by the classifiers.

	Classifiers in Error			
	1	2	3	4
Images	46 (26%)	48 (27%)	34 (19%)	51 (28%)

each measure is defined over a different range, Spearman Rank Correlation was used [12]. For each probe image, the gallery images were ranked by increasing distance to the probe image. This is done for each pair of distance measures. The result is two rank vectors, one for each distance measure. Spearman's Rank Correlation is the correlation coefficient for these two vectors.

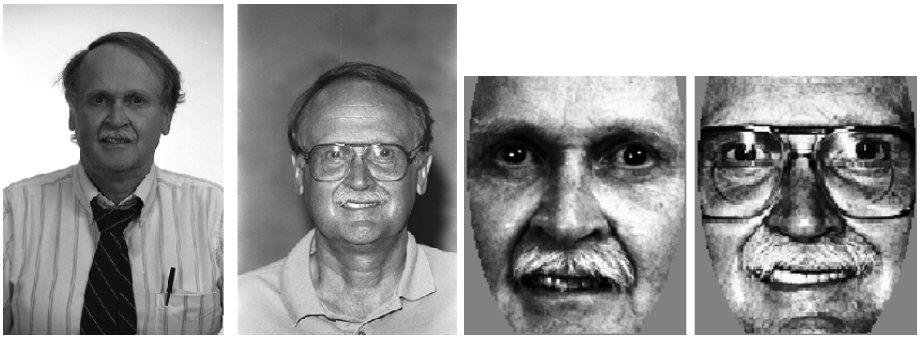
Table 2 presents the average correlation coefficient over all probe images for pairs of distance measures. Ranks based upon L2, Angle and Mahalanobis all correlate very closely to each other, although L1 correlates less well to Angle and Mahalanobis. This suggests that there might be some advantage to combining L1 with Angle or Mahalanobis, but that no combination of L2, Angle or Mahalanobis is very promising. This is consistent with the scores in Table 1a, which show that the combination of L1 & Angle and L1 & Mahalanobis outperform these classifiers individually.

We also constructed a list of images in the FB probe set that were grossly misclassified, in the sense that the matching gallery image was not one of the ten closest images according to one or more of the distance measures. A total of 179 images were poorly identified by at least one distance measure. To illustrate how difficult some of these recognition problems are, Figure 1 shows two images of the same individual that were consistently misclassified. Seeing these images, it is easy to see why classification algorithms have difficulty with this pair.

Table 3 shows the number of images that were poorly identified by one distance measure, two distance measures, three distance measures and all four distance measures. This table shows that there is shared bias among the classifiers, in that they seem to make gross mistakes on the same images. On the other hand, the errors do not overlap completely, suggesting that some improvement might still be achieved by some combination of these distance measures.

3.4 When is a Difference Significant

The data in Table 1 begs the question of when a difference in performance is significant. Intuitively, a 1% difference seems likely to arise by chance, while a 10% difference does not. However, to move beyond intuition requires that we formulate a precise hypothesis that can be evaluated using standard statistical hypothesis testing techniques. Moreover, even for such an apparently simple comparison as presented in Table 1, there are at least two very distinct ways



(a) (b)

Fig. 1. Individual identified poorly in the duplicate I probe set. a) raw image, b) normalized.

to elaborate the question. First, is the difference as seen over the entire set of probe images significant. Second, when the algorithms behave differently, is the difference significant.

Let us approach the question of significance over the entire sample first. In keeping with standard practice for statistical hypothesis testing, we must formulate our hypothesis and the associated null hypothesis. For simplicity, we will speak of each variant as a different algorithm. For example, when comparing the standard PCA classifier using the L1 distance to the standard PCA classifier using the L2 distance, the first may be designated algorithm A and the other algorithm B.

H1 Algorithm A correctly classifies images more often than does algorithm B.
H0 There is no difference in how well Algorithms A and B classify images.

