Feature Selection in Face Recognition: A Sparse Representation Perspective

Allen Y. Yang, *Member, IEEE,* John Wright, *Student Member, IEEE,*
Yi Ma, *Senior Member, IEEE,* and S. Shankar Sastry, *Fellow, IEEE.*

A. Yang and S. Sastry are with the Department of Electrical Engineering and Computer Science, University of California at Berkeley. J. Wright and Y. Ma are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. Corresponding author: Yi Ma, Email: yima@uiuc.edu.

Abstract

In this paper, we examine the role of feature selection in face recognition from the perspective of sparse representation. We cast the recognition problem as finding a sparse representation of the test image features *w.r.t.* the training set. The sparse representation can be accurately and efficiently computed by ℓ^1-minimization. The proposed simple algorithm generalizes conventional face recognition classifiers such as nearest neighbors and nearest subspaces. Using face recognition under varying illumination and expression as an example, we show that if sparsity in the recognition problem is properly harnessed, the choice of features is no longer critical. What is critical, however, is whether the number of features is sufficient and whether the sparse representation is correctly found. We conduct extensive experiments to validate the significance of imposing sparsity using the Extended Yale B database and the AR database. Our thorough evaluation shows that, using conventional features such as Eigenfaces and facial parts, the proposed algorithm achieves much higher recognition accuracy on face images with variation in either illumination or expression. Furthermore, other unconventional features such as severely down-sampled images and randomly projected features perform almost equally well with the increase of feature dimensions. The differences in performance between different features become insignificant as the feature-space dimension is sufficiently large.

Index Terms

Face Recognition, Feature Selection, Eigenface, Laplacianface, Fisherface, Randomface, Sparse Representation, ℓ^1-Minimization, Validation and Outlier Rejection.

I. INTRODUCTION

Human faces are arguably the most extensively studied object in image-based recognition. This is partly due to the remarkable face recognition capability of the human visual system [1], and partly due to numerous important applications for automatic face recognition technologies. In addition, technical issues associated with face recognition are representative of object recognition in general. A central issue in the study of object recognition has been the question of *which features of an object are the most important or informative for recognition*. Due to the special geometric shape and appearance of the face, instead of using fixed filter banks (*e.g.,* downsampling, Fourier, Gabor, wavelets) that are effective for analyzing stationary signals such as textures, the dominant approaches choose to construct facial features adaptively based on the given images, via techniques such as Eigenfaces [2], Fisherfaces [3], Laplacianfaces [4], and

(a) (b)

Fig. 1. (a). Original face image. (b). 120-D representations in terms of four different features (from left to right): Eigenfaces, Laplacianfaces, down-sampled (12×10 pixel) image, and Randomfaces (see Section V for precise description). We will demonstrate that all these features contain almost the same information about the identity of the subject and give similarly good recognition performance.

their variants [5], [6] (see Figure 1 for examples), as well as facial parts or components [7] (see Figure 10 for examples). The features extracted using such filters or masks are thought to be more relevant to face recognition, allowing reasonable recognition performance with simple, scalable classifiers such as *nearest neighbor* (NN) [8] and *nearest subspace* (NS) [9] (*i.e.,* minimum distance to the subspace spanned by images of each subject).

However, with so many proposed features but so little consensus about which feature are better or worse, practitioners lack guidelines to decide which features to use. In the face recognition community, there has been enormous effort and emphasis on finding the "optimal" features. This quest may have obscured other important factors that can help clarify the role of feature selection in the overall recognition process. For instance, the performance variation of different features may be due in part to the choice of classifiers (*e.g.,* NN or NS), or in part to the choice in the range of feature dimension. The performance of conventional features in conjunction with these classifiers has been generally unsatisfactory, far below human face recognition capability. Frustration with this state of affairs has led researchers to resort to nonlinear features and kernel methods [10], [11]. Nevertheless, it is still possible that those simple conventional features already contain sufficient information for accurate recognition and we have simply not employed pertinent tools to harness such information.

In this paper, we cast object recognition as finding a *sparse representation* of a given test image *w.r.t.* the training images of multiple objects:

> *We represent the test image as a linear combination of the fewest training images possible, out of the entire training set.*

If the test image is indeed an image of one of the subjects in the training database, this linear combination will only involve training images of that subject. Such a representation is naturally *sparse*: only a small fraction of the coefficients are nonzero.[1] Such a sparse representation can be effectively computed via ℓ^1-minimization [14]. We propose an extremely simple classification algorithm for face recognition based on the representation computed. Experimental results convincingly demonstrate the key role of sparsity in recognition: the algorithm achieves high recognition rates on both the Extended Yale B database and the AR database, significantly boosting the performance of popular face features such as Eigenfaces and facial parts.

a) Global versus local approach: Notice, however, that we are proposing to compute the sparsest representation in terms of all the training images. Computing such a *global* representation has several advantages over *local* (one image or subject at a time) methods such as NN and NS. The first is in *validation*, or outlier rejection: if the most compact representation in terms of the whole training set involves images from several classes, the test image is not likely to be a typical sample from any one of those classes. For face recognition, this is equivalent to asking "Does the most compact representation treat this image as a generic face or as the face of a single individual in the training dataset?" (see Section IV). The second advantage is in *identification*: sparse representation is similarly discriminative for identification of a (validated) test image within the subjects in the dataset. It selects out of all the training images a few that most compactly represent the test image and hence naturally avoids the problem with under-fitting or over-fitting (see Section III for more explanation). Through analysis and experiments, we will justify the superiority of this global scheme over local methods such as NN and NS, in both validation and identification.

b) Efficient computational tools: Casting recognition as a (globally) sparse representation problem also allows us to leverage powerful tools from the emerging mathematical theory of compressed sensing [15]. For example, while finding the sparsest representation is an NP-hard problem that does not even allow efficient approximation [16], for our problem of interest it

[1]The sparsity described here differs from the sparse features proposed in [12] for object detection or in [13] for image representation. We are concerned with sparsity of the representation (coefficients), whereas those works deal with spatial locality of the basis or features.

can be computed efficiently and exactly, by ℓ^1-minimization.[2] Although computing a global representation in terms of all of the training images may seem computationally extravagant, with an appropriate choice of optimization algorithm the complexity becomes linear in the number of training samples [17], just as with NN and NS. Furthermore, compressed sensing offers a new perspective on feature selection: it suggests the number of features is much more important than the details of how they are constructed. As long as the number of features is large enough, even randomly chosen features are sufficient to recover the sparse representation [14], [15].

c) Feature selection in the new context: Thus, we are compelled to re-examine feature selection in face recognition from this new perspective and try to answer the following question:

To what extent does the selection of features still matter, if the sparsity inherent in the recognition problem is properly harnessed?

Our experiments will show that the performances with conventional features (*e.g.,* Eigenfaces, Laplacianfaces) converge as the number of features used increases, as predicted by the theory of sparse representation and compressed sensing [15]. It should therefore come as no surprise that similarly high recognition rates can be achieved using the same number of down-sampled pixels or even completely random features, which we call *Randomfaces* (see Figure 1 for an example). For example, consider 12×10 down-sampled images, as shown in Figure 1. Our experiments will show that using such severely down-sampled images as features, our algorithm can still achieve a recognition rate as high as 92.1% over the Extended Yale B database (see Figure 3). Such performance arguably surpasses humans' ability to recognize down-sampled images – humans typically require about 16×16 pixels even for familiar faces [1].

d) What we do not do: In this paper, we assume the face images have been cropped and normalized. We only consider face recognition of frontal views, and we do not consider pose variations. So our conclusions on feature selection only apply to the frontal case. Feature selection for face recognition with pose variation, or 3D-model based face recognition, or face detection/alignment can be rather different problems. It remains an open problem whether the proposed new framework will have any implications on those problems too.

[2]Whereas many of the results in compressed sensing pertain to random matrix ensembles, we work with very specific matrices (determined by the training dataset). We will verify the validity of these results in this deterministic setting.

e) Relation to the companion paper: In this paper, we do not deal with occluded or corrupted test face images but that is the topic of a companion paper [18]. Although both papers deal with sparse solutions, the breakdown point of the proposed method is much more likely to be reached in the case of occluded images. Thus, a more careful characterization of the breakdown point is given in the companion paper.

II. Problem Formulation and Solution

A. Recognition as a Linear Sparse Representation

In this paper, we consider face recognition with frontal face images from k individuals that have been properly cropped and normalized. Each image is of the size $w \times h$ and can be viewed as a point in the space $\mathbb{R}^m$ with $m = w \times h$. The images of the same face under varying illumination or expressions have been shown to span an (approximately) low-dimensional linear subspace in $\mathbb{R}^m$, called a *face subspace* [3], [19]. Although the proposed framework and algorithm can apply to the situation where the training images of each subject have a multi-modal or nonlinear distribution (Section III discusses this issue in detail), for ease of presentation, let us first assume that the training images of a single subject do span a subspace. This is the only prior knowledge about the training images we will be using in our solution.[3]

Let us stack the n_i images associated with subject i as vectors $\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i} \in \mathbb{R}^m$ and assume these vectors are sufficient to span the i-th face subspace: Any new test image of the same subject, stacked as a vector $\boldsymbol{y} \in \mathbb{R}^m$, can be represented as a linear superposition of the training examples associated with Subject i:

$$\boldsymbol{y} = \alpha_{i,1}\boldsymbol{v}_{i,1} + \alpha_{i,2}\boldsymbol{v}_{i,2} + \cdots + \alpha_{i,n_i}\boldsymbol{v}_{i,n_i}, \tag{1}$$

for some scalars $\alpha_{i,j} \in \mathbb{R}, j = 1, \cdots, n_i$.