To gain confidence in H1, we need to establish that the probability of H0 is very small given our observation. The mechanics of how this is done appear in standard Statistics texts [3]. We review the testing procedure in Appendix 7, Section 7.1. Briefly, based upon the number of times each algorithm is observed to succeed, a normalized variable z is computed. The probability the null hypothesis is true, P_{HO}, is determined from z.

The z values and probabilities are shown in Table 4a for the six pairwise comparisons between the four base distance measures: L1, L2, Angle and Mahalanobis. The actual number of images correctly identified for these four is 253, 239, 246 and 305 respectively. There are a total of 722 images. A common cutoff for rejecting $H0$ is $P_{H0} < 0.05$. Using this cutoff, the only statistically significant differences are between Mahalanobis and the others.

The test of significance just developed is useful, but it fails to take full advantage of our experimental protocol. Specifically, the tests of the different distance measures are not on independently sampled images from some larger population, but are instead on the same set of sampled images. This observation leads us to

Table 4. Results of testing for statistical significance in the comparison of the four base distance measures: a) treating probe images as independent samples for each measure and testing significance over all samples, b) scoring paired tests into four outcomes and performing a Sign Test on only those where performance differs.

Measures	Variable z	$P_{H0} \leq$
L1 L2	0.777	0.21848
L1 Angle	0.387	0.34924
Mah. L1	2.810	0.00247
Angle L2	0.390	0.34826
Mah. L2	3.584	0.00017
Mah. Angle	3.196	0.00070

(a)

Measures	Outcomes SS	SF	FS	FF	$P_{H0} \leq$
L1 L2	219	34	20	449	0.038
L1 Angle	214	39	32	437	0.238
Mah. L1	220	85	33	384	9.2E-07
Angle L2	224	22	15	461	0.162
Mah. L2	214	91	25	392	2.7E-10
Mah. Angle	225	80	21	396	1.4E-09

(b)

the second and more discriminating question and associated test. Our hypotheses in this case become

H1 When algorithm's A and B differ on a particular image, algorithm A is more likely to correctly classify that image.

H0 When algorithm's A and B differ on a particular image, each is equally likely to classify the image correctly.

To test this hypothesis, we need to record which or four possible outcomes are observed for each probe image:

SS Both algorithms successfully classify the image.

SF Algorithm A successfully classifies the image, algorithm B fails.

FS Algorithm A fails to classify the image correctly, algorithm B succeeds.

FF Both algorithms fail to classify the image correctly.

The number of times each outcomes is observed for all pairs of the four base distance measures is shown in Table 4b.

The test we use to bound the probability of H0 is called McNemar's Test, which actually simplifies to a Sign Test [5]. We review the testing procedure in Appendix 7, Section 7.2. The probability bounds for each pairwise comparison are shown in the last column of Table 4b. Observe how much more conclusively the null hypothesis can be rejected for those pairs including Mahalanobis distance. Also observe that the L1 L2 comparison is now technically significant if a 0.05 cutoff is assumed. In other words, $H0$ would be rejected with $P_{H0} \leq 0.038$. What we are seeing is that this test is much more sensitive to differences in the possible outcomes of the two algorithms.

Which tests just is preferable depends the question one really wants answered. In the first case, the overall difference in performance on all the data is taken into consideration. This is appropriate if the goal is to draw conclusion about performance over all the problems. However, if the goal is to spotlight differences in performance where such differences arise, the second is more appropriate.

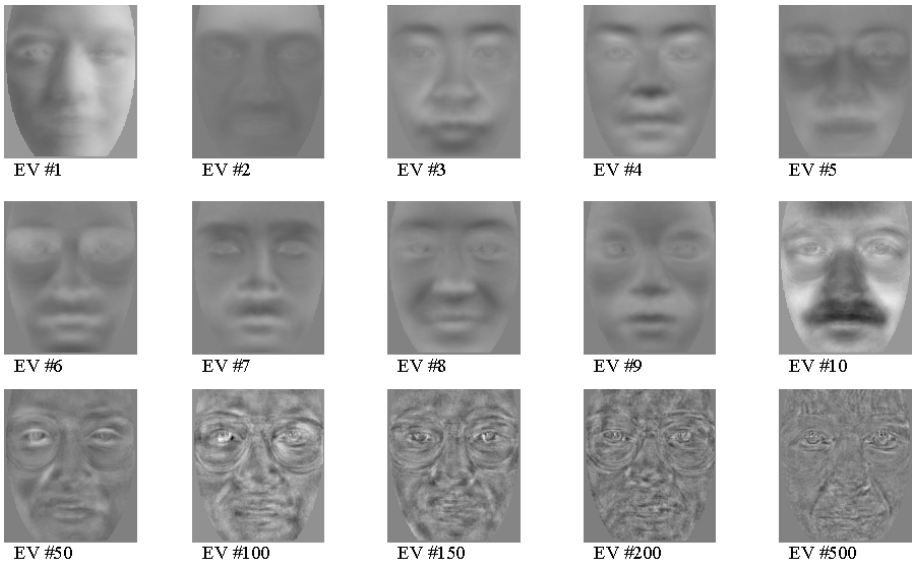


Fig. 2. Eigenvectors 1-10, 50, 100, 150, 200 and 500

4 Selecting Eigenvectors

In the FERET database, images may vary because of differences in illumination, facial expression, clothing[1], presence and/or style of glasses, and even small changes in viewpoint, none of which are relevant to the task of identifying the image subject. The problem, of course, is knowing which Eigenvectors correspond to useful information and which are simply meaningless variation.

By looking at the images of specific Eigenvectors, it is sometimes possible to determine what features are encoded in that Eigenvector. Images of the Eigenvectors used in the FERET evaluation are shown in Figure 3, ordered by Eigenvalue. Eigenvector 1 seems to encode lighting from right to left, while Eigenvector two apparently encodes lighting from top to bottom. Since the probe and gallery images of a single subject may not have the same lighting, it is reasonable to assume that removing these Eigenvectors might improve performance. Results from the FERET evaluation [9] verify this assumption.

Other Eigenvectors also appear to encode features. For example, Eigenvector 6 clearly shows glasses. As we examine the higher order Eigenvectors (100, 150, 200, 500), they become more blotchy and it becomes difficult to discern the semantics of what they are encoding. This indicates that eliminating these Eigenvectors from the Eigenspace should have only a minimal effect on perfor-

[1] In the FERET database, background, clothing and hair are eliminated as much as possible during image registration. Unfortunately, shirt collars and a few other effects (such as shadows cast by some hair styles) remain as meaningless sources of variation.