Collect all the $n \doteq n_1 + \cdots + n_k$ training images as column vectors of one matrix:

$$A \doteq [\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2}, \ldots, \boldsymbol{v}_{1,n_1}, \boldsymbol{v}_{2,1}, \ldots, \boldsymbol{v}_{k,n_k}] \in \mathbb{R}^{m \times n}. \tag{2}$$

[3]We actually do not need to know whether the linear structure is due to varying illumination or expression. Thus, we do not need to use domain-specific knowledge such as an illumination model [20] to eliminate the variability in the training and testing images.

Then ideally the test image $\boldsymbol{y}$ of subject i can be represented in terms of all of the images in the training set as

$$\boldsymbol{y} \;=\; A\boldsymbol{x}_0 \quad \in \mathbb{R}^m, \tag{3}$$

where $\boldsymbol{x}_0 = [0, \cdots, 0, \alpha_{i,1}, \alpha_{i,2}, \ldots, \alpha_{i,n_i}, 0, \ldots, 0]^T \in \mathbb{R}^n$ is a coefficient vector whose entries are mostly zero except those associated with the i-th subject. We therefore exploit the following simple observation: A valid test image can be sufficiently represented using only the training images of the same subject. This representation is naturally sparse if the number of subjects k is reasonably large.

As the entries of the vector $\boldsymbol{x}_0$ encode the identity of the test image $\boldsymbol{y}$, it is tempting to attempt to obtain it by solving the linear system (3). This linear system of equations is *global*, involving the entire training set. The reader may wonder if a global approach is really necessary, given that there is a sparse representation in terms of just the training images of one subject. We will see that the global approach has advantages over local methods such as NN and NS in discriminating between subjects in the training set (Section III) and in rejecting invalid test images (*i.e.,* outliers) (Section IV). These advantages come without an increase in the order of growth of the computation: as we will see, the complexity remains linear in the number of training samples.

If the system (3) is under-determined ($m < n$ for $A \in \mathbb{R}^{m \times n}$), its solution is not unique. Traditionally a solution is chosen with minimum ℓ^2-norm:

$$(P_2) \qquad \min \|\boldsymbol{x}\|_2 \quad \text{subject to} \quad \boldsymbol{y} = A\boldsymbol{x}. \tag{4}$$

Similarly, if the system is over-determined ($m > n$), one often seeks the *least-squares* solution by minimizing $\|\boldsymbol{y} - A\boldsymbol{x}\|_2$.[4]

However, these approaches fail to address two important properties of the recognition problem:

1) The data are *very high-dimensional*. For instance, for a 640×480 grayscale image, the dimension m is on the order of 10^5, while we normally have only a few samples per subject. Such small sample size only exacerbates "the curse of dimensionality" that

[4]A solution we would also discourage in the companion paper [18].

plagues high-dimensional statistics [21], and in particular face recognition [10].[5] Aside from the computational cost of solving such large systems of equations, the least-squares (or minimum ℓ^2-norm) solution can exhibit severe bias if the system is not properly regularized [22]. This degeneracy phenomenon noticeably hinders recognition methods such as *nearest neighbor* and *nearest subspace* (see Table IV in Section VI).

2) The desired solution is *sparse*. The ratio of the nonzero entries in $\boldsymbol{x}_0$ is only $\frac{n_i}{n} \approx \frac{1}{k}$: For instance, if $k = 20$, only 5% of the entries of $\boldsymbol{x}_0$ should be nonzero. The more sparse the recovered $\boldsymbol{x}$ is, the easier it will be to accurately determine the identity of the test image $\boldsymbol{y}$. Unfortunately, the minimum ℓ^2-norm solution of the equation $\boldsymbol{y} = A\boldsymbol{x}$ is generally non-sparse, and can be very far from the true sparse solution in (3) when the system is under-determined or there is a large error in $\boldsymbol{y}$ [23]–[25].

B. Sparse Solution in a Reduced Dimension

To tackle the above difficulties, people typically seek methods that 1) reduce the data dimension m to $d \ll m$ and 2) explicitly compute the sparse representation of $\boldsymbol{y}$ in the lower-dimensional space. We will see that these two goals are complementary: Appropriately enforcing sparsity renders the outcome less dependent on the details of dimension reduction.

In the computer vision literature, numerous dimension reduction methods have been investigated for projecting high-dimensional face images to low-dimensional feature spaces. One class of methods extracts holistic face features, such as Eigenfaces [2], Fisherfaces [3], and Laplacianfaces [4]. Another class of methods tries to extract significant partial facial features (*e.g.,* eye corners) [1], [26]. For such face features, the projection from the image space to the feature space can be represented as a matrix $R \in \mathbb{R}^{d \times m}$ with $d \ll m$. Applying R to both sides of equation (3) yields:

$$\tilde{\boldsymbol{y}} \doteq R\boldsymbol{y} = RA\boldsymbol{x}_0 \quad \in \mathbb{R}^d. \tag{5}$$

After projection, the dimension d of the feature space usually becomes smaller than n. Hence, the system of equations (5) is under-determined, and the solution $\boldsymbol{x}$ is not unique. Nevertheless,

[5]"The curse of dimensionality" as a phase has been used in different areas with different meanings. We use it here to emphasize two major barriers associated with face recognition: The imagery data are high-dimensional, and the global sparse representation is also high-dimensional.

the desired $\boldsymbol{x}_0$ should still be sparse. Under very mild conditions on $\tilde{A} = RA$, the sparsest solution to the system of equations is indeed unique [23]. In other words, the desired $\boldsymbol{x}_0$ is the unique solution to the following optimization problem:

$$(P_0) \qquad \min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad \tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}, \tag{6}$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which simply counts the number of nonzero entries in a vector. Solving (P_0) is NP-hard and even difficult to approximate by polynomial-time algorithms [27]: In the general case, no known procedure for finding the sparsest solution is significantly more efficient than exhausting all subsets of the entries for $\boldsymbol{x}$.

C. Sparse Solution via ℓ^1-Minimization

Recent development in the emerging *compressed sensing* theory [23]–[25] reveals that if the solution $\boldsymbol{x}_0$ sought is *sparse enough*, the combinatorial problem (P_0) is equivalent to the following ℓ^1-minimization problem:

$$(P_1) \qquad \min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}. \tag{7}$$

This problem can be solved in polynomial time by standard linear programming or quadratic programming methods [28]. Even more efficient methods are available when the solution is known to be very sparse. For example, homotopy algorithms recover solutions with k nonzeros in $O(k^3 + n)$ time, linear in the size of the training set [17].

Figure 2 gives a geometric interpretation (essentially due to [29]) of why minimizing the ℓ^1-norm recovers sparse solutions. Let C_α denote the ℓ^1-ball (or cross-polytope) of radius α:

$$C_\alpha \doteq \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 \leq \alpha\} \subset \mathbb{R}^n. \tag{8}$$

In Figure 2, the unit ℓ^1-ball C_1 is mapped to the polytope $P \doteq \tilde{A} \cdot C_1 \subset \mathbb{R}^d$ consisting of all $\tilde{\boldsymbol{y}}$ that satisfy $\tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}$ for some $\boldsymbol{x}$ whose ℓ^1-norm is ≤ 1.

The geometric relationship between C_α and the polytope $\tilde{A} \cdot C_\alpha$ is invariant to scaling. That is, if we scale C_α, its image under $\tilde{A}$ is also scaled by the same amount. Geometrically, finding the minimum ℓ^1-norm solution $\boldsymbol{x}_1$ to (P_1) is equivalent to expanding the ℓ^1-ball C_α until the polytope $\tilde{A} \cdot C_\alpha$ first touches $\tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}_0$. The value of α at which this occurs is exactly $\|\boldsymbol{x}_1\|_1$.

Now suppose that $\tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}_0$ for some sparse $\boldsymbol{x}_0$. We wish to know when solving (P_1) correctly recovers $\boldsymbol{x}_0$. This question is easily resolved from the geometry of Figure 2: Since $\boldsymbol{x}_1$ is found

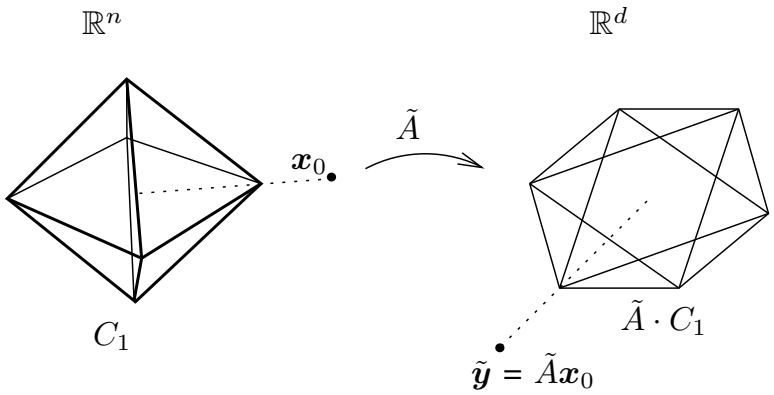

Fig. 2. Geometry of the sparse solution via ℓ^1-minimization. The ℓ^1-minimization determines in which facet of the polytope $\tilde{A} \cdot C^1$ the point $\tilde{\boldsymbol{y}}$ lies, and then $\tilde{\boldsymbol{y}}$ is represented as a linear combination of the vertices of that facet, with the coefficients $\boldsymbol{x}_0$.

by expanding both C_α and $P = \tilde{A} \cdot C_\alpha$ until a point of P touches $\tilde{\boldsymbol{y}}$, the ℓ^1-minimizer $\boldsymbol{x}_1$ must generate a point $\tilde{A}\boldsymbol{x}_1$ on the boundary of P.

Thus $\boldsymbol{x}_1 = \boldsymbol{x}_0$ if and only if the point $\tilde{A}(\boldsymbol{x}_0/\|\boldsymbol{x}_0\|_1)$ lies on the boundary of P. For the example shown in Figure 2, it is easy to see that ℓ^1-minimization recovers all $\boldsymbol{x}_0$ with only one nonzero entry. This equivalence holds because all of the vertices of C_1 map to points on the boundary of P.

If $\tilde{A}$ maps all k-dimensional faces of C_1 to faces of P, the polytope P is referred to as *(centrally) k-neighborly* [29]. From the above, we see that (P_1) recovers all $\boldsymbol{x}_0$ with $\leq k+1$ nonzeros iff P is k-neighborly. This condition is surprisingly common: the results of [15] show that even random matrices (*e.g.,* uniform, Gaussian, and partial Fourier) are highly neighborly and therefore admit sparse solution by ℓ^1-minimization.

Unfortunately, there is no known algorithm for efficiently verifying the neighborliness of a given polytope P. The best known algorithm is combinatorial and therefore only practical when the dimension n is moderate [30]. When n is large, it is known that with overwhelming probability, the neighborliness of a randomly chosen polytope P is loosely bounded between:

$$c \cdot n < k < n/3 \tag{9}$$

for some small constant $c > 0$ (see [23], [29]). In other words, in general, as long as the number of nonzero entries of $\boldsymbol{x}_0$ is a small fraction of the dimension n, ℓ^1-minimization will recover $\boldsymbol{x}_0$. This is precisely the situation in face recognition: the support of the desired solution $\boldsymbol{x}_0$

is a fixed fraction of the number of training images; the more subjects there are, the smaller the fraction. In Section VI, our experimental results will verify the ability of ℓ^1-minimization to recover sparse representations for face recognition, suggesting that the data-dependent features popular in face recognition (*e.g.,* Eigenfaces) may indeed give highly neighborly polytopes P.

Since real images are noisy, it may not be possible to express the (features of) the test image exactly as a sparse superposition of (features of) the training images. To model noise and error in the data, one can consider a stable version of (5) that includes a noise term with bounded energy $\|\boldsymbol{z}\|_2 < \epsilon$:

$$\tilde{\boldsymbol{y}} = \tilde{A}\boldsymbol{x}_0 + \boldsymbol{z} \in \mathbb{R}^d. \tag{10}$$

It has been shown that in this case a sparse near solution can be found via the following program:

Theorem 1 (Minimal ℓ^1-Norm Near Solution [14]): Consider an under-determined system (10) with a large dimension d and $\|\boldsymbol{z}\|_2 < \epsilon$. Let $\boldsymbol{x}_1$ denote the solution to the following problem:

$$(P_1') \quad \min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|\tilde{\boldsymbol{y}} - \tilde{A}\boldsymbol{x}\|_2 \leq \epsilon. \tag{11}$$

Then with overwhelming probability, there exist $\rho > 0$ and $\zeta > 0$ such that for all sparse $\boldsymbol{x}_0$ with $\|\boldsymbol{x}_0\|_0 \leq \rho d$,

$$\|\boldsymbol{x}_1 - \boldsymbol{x}_0\|_2 \leq \zeta\epsilon. \tag{12}$$

The minimization problem (11) can be efficiently solved via convex optimization [28] (see Section VI for our algorithm of choice).

D. Classification from Sparse Coefficients

Ideally, the nonzero entries in the estimate $\boldsymbol{x}$ will all be associated with the columns in $\tilde{A}$ from a single subject, and we can easily assign the test image $\boldsymbol{y}$ to that subject. However, due to noise, the nonzero entries may be associated with multiple subjects (see Figure 3). Many classifiers can resolve this problem. For instance, we can simply assign $\boldsymbol{y}$ to the subject with the single largest entry of $\boldsymbol{x}$. However, such heuristics do not harness the subspace structure associated with face images. To better harness this structure, we instead classify $\boldsymbol{y}$ based on how well the coefficients associated with all training images of each subject reproduce $\boldsymbol{y}$.

For each subject i, define its characteristic function $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ which selects the coefficients associated with the i-th subject. For $\boldsymbol{x} \in \mathbb{R}^n$, $\delta_i(\boldsymbol{x}) \in \mathbb{R}^n$ is a new vector whose only nonzero

entries are the entries in $\boldsymbol{x}$ that are associated with subject i, and whose entries associated with all other subjects are zero. We then set

$$\text{identity}(\boldsymbol{y}) = \arg\min_i \; r_i(\boldsymbol{y}), \quad \text{where} \quad r_i(\boldsymbol{y}) \doteq \|\tilde{\boldsymbol{y}} - \tilde{A}\,\delta_i(\boldsymbol{x})\|_2. \tag{13}$$

That is, we assign $\boldsymbol{y}$ to the subject whose associated coefficients, $\delta_i(\boldsymbol{x})$, give the best approximation to $\boldsymbol{y}$. Algorithm 1 below summarizes the complete recognition procedure.

Algorithm 1 (Recognition via Sparse Representation)

1: **Input:** a matrix of training images $A \in \mathbb{R}^{m\times n}$ for k subjects, a linear feature transform $R \in \mathbb{R}^{d\times m}$, a test image $\boldsymbol{y} \in \mathbb{R}^m$, and an error tolerance ϵ.

2: Compute features $\tilde{\boldsymbol{y}} = R\boldsymbol{y}$ and $\tilde{A} = RA$, and normalize $\tilde{\boldsymbol{y}}$ and columns of $\tilde{A}$ to unit length.

3: Solve the convex optimization problem (P_1'):

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|\tilde{\boldsymbol{y}} - \tilde{A}\boldsymbol{x}\|_2 \le \epsilon.$$

4: Compute the residuals $r_i(\boldsymbol{y}) = \|\tilde{\boldsymbol{y}} - \tilde{A}\,\delta_i(\boldsymbol{x})\|_2$ for $i = 1, \ldots, k$.

5: **Output:** $\text{identity}(\boldsymbol{y}) = \arg\min_i r_i(\boldsymbol{y})$.

Example 1 (ℓ^1-Minimization versus ℓ^2-Minimization): To illustrate how Algorithm 1 works, we randomly select half of the $2,414$ images in the Extended Yale B database as the training set, and the rest for testing. In this example, we choose R to simply be the down-sampling filter that sub-samples the images from 192×168 to size 12×10. The pixel values of the down-sampled image are used as features, and hence the feature space dimension is $d = 120$. Figure 3 top illustrates the sparse coefficients recovered by Algorithm 1 for a test image from Subject 1. The figure also shows the features and original images that correspond to the two largest coefficients. As we see, the two largest coefficients are both associated with training samples from Subject 1. Figure 3 bottom plots the residuals *w.r.t.* the 38 projected coefficients $\delta(\boldsymbol{x}_i)$, $i = 1, 2, \ldots, 38$. With 12×10 down-sampled images as features, Algorithm 1 achieves an overall recognition rate of 92.1% across the Extended Yale B database (see Section VI). For comparison, Figure 4 top shows the coefficients of the same image features given by the conventional ℓ^2-minimization (4), and Figure 4 bottom shows the corresponding residuals *w.r.t.* the 38 subjects. The coefficients are much less sparse than those given by ℓ^1-minimization (in