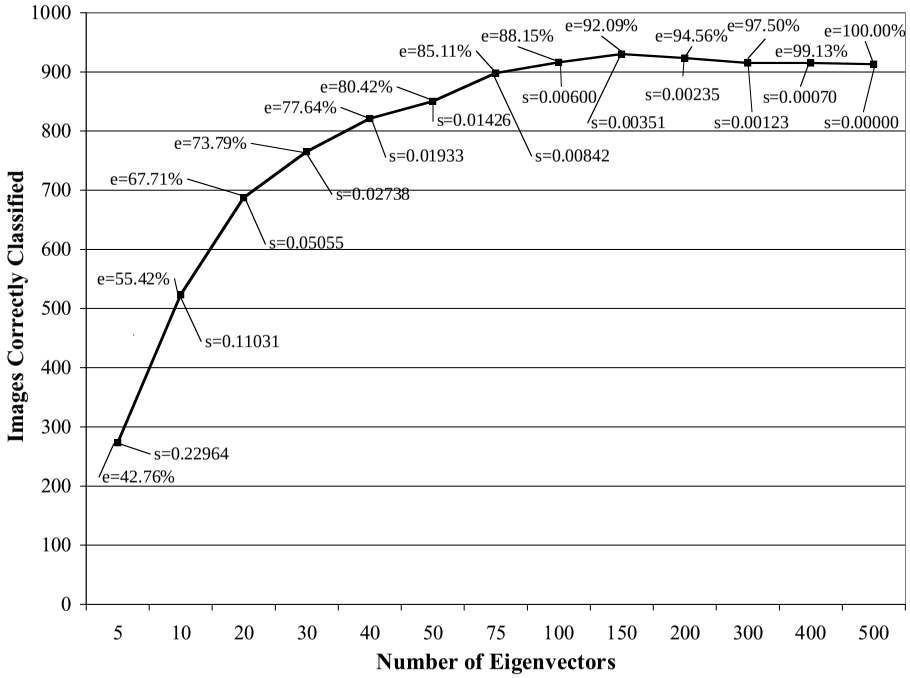


Fig. 3. The Energy and Stretching dimensions on the FERET data.

mance. In the FERET evaluation, the first 200 Eigenvectors were used (with the L1 distance metric) to achieve optimal performance [9]. Removing specific Eigenvectors could in fact improve performance, by removing noise.

4.1 Removing the Last Eigenvectors

The traditional motivation for selecting the Eigenvectors with the largest Eigenvalues is that the Eigenvalues represent the amount of variance along a particular Eigenvector. By selecting the Eigenvectors with the largest Eigenvalues, one selects the dimensions along which the gallery images vary the most. Since the Eigenvectors are ordered high to low by the amount of variance found between images along each Eigenvector, the last Eigenvectors find the smallest amounts of variance. Often the assumption is made that noise is associated with the lower valued Eigenvalues where smaller amounts of variation are found among the images.

There are three variations of deciding how many of the last Eigenvectors to eliminate. The first of these variations removes the last 40% of the Eigenvectors [9]. The second variation uses the minimum number of Eigenvectors to guarantee that energy e is greater than a threshold. A typical threshold is 0.9. If we define e_i as the energy of the ith Eigenvector, it is the ratio of the sum of

all Eigenvalues up to and including i over the sum of all the Eigenvalues:

$$e_i = \frac{\sum_{j=1}^{i} \lambda_j}{\sum_{j=1}^{k} \lambda_j} \tag{2}$$

Kirby defines e_i as the energy dimension [6]. The third variation depends upon the stretching dimension, also defined by Kirby [6]. The stretch s_i for the ith Eigenvector is the ratio of that Eigenvalue over the largest Eigenvalue (λ_1):

$$s_i = \frac{\lambda_i}{\lambda_1} \tag{3}$$

Typically, all Eigenvectors with s_i greater than a threshold are retained. A typical threshold is 0.01. An example of the Energy and Stretching dimensions of the FERET data can be seen in Figure 4.

Specifying the cutoff point beyond which Eigenvectors are removed in terms of a percent of the total is sensitive to the gallery size and insensitive to the actual information present in the Principle Components. Either of the other two measures, energy or stretching, ought to provide a more stable basis for assigning the cutoff point.

4.2 Removing the First Eigenvector

It is possible that the first Eigenvectors encode information that is not relevant to the image identification/classification, such as lighting. Another variation removes the first Eigenvector [9].

4.3 Eigenvalue Ordered by Like-Image Difference

Ideally, two images of the same person should project to the same point in Eigenspace. Any difference between the points is unwanted variation. On the other hand, two images of different subjects should project to points that are as widely separated as possible. To capture this intuition and use it to order Eigenvectors, we define a like-image difference ω_i for each the m Eigenvectors.

To define ω, we will work with pairs of images of the same people projected into Eigenspace. Let X be images in the gallery and Y images of the corresponding people in the probe set ordered such that $x_j \in X$ and $y_j \in Y$ are images of the same person. Define ω_i as follows:

$$\omega_i = \frac{\delta_i}{\lambda_i} \quad \text{where} \quad \delta = \sum_{j=1}^{m} |x_j - y_j| \tag{4}$$

When the difference between images that ought to match is large relative to the variance for that dimension λ_i, then ω_i is large. Conversely, when the difference between images that ought to match is small relative to the variance, ω_i is small.