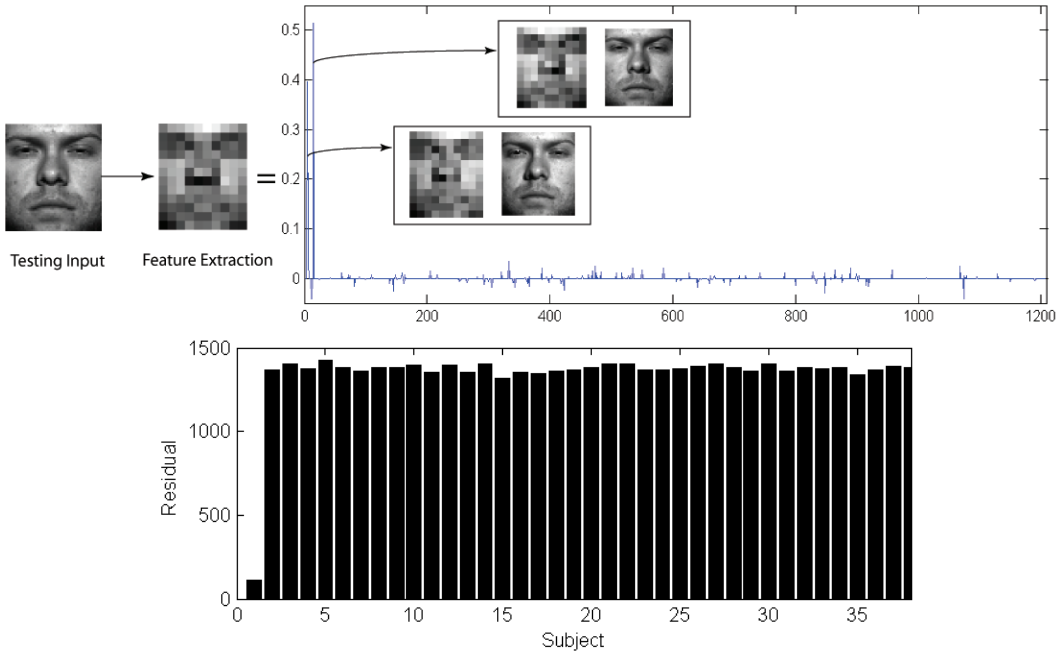

Fig. 3. Top: Recognition with 12×10 down-sampled images as features. The test image $\boldsymbol{y}$ belongs to Subject 1. The values of the sparse coefficients recovered from Algorithm 1 are plotted on the right together with the two training examples that correspond to the two largest sparse coefficients. Bottom: The residuals $r_i(\boldsymbol{y})$ of a test image of Subject 1 *w.r.t.* the projected sparse coefficients $\delta_i(\boldsymbol{x})$ by ℓ^1-minimization. The ratio between the magnitudes of the two smallest residuals is about 1:8.6.

Figure 3), and the dominant coefficients are not associated with Subject 1 either. As a result, the smallest residual in Figure 4 is not at Subject 1.

III. Relationships to Nearest Neighbor and Nearest Subspace

The above example illustrates ℓ^1-minimization's superior ability to recover the desired sparse representation, compared to ℓ^2-minimization. One may notice that the use of *all* the training images of *all* subjects to represent the test image goes against the conventional classification methods popular in face recognition literature and existing systems. These methods typically suggest using residuals computed from "one training image at a time" or "one subject at a time" to classify the test image. The representative methods include:

1) The *nearest neighbor* (NN) classifier: Assign the test image $\boldsymbol{y}$ to subject i if the smallest distance from $\boldsymbol{y}$ to the nearest training image of subject i

$$r_i(\boldsymbol{y}) = \min_{j=1,\ldots,n_i} \|\tilde{\boldsymbol{y}} - \tilde{\boldsymbol{v}}_{i,j}\|_2 \tag{14}$$

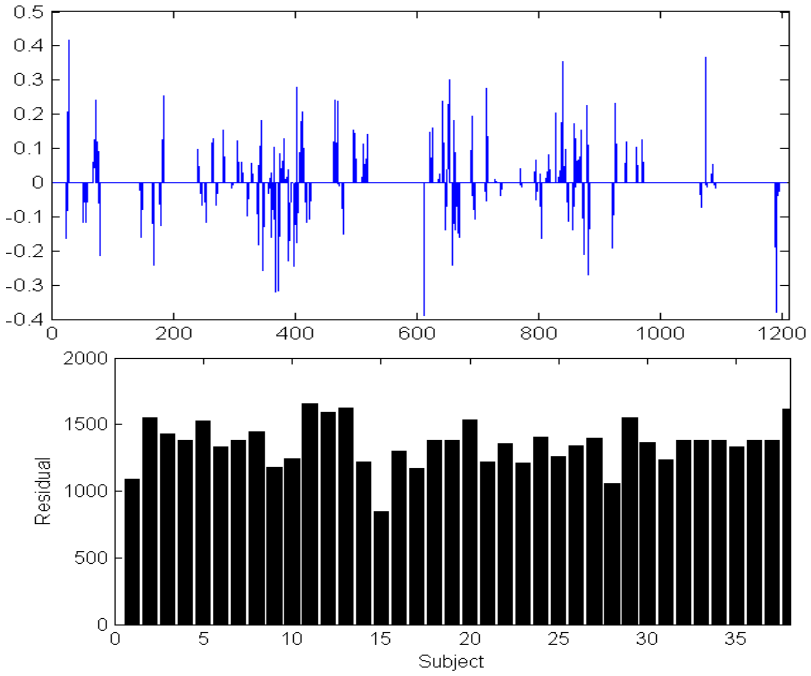

Fig. 4. Top: Coefficients from ℓ^2-minimization, using the same test image as Figure 3. The recovered solution is not sparse and hence less informative for recognition (large coefficients do not correspond to training images of this test subject). Bottom: The residuals of the test image from Subject 1 *w.r.t.* the projection $\delta_i(\boldsymbol{x})$ of the coefficients obtained by ℓ^2-minimization. The ratio between the magnitudes of the two smallest residuals is about 1:1.3. The smallest residual is not associated with Subject 1.

is the smallest among all subjects.[6]

2) The *nearest subspace* (NS) classifier (*e.g.,* [31]): Assign the test image $\boldsymbol{y}$ to subject i if the distance from $\boldsymbol{y}$ to the subspace spanned by all images $A_i = [\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,n_i}]$ of the subject i:

$$r_i(\boldsymbol{y}) = \min_{\boldsymbol{x}_i \in \mathbb{R}^{n_i}} \|\tilde{\boldsymbol{y}} - \tilde{A}_i \boldsymbol{x}_i\|_2 \tag{15}$$

is the smallest among all subjects.

Clearly, NN seeks the best representation in terms of just a single training image,[7] while NS seeks the best representation in terms of all the training images of each subject.

[6]Another popular distance metric for the residual is the ℓ^1-norm distance $\|\cdot\|_1$. This is not to be confused with the ℓ^1-minimization in this paper.

[7]Alternatively, a similar classifier *KNN* considers K nearest neighbors.

A. Relationship to Nearest Neighbor

Let us first assume that a test image $\boldsymbol{y}$ can be well-represented in terms of one training image, say $\boldsymbol{v}_i$ (one of the columns of A):

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{v}}_i + \boldsymbol{z}_i \tag{16}$$

where $\|\boldsymbol{z}_i\|_2 \leq \epsilon$ for some small $\epsilon > 0$. Then according to Theorem 1, the recovered sparse solution $\boldsymbol{x}$ to (11) satisfies

$$\|\boldsymbol{x} - \boldsymbol{w}_i\|_2 \leq \zeta\epsilon$$

where $\boldsymbol{w}_i \in \mathbb{R}^n$ is the vector whose i-th entry is 1 and others are all zero, and ζ is a constant that depends on $\tilde{A}$. Thus, in this case, the ℓ^1-minimization based classifier will give the same identification for the test image as NN.

On the other hand, test images may have large variability due to different lighting conditions or facial expressions, and the training sets generally do not densely cover the space of all possible images (we will see in the next section that this is the case with the AR database). In this case, it is unlikely that any single training image will be very close to the test image, and nearest-neighbor classification may perform poorly.

Example 2: Figure 5 shows the ℓ^2-distances between the down-sampled face image from Subject 1 in Example 1 and each of the training images. Although the smallest distance is correctly associated with Subject 1, the variation of the distances for other subjects is quite large. As we will see in Section VI, this inevitably leads to inferior recognition performance when using NN (only 71.6% in this case, comparing to 92.1% of Algorithm 1).[8]

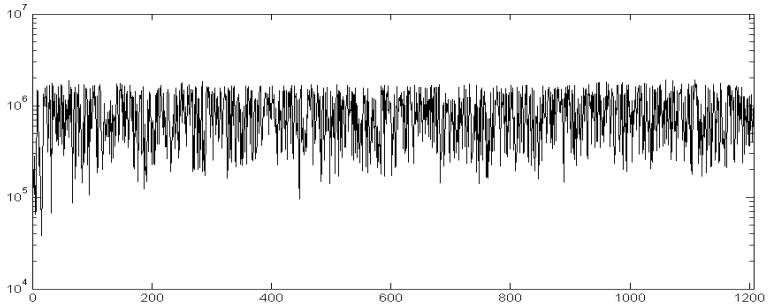

Fig. 5. The ℓ^2 distances (logarithmic scale) between the test image and the training images in Example 1.

[8]Other commonly used distance metrics in NN such as ℓ^1-distance give results similar to Figure 5.

B. Relationship to Nearest Subspace

Let us now assume that a test image $\boldsymbol{y}$ can be well-represented as a linear combination of the training images A_i of subject i:

$$\tilde{\boldsymbol{y}} = \tilde{A}_i \boldsymbol{x}_i + \boldsymbol{z}_i \tag{17}$$

where $\|\boldsymbol{z}_i\|_2 \leq \epsilon$ for some small $\epsilon > 0$. Then again according to Theorem 1, the recovered sparse solution $\boldsymbol{x}$ to (11) satisfies

$$\|\boldsymbol{x} - \boldsymbol{w}_i\|_2 \leq \zeta\epsilon$$

where $\boldsymbol{w}_i \in \mathbb{R}^n$ is a vector of the form $[0, \ldots, 0, \boldsymbol{x}_i^T, 0, \ldots, 0]^T$ (if $\boldsymbol{w}_i$ is the unique solution that satisfies $\|\tilde{\boldsymbol{y}} - \tilde{A}\boldsymbol{w}_i\|_2 < \epsilon$). That is,

$$\delta_i(\boldsymbol{x}) \approx \boldsymbol{w}_i \;\text{ and }\; \|\delta_j(\boldsymbol{x})\| < \zeta\epsilon \quad \text{for all } j \neq i. \tag{18}$$

We have

$$\|\tilde{\boldsymbol{y}} - \tilde{A}\delta_i(\boldsymbol{x})\|_2 \approx \|\boldsymbol{z}_i\|_2 \leq \epsilon, \quad \|\tilde{\boldsymbol{y}} - \tilde{A}\delta_j(\boldsymbol{x})\|_2 \approx \|\tilde{\boldsymbol{y}}\|_2 \gg \epsilon \quad \text{for all } j \neq i. \tag{19}$$

Thus, in this case, the ℓ^1-minimization based classifier will give the same identification for the test image as NS. Notice that for $j \neq i$, $\delta_j(\boldsymbol{x})$ is rather different from $\boldsymbol{x}_j$ computed from $\min_{\boldsymbol{x}_j} \|\tilde{\boldsymbol{y}} - A_j\boldsymbol{x}_j\|_2$. The norm of $\delta_j(\boldsymbol{x})$ is bounded by the approximation error (18) when $\boldsymbol{y}$ is represented just within class j, whereas the norm of $\boldsymbol{x}_j$ can be very large as face images of different subjects are highly correlated. Further notice that each of the $\boldsymbol{x}_j$ is *an optimal representation* (in the 2-norm) of $\boldsymbol{y}$ in terms of some (different) subset of the training data, whereas *only one* of the $\{\delta_j(\boldsymbol{x})\}_{j=1}^k$ computed via ℓ^1-minimization is optimal in this sense; the rest have very small norm. In this sense, ℓ^1-minimization is *more discriminative* than NS, as is the set of associated residuals $\{\|\tilde{\boldsymbol{y}} - \tilde{A}\delta_j(\boldsymbol{x})\|_2\}_{j=1}^k$.

Example 3: Figure 6 shows the residuals of the down-sampled features of the test image in Example 1 *w.r.t.* the subspaces spanned by the 38 subjects. Although the minimum residual is correctly associated with Subject 1, the difference from the residuals of the other 37 subjects is not as dramatic as that obtained from Algorithm 1. Compared to the ratio 1:8.6 between the two smallest residuals in Figure 3, the ratio between the two smallest residuals in Figure 6 is only 1:3. In other words, the solution from Algorithm 1 is more discriminative than that from NS. As we will see Section VI, for the 12×10 down-sampled images, the recognition rate of NS is lower than that of Algorithm 1 (91.1% versus 92.1%).

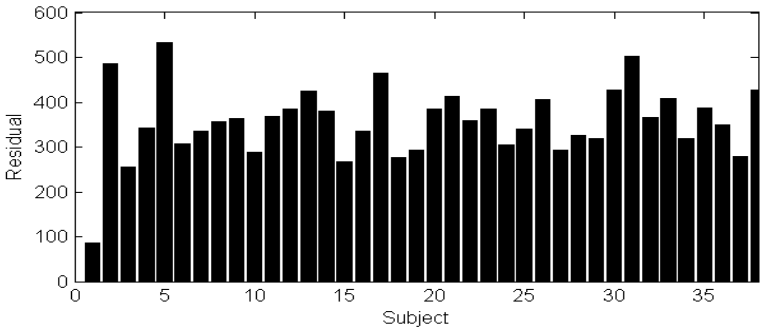

Fig. 6. The residuals of the test image in Example 1 *w.r.t.* the 38 face subspaces. The ratio between the magnitudes of the two smallest residuals is about 1:3.

Be aware that the subspace for each subject is only an approximation to the true distribution of the face images. In reality, due to expression variations, specularity, or alignment error, the actual distribution of face images could be nonlinear or multi-modal. Using only the distance to the entire subspace ignores information about the distribution of the samples within the subspace, which could be more important for classification. Even if the test image is generated from a simple statistical model: $\tilde{\boldsymbol{y}} = \tilde{A}_i \boldsymbol{x}_i + \boldsymbol{z}_i$ with $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ independent Gaussians, any sufficient statistic (for the optimal classifier) depends on both $\|\boldsymbol{x}_i\|_2$ and $\|\boldsymbol{z}_i\|_2$, not just the residual $\|\boldsymbol{z}_i\|_2$. While the ℓ^1-based classifier is also suboptimal under this model, it does implicitly use the information in $\boldsymbol{x}_i$ as it penalizes $\boldsymbol{x}_i$ that has a large norm – the ℓ^1-minimization based classifier favors small $\|\boldsymbol{z}_i\|_2$ as well as small $\|\boldsymbol{x}_i\|_1$ in representing the test image with the training data.

Furthermore, using all the training images in each class may over-fit the test image. In the case when the solution $\boldsymbol{x}_i$ to

$$\tilde{\boldsymbol{y}} = \tilde{A}_i \boldsymbol{x}_i + \boldsymbol{z}_i \quad \text{subject to} \quad \|\boldsymbol{z}_i\|_2 < \epsilon$$

is not unique, the ℓ^1-minimization (7) will find the sparsest $\boldsymbol{x}_i^s$ instead of the least-squares solution $\boldsymbol{x}_i^2 = (\tilde{A}_i^T \tilde{A}_i)^\dagger \tilde{\boldsymbol{y}}$. That is, the ℓ^1-minimization will use the smallest number of sample images necessary in each class to represent the test image, subject to a small error. To see why such an solution $\boldsymbol{x}_i^s$ respects better the actual distribution of the training samples (inside the subspace spanned by all samples), consider the two situations illustrated in Figure 7.

In the figure on the left, the training samples have a nonlinear distribution within the subspace. For the given positive test sample "+," only two training samples are needed to represent it well

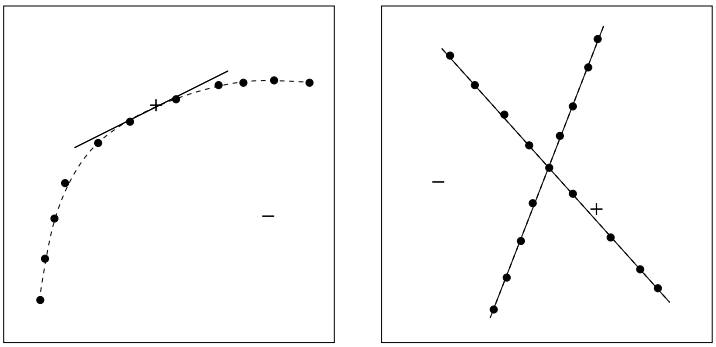

Fig. 7. A sparse solution within the subspace spanned by all training samples of one subject. Left: the samples exhibit a nonlinear distribution within the subspace. Right: the samples lie on two lower-dimensional subspaces within the subspace spanned by all the samples.

linearly. For the other negative test sample "-," although it is inside the subspace spanned by all the samples, it deviates significantly from the sample distribution. In the figure on the right, the training samples of one subject are distributed on two lower-dimensional subspaces. This could represent the situation when the training images contain both varying illuminations and expressions. Again, for a positive test sample "+," typically a small subset of the training samples are needed to represent it well. But if we use the span of all the samples, that could easily over-fit negative samples that do not belong to the same subject. For example, as we have shown in Figure 3, although Subject 1 has 32 training examples, the test image is well represented using less than 5 large coefficients. In other words, l^1-minimization is very efficient in harnessing sparse structures even within each face subspace.

From our discussions above, we see that the ℓ^1-minimization based classifier works under a wider range of conditions than NN and NS combined. It strikes a good balance between NN and NS: To avoid under-fitting, it uses multiple (instead of the nearest one) training images in each class to linearly extrapolate the test image, but it uses only the smallest necessary number of them to avoid over-fitting. For each test image, the number of samples needed is automatically determined by the ℓ^1-minimization, because in terms of finding the sparse solution $\boldsymbol{x}$, the ℓ^1-minimization is equivalent to the ℓ^0-minimization. As a result, the classifier can better exploit the actual (possibly multi-modal and nonlinear) distributions of the training data of each subject and is therefore likely to be more discriminative among multiple classes. These advantages of Algorithm 1 are corroborated by experimental results presented in Section VI.