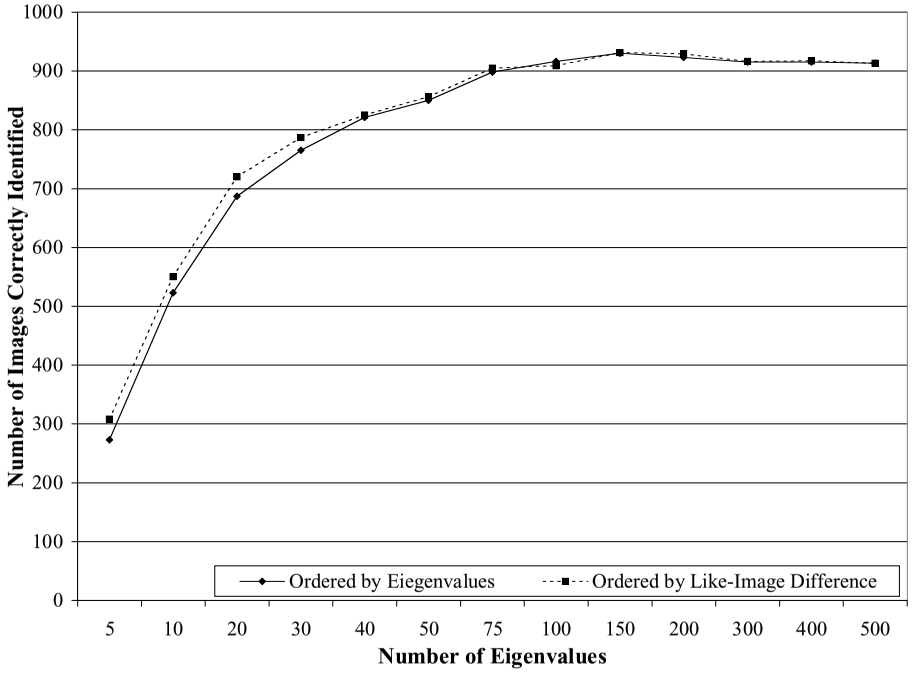


Fig. 4. Performance when ordering by Eigenvalue versus Like-Image Difference.

Since our goal is to select Eigenvectors, i.e. dimensions, that bring like images close to each other, we rank the Eigenvectors in order of ascending ω_i.

For each of the 1,195 probe/gallery matches in the FB probe set of the original FERET dataset, we calculate ω and reorder the Eigenvectors accordingly. The top N Eigenvectors are selected according to this ranking, and the FB probe set was reevaluated using the L1 distance measure. Figure 4 shows the performance scores of the reordered Eigenvalues compared to the performance of the Eigenvectors ordered by Eigenvalue, as performed by Moon & Phillips [9].

Reordering by the like-image difference improves performance for small numbers of Eigenvectors (up to the first 40). This suggests that the like-image difference should be used when selecting a small amount of Eigenvectors. However, there is a problem with the methodology employed in this test. The test (probe) images were used to develop the like-image difference measure ω. A better experiment would reserve a subset of the test imagery for computing ω, and then record how well the classifier performed on the remaining test images. This improved methodology is used in the next section.

4.4 Variation Associated with Different Test/Training Sets

All of the results above use a single Eigenspace developed using 500 gallery images. One question to raise is how sensitive the results are to the choice of

imagery used to construct the Eigenspace and carry out the testing. Here, we take up this question, and pursue whether the differences in performance between algorithm variations are significant relative to variation resulting from computing new Eigenspaces based upon the available training images.

To allow us to run experiments with alternative assignments of imagery to training and test sets, we restructured the FERET dataset so that there was four images of each individual. The resulting dataset consisted of images of 160 individuals. Four pictures are stored of each individual, two of the pictures are taken on the same day, where one picture is of the individual with a neutral facial expression and the other is with a different expression. The other two pictures are taken on a different day with the same characteristics.