IV. VALIDATION OF THE TEST IMAGE

Validation is a problem closely related to but *different* from identification. Given a test image, before we identify which subject it is, we first need to decide if it is a valid image of one of the subjects in the dataset. The ability to detect and then reject invalid test images, also known as "outliers," is crucial for a recognition system to work in a real-world situation: the system can be given a face image of a subject that is not in the dataset, or an image that is not a face at all.

In the NN or NS paradigm, the residuals $r_i(\boldsymbol{y})$ are also used for validation, in addition to identification. That is, the algorithm accepts or rejects a test image based on how small the smallest residual is. However, each residual $r_i(\boldsymbol{y})$ is computed without any knowledge of images of other subjects in the training dataset and only measures similarity between the test image and each individual subject. In the sparse representation paradigm, the residuals $r_i(\boldsymbol{y})$ are computed globally, in terms of images of all subjects. In a sense, it can potentially harness the joint distribution of all subjects for validation.

From Algorithm 1, in addition to the residuals $r_i(\boldsymbol{y})$, we also obtain the coefficients $\boldsymbol{x}$. We contend that the coefficients $\boldsymbol{x}$ are better statistics for validation than the residuals. Let us first see an example. We randomly select an irrelevant image from Google, and down-sample it to 12×10. We then compute the sparse representation of the image against the same training data as in Example 1. Figure 8 top plots the obtained coefficients, and bottom plots the corresponding residuals. Compared to the coefficients of a valid test image in Figure 3, notice that the coefficients $\boldsymbol{x}$ here are not concentrated on any one subject and instead spread widely across the entire training set. Thus, the distribution of the estimated sparse coefficients $\boldsymbol{x}$ contains important information about the validity of the test image: A valid test image should have a sparse representation whose nonzero entries concentrate mostly on one subject, whereas an invalid image has sparse coefficients spread widely among multiple subjects.

To quantify this observation, we define the following measure of how concentrated the coefficients are on a single subject in the dataset:

Definition 1 (Sparsity Concentration Index): Suppose $\boldsymbol{x}$ is the sparse solution to either (P_1) (7) or (P_1') (11). The *sparsity concentration index* (SCI) of $\boldsymbol{x}$ is defined as

$$\text{SCI}(\boldsymbol{x}) \doteq \frac{k \cdot \max_i \|\delta_i(\boldsymbol{x})\|_1 / \|\boldsymbol{x}\|_1 - 1}{k - 1} \quad \in [0, 1]. \tag{20}$$

Obviously, when SCI = 1, the test image is represented using only images from a single subject,

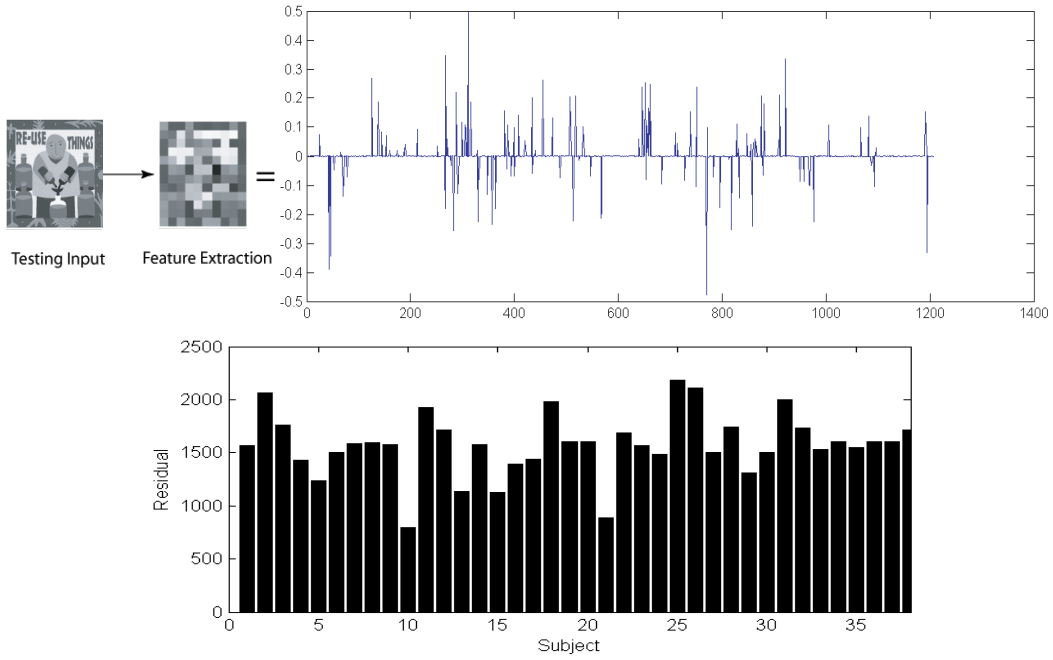

Fig. 8. Top: The sparse coefficients for an invalid test image *w.r.t.* the same training data set from Example 1. The test image is a randomly selected irrelevant image. The values of the sparse coefficients recovered from Algorithm 1 are plotted on the right. Bottom: The residuals of the invalid test image *w.r.t.* the projection $\delta_i(\boldsymbol{x})$ of the sparse representation computed by ℓ^1-minimization. The ratio between the magnitudes of the two smallest residuals is about 1:1.2.

and when SCI = 0, the sparse coefficients are spread evenly over all classes. Thus, we choose a threshold $\tau \in (0, 1)$ and accept a test image as valid if

$$\text{SCI}(\boldsymbol{x}) \geq \tau, \tag{21}$$

and reject as invalid otherwise.

Unlike NN or NS, this new rule completely avoids the use of the residuals $r_i(\boldsymbol{y})$ for validation. Notice that in Figure 8, even for a non-face image, with a large training set, the smallest residual of the invalid test image is not so large at all. Rather than relying on a single statistic for both validation and identification, our approach separates the information required for these tasks: the residuals for identification and the sparse coefficients for validation. In a sense, the residual measures how well the representation approximates the test image; and the sparsity concentration index measures how good the representation itself is, in terms of localization.

Another benefit of this approach is improved validation performance against generic face images. A generic face might be rather similar to some of the subjects in the dataset and

may have small residuals w.r.t. their training images. Using residuals for validation more likely leads to a false positive. But a generic face is unlikely to pass the new validation rule as a good representation of it typically requires contribution from images of multiple subjects in the dataset. Thus, the new rule can better judge whether the test image is a generic face or the face of one particular subject in the dataset. In Section VI-C we will demonstrate that the new validation rule outperforms the NN and NS methods, with as much as 10–20% improvement in verification rate for a given false accept rate (see Figure 11).

V. Feature Selection and Randomfaces

With Algorithm 1 in place, the remaining question is how the choice of the feature transform R affects its recognition performance. Obviously, R affects the performance through the matrix $\tilde{A} = RA$'s ability to recover sparse solutions via ℓ^1-minimization. When the number of non-zero entries in $\boldsymbol{x}$ increases beyond a critical value, ℓ^1-minimization no longer always finds the correct, sparse $\boldsymbol{x}_0$. This value is called the *equivalence breakdown point* of $\tilde{A}$ [23]. Different $\tilde{A}$'s have different breakdown points. Although there is no known closed-form formula nor polynomial-time algorithm to determine the breakdown point of a given matrix, in our context the large number of subjects typically ensures us that the algorithm operates well below the breakdown point. Therefore, the sparse solution $\boldsymbol{x}$ will not depend so much on the chosen transformation R and subsequently, the recognition rate of Algorithm 1 will be similar for different features (either using fixed filter banks or adaptive bases).

For extant face recognition methods, it is known that increasing the dimension of the feature space generally improves the recognition rate, as long as the feature distribution does not become degenerate [10]. However, degeneracy is no longer an issue for our algorithm since ℓ^1-minimization properly regularizes linear regression [22]. We can use very high-dimensional feature spaces with little concern about degeneracy. In addition, it is easy to show that the breakdown point of $\tilde{A}$ increases with d [15], [25]. As we will demonstrate in Section VI, the performance of our algorithm improves gracefully as d increases. The optimization problem (P_1) or (P_1') can be efficiently solved by linear programming or convex optimization, allowing us to experiment with feature space dimensions up to $d = 12,000$.

Algorithm 1's ability to handle high-dimensional features allows us to observe an important phenomenon about feature selection, which is unique in the framework of sparse representation.