In this experiment several factors are varied. The standard recognition system is run with each of the four distance measures. Both large and small Eigenspaces are considered: the small space keeping only the first 20 Eigenvectors and the large discarding the last 40% (keeping 127). Both the large and small space are created using the standard order, where Eigenvectors are sorted by descending Eigenvalue, and our variant, where Eigenvectors are sorted by like-image difference. Finally, 10 trials are run for each variant using a different random assignment of images to the training (gallery) and test (probe) sets.

More precisely, for each of the 160 individuals, two images taken on the same day are randomly selected for training data. Of the 160 individuals, select 20 to compute the like-image difference. For these 20, also include a third image of the individual when computing the like-image difference. For the remaining 140 individuals, select one of the remaining two images at random to construct the test image set. In summary, 320 images are used in each trial, 180 for training and 140 for testing. Only the like-image difference algorithm actually uses the images of the 20 individuals set aside for this purpose, the algorithms that sort by Eigenvalue simply ignores these.

The number of correctly classified images for each of the variants in this experiment are reported in Table 5. Table 5a gives results for the case where the last 40% of the Eigenvectors are discarded. This case is subdivided into two parts. In the first, the Eigenvectors are sorted by decreasing Eigenvalue, the standard case, and the second where the Eigenvectors are sorted by decreasing like-image difference as defined above. For both sorting strategies, the standard PCA nearest neighbor classifier is run using each of the for distance measures: L1, L2, Angle and Mahalanobis. Table 5b gives the analogous results for the case where fewer Eigenvectors are retained: only the first 20.

Let us make three observations based upon the data in Table 5.

Observation 1 There is no apparent improvement when Eigenvectors are sorted by our like-image distance measure. In fact, the average number of correctly classified images drops slightly in 7 out of the 8 cases. However, the net differences in these averages is very small, being less then 1 in half the cases, less than 2 in two cases, and less than 4 in the last two.

Observation 2 Recognition rates are significanlty higher for the larger Eigenspace. This is consistent with prior studies [9]. But the results are compelling and

Table 5. Number of correctly classified images, out of 140, for different algorithm variations. Each row gives results for a different random selection of training and test data. a) Discard last 40% of the Eigenvectors, b) Keep only the first 20 Eigenvectors.

	Standard Order				Sig. Diff. Order			
T	L2	L1	A	M	L2	L1	A	M
1	59	69	67	89	63	68	65	85
2	61	70	62	82	60	68	62	83
3	54	76	71	89	59	76	70	86
4	53	73	67	83	61	71	66	84
5	62	71	58	83	54	70	59	84
6	50	72	61	89	64	67	61	79
7	55	75	66	91	63	72	66	85
8	61	67	61	91	53	69	61	83
9	59	71	60	86	56	72	59	79
10	63	73	66	92	61	73	66	86
	57.7	71.7	63.9	87.5	59.4	70.6	63.5	83.4

(a)

	Standard Order				Sig. Diff. Order			
T	L2	L1	A	M	L2	L1	A	M
1	46	46	52	55	47	45	53	56
2	50	46	50	56	47	44	49	53
3	54	47	58	48	49	43	56	53
4	46	46	46	52	55	47	45	42
5	42	42	51	50	44	44	52	50
6	49	41	55	48	41	42	56	45
7	53	46	44	49	49	46	44	45
8	45	45	50	45	50	44	48	47
9	55	41	49	51	45	41	49	45
10	43	48	48	52	53	49	46	47
	48.3	44.8	50.3	50.6	48.0	44.5	49.8	48.3

(b)

here we can compare the variations associated with the larger Eigenspace with variations arising out of different sets of gallery and probe images. Looking at the standard sorting case, there are four comparisons possible, one for each distance measure. Using a one sided paired sample t test, P_{H0} is less than or equal to $3.0x10^{-3}$, $3.4x10^{-10}$, $2.4x10^{-5}$ and $1.3x10^{-8}$ for the L1, L2, Angle and Mahalanobis distances respectively. In other words, the probability of the null hypothesis is exceedingly low, and it may be rejected with very high confidence.