Theoretical results of [15], [25] have shown that if the signal $\boldsymbol{x}$ is sparse, then with overwhelming probability, it can be correctly recovered via ℓ^1-minimization from *any* sufficiently large dimension d of linear measurements $\tilde{\boldsymbol{y}}$. More precisely, if $\boldsymbol{x}$ has $k \ll n$ nonzeros, then

$$d \geq 2k \log(n/d) \tag{22}$$

random measurements are sufficient for sparse recovery with high probability [32]. This surprising phenomenon has been dubbed the "blessing of dimensionality" [15], [21]. Thus, one should expect to see similarly good recognition performance from Algorithm 1 even with randomly selected facial features:

Definition 2 (Randomfaces): Consider a transform matrix $R \in \mathbb{R}^{d \times m}$ whose entries are independently sampled from a zero-mean normal distribution and each row is normalized to unit length. These row vectors of R can be viewed as d random faces in $\mathbb{R}^m$.

Random projection has been previously studied as a general dimensionality reduction method for numerous clustering problems [33]–[35], as well as for learning nonlinear manifolds [36], [37]. Regardless of whether the estimated signal is sparse or not, random projection has the following advantages over classical methods such as *principal component analysis* (PCA) and *linear discriminant analysis* (LDA):

1) The computation of a random matrix R does not rely on a specific (good) training set, *i.e.,* it is data independent.
2) A random matrix is extremely efficient to generate, even in very high-dimensional feature spaces.

These advantages make random projection an very promising approach to dimensionality reduction in many practical applications of face recognition. For instance, a face-recognition system for access control may not be able to acquire in advance a complete database of all subjects of interest, and the subjects in the database may change dramatically over time. When there is a new subject added to the database, there is no need for recomputing the random transformation R.

One concern about random projection is its stability, *i.e.,* for an individual trial, the selected features could be bad [38]. Our experiments in Section VI show that for face recognition this is usually not a problem as long as the number of features, d, is sufficiently large. Nevertheless, one

can always ensure higher stability by aggregating the results from multiple random projections. We next outline one simple scheme to do so.

In Algorithm 1, the classification of a test sample $\boldsymbol{y}$ depends on the residuals $r_i, i = 1, \ldots, k$, *w.r.t.* the k classes. When the projection matrix R is randomly generated, we seek a robust estimate of the residuals r_i by using an ensemble of Randomface matrices $R^j, j = 1, \ldots, l$. Define r_i^j as the residual of the test image *w.r.t.* the i-th class using the projection matrix R^j in Algorithm 1, then the empirical average of the i-th residual over l projections is

$$E[r_i] = \frac{1}{l} \sum_{j=1,\ldots,l} r_i^j. \tag{23}$$

Hence the identity of $\boldsymbol{y}$ can be assigned as

$$\text{identity}(\boldsymbol{y}) = \arg\min_i E[r_i]. \tag{24}$$

The optimal membership estimate corresponds to the minimum average residual using the sparse representations over l randomly generated projection matrices R.

The modified algorithm is summarized as Algorithm 2. The improvement of this algorithm over Algorithm 1 will be demonstrated in Section VI.

Algorithm 2 (Recognition via an Ensemble of Randomfaces)

1: **Input:** a matrix of training images $A \in \mathbb{R}^{m \times n}$ for k subjects, a test image $\boldsymbol{y} \in \mathbb{R}^m$, and an error tolerance ϵ.

2: Generate l random projection matrices $R^1, \ldots, R^l \in \mathbb{R}^{d \times m}$,

3: **for all** $j = 1, \ldots, l$ **do**

4: Compute features $\tilde{\boldsymbol{y}} = R^j \boldsymbol{y}$ and $\tilde{A} = R^j A$, normalize $\tilde{\boldsymbol{y}}$ and columns of $\tilde{A}$ to unit length.

5: Solve the convex optimization problem (P_1'):

$$\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|\tilde{\boldsymbol{y}} - \tilde{A}\boldsymbol{x}\|_2 \leq \epsilon.$$

6: Compute $r_i^j(\boldsymbol{y}) = \|\tilde{\boldsymbol{y}} - \tilde{A}\,\delta_i(\boldsymbol{x})\|_2$, for $i = 1, \ldots, k$.

7: **end for**

8: For each class i, $E[r_i] \leftarrow \text{mean}\{r_i^1, \ldots, r_i^l\}$.

9: **Output:** $\text{identity}(\boldsymbol{y}) = \arg\min_i E[r_i]$.

VI. Experimental Verification

In this section, we quantitatively verify the performance of Algorithm 1 and Algorithm 2 using two public face databases, namely, the Extended Yale B database [39] and the AR database [40]. The ℓ^1-minimization in our algorithm is based on the "ℓ^1-magic" MATLAB toolbox at: `http://www.acm.caltech.edu/l1magic/`. The MATLAB implementation of our algorithms only takes a few seconds to classify one test image on a typical 3G Hz PC.

We compare the two algorithms with two classical algorithms, namely, *nearest neighbor* (NN) and *nearest subspace* (NS), discussed in the previous section. We denote the results from Algorithm 2 on Randomface ensembles as "E-Random" in this section (five random projection matrices are used to compute the average). For all the experiments, the error distortion ϵ for Algorithm 1 and Algorithm 2 is set to be $\epsilon = 0.05$.

A. Boosting the Performance of X-Face Features

We first test our algorithms using several conventional holistic face features, namely, Eigenfaces, Laplacianfaces, and Fisherfaces. We compare their performance with two unconventional features: Randomfaces and down-sampled images.

1) Extended Yale B Database: The Extended Yale B database consists of 2,414 frontal-face images of 38 individuals. The cropped and normalized 192×168 face images were captured under various laboratory-controlled lighting conditions. To fairly compare the performance, for each subject, we randomly select half of the images for training (*i.e.,* about 32 images per subject), and the other half for testing. The reason for randomly choosing the training set is to make sure that our results and conclusions will not depend on any special choice of the training data.

We chose to compute the recognition rates with the feature space dimensions 30, 56, 120, and 504, respectively. Those numbers correspond to the dimensions of the down-sampled image with the ratios 1/32, 1/24, 1/16, and 1/8, respectively.[9] Notice that Fisherfaces are different from the other features because the maximal number of valid Fisherfaces is one less than the number

[9]We cut off the dimension at 504 as the implementation of Eigenfaces and Laplacianfaces reaches the memory limit of MATLAB. Although our algorithm persists to work far beyond on the same computer, 504 is already sufficient to reach all our conclusions.

TABLE I

RECOGNITION RATES OF ALGORITHM 1 (OR 2) ON THE EXTENDED YALE B DATABASE. THE BOLD NUMBERS INDICATE THE BEST AMONG ALL FEATURES.

Dimension (d)	30	56	120	504
Eigen [%]	86.5	91.63	93.95	96.77
Laplacian [%]	87.49	91.72	93.95	96.52
Random [%]	82.6	91.47	95.53	98.09
Downsample [%]	74.57	86.16	92.13	97.1
Fisher [%]	86.91	N/A	N/A	N/A
E-Random [%]	**90.72**	**94.12**	**96.35**	**98.26**

TABLE II

RECOGNITION RATES OF NEAREST NEIGHBOR (LEFT) AND NEAREST SUBSPACE (RIGHT) ON THE EXTENDED YALE B DATABASE. THE BOLD NUMBERS INDICATE THE BEST AMONG ALL FEATURES.

Dimension (d)	30	56	120	504
Eigen [%]	74.32	81.36	85.50	88.40
Laplacian [%]	77.13	**83.51**	**87.24**	**90.72**
Random [%]	70.34	75.56	78.79	79.04
Downsample [%]	51.69	62.55	71.58	77.96
Fisher [%]	**87.57**	N/A	N/A	N/A

Dimension (d)	30	56	120	504
Eigen [%]	**89.89**	**91.13**	92.54	93.21
Laplacian [%]	88.98	90.39	91.88	93.37
Random [%]	87.32	91.47	**93.87**	**94.12**
Downsample [%]	80.78	88.15	91.13	93.37
Fisher [%]	81.94	N/A	N/A	N/A

of classes k [3], which is 38 in our case. As a result, the recognition result for Fisherfaces is only available at dimension 30 in our experiment.

The subspace dimension for the NS algorithm is 9, which has been mostly agreed upon in the literature for processing facial images with only illumination change.[10] Tables I and II show the recognition rates for the X-face features.

These recognition rates shown in Table II are consistent with those that have been reported in the literature, although some reported on different databases or with different training subsets.

[10]We have experimented with other subspace dimensions that are either less or greater than 9, and they eventually led to a decrease in performance.

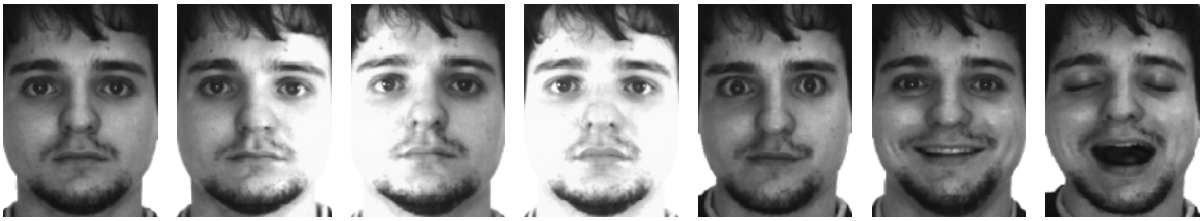

Fig. 9. The seven training images of an individual in the AR database.