Observation 3 Results are much better using Mahalanobis distance in the case of the larger number of Eigenvectors, but not when only the first 20 Eigenvectors are used. When using the top 60% of the Eigenvectors sorted by Eigenvalue, the comparison of Mahalanobis to L1, L2 and Angle using a one sided paired sample t test yields P_{H0} is less than or equal to $3,3x10^{-8}$, $5,2x10^{-7}$ and $2.2x10^{-8}$. However, using the first 20 Eigenvectors, the t test comparing Mahalanobis to L1 and Angle yields P_{H0} is less than or equal to 0.13 and 0.17 respectively. In these two latter cases, the null hypothesis cannot be rejected and no statistically meaningful difference in performance can be inferred.

5 Conclusion

Using the original FERET testing protocol, a standard PCA classifier did better when using Mahalanobis distance rather than L1, L2 or Angle. In a new set of experiments where the the training (gallery) and testing (probe) images were selected at random over 10 trials, Mahalanobis was again superior when 60% of the Eigenvectors were used. However, when only the first 20 Eigenvectors were used, L2, Angle and Mahalanobis were equivalent. L1 did slightly worse.

Our efforts to combine distance measures did not result in significant performance improvement. Moreover, the correlation among the L1, L2, Angle and Mahalanobis distance measures, and their shared bias, suggests that although improvements may be possible by combining the L1 measure with other measures, such improvements are likely to be small.

We also compared the standard method for selecting a subset of Eigenvectors to one we developed. While our method seemed a good idea, it did not perform better in our experiments. More recent work suggests this technique is working better than the standard when used in conjunction Fischer Discriminant analysis, but it is premature to say too much about this work.

6 Appendix A:Distance Measures

The L1, L2, Angle and Mahalanobis distances are defined as follows:

L1 City Block Distance

$$d(x,y) = |x-y| = \sum_{i=1}^{k} |x_i - y_i| \tag{5}$$

L2 Euclidean Distance (Squared)

$$d(x,y) = \|x-y\|^2 = \sum_{i=1}^{k} (x_i - y_i)^2 \tag{6}$$

Angle Negative Angle Between Image Vectors

$$d(x,y) = -\frac{x \cdot y}{\|x\|\,\|y\|} = -\frac{\sum_{i=1}^{k} x_i\, y_i}{\sqrt{\sum_{i=1}^{k}(x_i)^2 \sum_{i=1}^{k}(y_i)^2}} \tag{7}$$

Mahalanobis Mahalanobis Distance

$$d(x,y) = -\sum_{i=1}^{k} \frac{1}{\sqrt{\lambda_i}} x_i y_i \tag{8}$$

Where λ_i is the ith Eigenvalue corresponding to the ith Eigenvector. This is a simplification of Moon's definition:

$$d(x,y) = -\sum_{i=1}^{k} z_i x_i y_i \quad \text{where} \quad z_i = \sqrt{\frac{\lambda_i}{\lambda_i + \alpha^2}} \simeq \frac{1}{\sqrt{\lambda_i}} \quad \text{and} \quad \alpha = 0.25 \tag{9}$$

Our original experiments with their definition yielded poor results, hence our adoption of the definition in equation 8.

7 Appendix B: Statistical Tests

7.1 Large Sample Inference Concerning Two Population Proportions

Assume the probe images are drawn from a population of possible images. Let π be the ratio of solvable images over the total number of images in this population. The observed proportion p of problems solved on the sample (probe) images is an estimate of π.

When comparing results using two algorithms A and B, the null hypothesis $H0$ is that $\pi_A = \pi_B$. Following the development in the section "Large-Sample Inferences Concerning A Difference Between Two Population Proportions" in [3], the probability of $H0$ is determined using a standardized variable z:

$$z = \frac{p_A - p_B}{\sqrt{\frac{2p_c(1-p_c)}{n}}} \quad \text{where} \quad p_c = \frac{p_A + p_B}{2} \tag{10}$$

p_A and p_B are the observed proportions of successes in the sample (probe) images for algorithms A and B, and n is the total number of images.

The standardized variable is Gaussian with 0 mean and standard deviation 1. When performing a one sided test, i.e. testing for the case $\pi_A > \pi_B$, the probability of $H0$ is bounded by:

$$P_{H0} \leq \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}} dx \tag{11}$$

7.2 Paired Success/Failure Trials: McNemar's Test

McNemar's test ignores those outcomes where the algorithms do the same thing: either SS or FF. For the remaining outcomes, SF and FS, a Sign Test is used. The null hypothesis $H0$ is that the probability of observing SF is equal to that of observing FS is equal to 0.5. Let a be the number of times SF is observed and b the number of times FS is observed. We are interested in the one sided version of this test, so order our choice of algorithms so $a \geq b$ and assume $H1$ is that algorithm A fails less often then B. Now, the probability of the null hypothesis is bounded by

$$P_{H0} \leq \sum_{i=0}^{b} \frac{n!}{i!\,(n-i)!}\, 0.5^n \quad \text{where} \quad n = a + b \tag{12}$$

8 Acknowledgment

We thank Jonathan Phillips at the National Institute of Standards and Technology for providing us with the results and images from the FERET evaluation. We also thank Jonathan for patiently answering numerous questions. We thank Geof Givens from the Statistics Department at Colorado State University for his insight in pointing us toward McNemar's Test.

References

[1] Peter Belhumeur, J. Hespanha, and David Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):771 – 720, 1997.

[2] L. Breiman. Bagging predictors. Technical Report Technical Report Number 421, Dept. of Statistics, University of California, Berkeley, 1994.

[3] Jay Devore and Roxy Peck. *Statistics: The Exploration and Analysis of Data, Thrid Edition*. Brooks Cole, 1997.

[4] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correction output code. *Journal of Artificial Intelligence Research*, 2:263 – 286, 1995.

[5] IFA. Statistical tests, `http://fonsg3.let.uva.nl:8001/service/statistics.html`). Website, 2000.

[6] M. Kirby. *Dimensionality Reduction and Pattern Analysis: an Empirical Approach*. Wiley (in press), 2000.

[7] E. B. Kong and T. Dietterich. Why error-correcting output coding workds with decision trees. Technical report, Dept. of Computer Science, Oregon State University, Corvalis, 1995.

[8] M. Kirby and L. Sirovich. Application of the Karhunen-Loeve Procedure for the Characterization of Human Faces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 12(1):103 – 107, January 1990.

[9] H. Moon and J. Phillips. Analysis of pca-based face recognition algorithms. In K. Boyer and J. Phillips, editors, *Empirical Evaluation Techniques in Computer Vision*. IEEE Computer Society Press, 1998.

[10] J. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation. In H. Wechslet, J. Phillips, V. Bruse, F. Soulie, and T. Hauhg, editors, *Face Recognition: From Theory to Application*. Springer-Verlag, Berlin, 1998.

[11] J. Phillips, H. Moon, S. Rizvi, and P. Rauss. The feret evaluation methodology for face-recognition algorithms. Technical Report Technical Report Number 6264, NIST, 1999.

[12] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 1988.

[13] Shree K. Nayar, Sameer A. Nene and Hiroshi Murase. Real-Time 100 Object Recognition System. In *Proceedings of ARPA Image Understanding Workshop*. Morgan Kaufmann, 1996. http://www.cs.columbia.edu/CAVE/rt-sensors-systems.html.

[14] L. Sirovich and M. Kirby. A low-dimensional procedure for the characterization of human faces. *The Journal of the Optical Society of America*, 4:519 – 524, 1987.

[15] D. Swets and J. Weng. Using discrimant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

[16] D. Swets and J. Weng. Hierarchical discriminant analysis for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):386–401, 1999.