For example, He *et. al.* [4] reported the best recognition rate of 75% using Eigenfaces at 33 dimension, and 89% using Laplacianfaces at 28 dimension on the Yale face database, both using NN. In [31], Lee *et. al.* reported 95.4% accuracy using the NS method on the Yale B database.

2) AR Database: The AR database consists of over 4,000 frontal images for 126 individuals. For each individual, 26 pictures were taken in two separate sessions. These images include more facial variations including illumination change, expressions, and facial disguises comparing to the Extended Yale B database. In the experiment, we chose a subset of the dataset consisting of 50 male subjects and 50 female subjects. For each subject, 14 images with only illumination change and expressions were selected[11]: The seven images from Session 1 for training, and the other seven from Session 2 for testing. The images are properly cropped with dimension 165×120, and all converted to grayscale. We selected four feature space dimensions: 30, 54, 130, and 540, which correspond to the down-sample rations 1/24, 1/18, 1/12, and 1/6, respectively. Because the number of subjects is 100, results for Fisherfaces are only given at dimension 30 and 54.

This database is substantially more challenging than the Yale database, since the number of subjects is now 100 but the training images is reduced to seven per subject: Four neutral faces with different lighting conditions and three faces with different expressions (see Figure 9 for an example).

For NS, since the number of training images per subject is seven, any estimate of the face subspace cannot have dimension higher than 7. We chose to keep all seven dimensions for NS in this case. Tables III and IV show the recognition rates for the X-face features.

Based on the results on the Extended Yale B database and the AR database, we draw the

[11]Please refer to the companion submission [18] for our proposed treatment enforcing the sparsity constraint in the presence of (facial) image occlusion.

TABLE III

RECOGNITION RATES OF ALGORITHM 1 (OR 2) ON THE AR DATABASE. THE BOLD NUMBERS INDICATE THE BEST AMONG ALL FEATURES.

Dimension (d)	30	54	130	540
Eigen [%]	71.14	80	85.71	91.99
Laplacian [%]	73.71	84.69	90.99	94.28
Random [%]	57.8	75.54	87.55	94.7
Downsample [%]	46.78	67	84.55	93.85
Fisher [%]	**86.98**	**92.27**	N/A	N/A
E-Random [%]	78.54	85.84	**91.23**	**94.99**

TABLE IV

RECOGNITION RATES OF NEAREST NEIGHBOR (LEFT) AND NEAREST SUBSPACE (RIGHT) ON THE AR DATABASE. THE BOLD NUMBERS INDICATE THE BEST AMONG ALL FEATURES.

Dimension (d)	30	54	130	540
Eigen [%]	68.10	74.82	79.26	80.54
Laplacian [%]	73.10	77.11	**83.83**	**89.70**
Random [%]	56.65	63.66	71.39	74.96
Downsample [%]	51.65	60.94	69.24	73.68
Fisher [%]	**83.40**	**86.84**	N/A	N/A

Dimension (d)	30	54	130	540
Eigen [%]	64.09	77.11	81.97	85.12
Laplacian [%]	65.95	77.54	**84.26**	**90.27**
Random [%]	59.23	68.24	79.97	83.26
Downsample [%]	56.22	67.67	76.97	82.12
Fisher [%]	**80.26**	**85.84**	N/A	N/A

following conclusions:

1) In general for both the Yale database and AR database, the best performance of Algorithm 1 and Algorithm 2 consistently outperforms the other two classical methods NN and NS at each individual feature dimension, and by a large margin. By imposing *sparsity* via ℓ^1-minimization, the recognition rates of all features improve and converge gracefully when the feature dimension increases, for both the Yale and AR databases. More specifically, the best recognition rate for the Yale database via ℓ^1-minimization is 98.3%, compared to 90.7% using NN and 94% using NS; the best rate for the AR database via ℓ^1-minimization is 95%, compared to 89.7% for NN and 90% for NS.

2) In Table I and Table III, the experiments have successfully verified that when sparsity is properly harnessed, highly accurate recognition can be achieved even with the "unconventional" down-sampled images or Randomface features. For example in Table III, the performance of the down-sampled image features gracefully increased from 46.8% at 30-D to 93.9% at 540-D. In comparison, the same features only achieve 73.7% for NN and 82.1% for NS at 540-D, when such sparsity constraint is not utilized.
3) The results corroborate the theory of compressed sensing, which suggests that $d \approx 128$, according to equation (22), random linear measurements should be sufficient for sparse recovery in the Yale database, while $d \approx 88$ random linear measurements should suffice for sparse recovery in the AR database [32]. Beyond these dimensions, the performances of various features in conjunction with ℓ^1-minimization converge, with a single randomly chosen projection performing the best (98.1% recognition rate on Yale, 94.7% on AR).
4) Algorithm 2 further enhances the performance of Randomfaces. For all feature dimensions on the Yale database and the highest two dimensions on the AR database, Randomface ensembles achieve the highest accuracy over all other facial features.
5) From the results of NN and NS in Table II and Table IV, the choice of a good combination of features and classifiers indeed makes some difference. For example, NS outperforms NN in most cases on both databases. The Fisherface features excel in almost all low-dimensional facial feature spaces for NN and NS, while it is the Laplacianfaces that achieve the highest accuracy in higher-dimensional feature spaces.

B. Partial Face Features

There have been extensive studies in both human vision and computer vision literature about the effectiveness of partial features in recovering the identity of a human face, *e.g.,* see [1], [26]. As a second set of experiments, we test Algorithm 1 on the following three partial facial features: nose, right eye, and mouth & chin. We use the Extended Yale B database for the experiment and the training and test datasets are the same as the experiment in subsection VI-A.1. See Figure 10 for a typical example of the extracted features.

Notice that the dimension d of either feature is larger than the number of training samples ($n = 1,207$), and the linear system (5) to be solved becomes over-determined. Nevertheless, we apply the same Algorithm 1 anyway to encourage sparsity of the solution. The results in Table

Fig. 10. Illustration of the three partial features. Left: Nose. Middle: Right eye. Right: Mouth & chin.

TABLE V

LEFT: EXAMPLE OF RIGHT-EYE (RE) FEATURE AND RIGHT-FACE (RF) FEATURE. RIGHT: RECOGNITION RATES OF ALGORITHM 1, NN, AND NS ON THE EXTENDED YALE B DATABASE.

Features	Nose	Right Eye	Mouth & Chin
Dimension (d)	4,270	5,040	12,936
Algorithm 1 [%]	**87.32**	**93.7**	**98.32**
NS [%]	83.68	78.62	94.37
NN [%]	49.21	68.77	72.66

V again show that the proposed algorithm achieves much better recognition rates than both NN and NS, which again demonstrate the significance of imposing the sparsity constraint in face recognition. These experiments also shows the scalability of the proposed algorithm in working with features of over 10^4 dimensions.

C. Receiver Operating Characteristics

In this experiment, we verify how Algorithm 1, together with the outlier rejection rule (21) given in Section IV, can effectively detect and reject invalid testing images. Conventionally, the two major indices used to measure the accuracy of outlier rejection are the *false acceptance rate* (FAR) and the *verification rate* (VR). False acceptance rate calculates the percentage of test samples that are accepted and wrongly classified. Verification rate is one minus the percentage of valid test samples that are wrongfully rejected. A good recognition system should achieve high

verification rates even at very low false acceptance rates. Therefore, the accuracy and reliability of a recognition system are typically evaluated by the FAR-VR curve (sometimes it is loosely identified as the *receiver operating characteristic* (ROC) curve).

In this experiment, we only use the more challenging AR dataset – more subjects and more variability in the testing data make outlier rejection a more relevant issue. The experiments are run under two different settings. The first setting is the same as in subsection VI-A.2: 700 training images for all 100 subjects and another 700 images for testing. So in this case, there is no real outliers. The role of validation is simply to reject test images that are difficult to classify. In the second setting, we remove the training samples of every third of the subjects and add them into the test set. That leaves us 469 training images for 67 subjects and 700 + 231 = 931 testing images for all 100 subjects. So about half of the test images are true outliers.[12] We compare three algorithms: Algorithm 1, NS, and NS. To be fair, all three algorithms use exactly the same features, 504-dimensional eigenfaces.[13]

Figure 11 shows the FAR-VR curves obtained under the two settings. Notice that Algorithm 1 significantly outperforms NS and NN, as expected. In our companion paper [18], we have also computed the ROC curves for the Extended Yale B dataset using the entire image as feature. We observe there that the validation performance of Algorithm 1 improves much further with the full image whereas the other methods do not – their performance saturates when the feature dimension is beyond a few hundred.

VII. Conclusions and Discussions

In this paper, we have contended both theoretically and experimentally that exploiting sparsity is critical for high-performance face recognition. With sparsity properly harnessed, the choice of features becomes less important than the number of features used (in our face recognition example, approximately 100 are sufficient to make the difference negligible). Furthermore, when the number of features are large enough (from our experiments, approximately 500), even randomly generated features or severely down-sampled images are just as good as conventional face features such as Eigenfaces and Fisherfaces. This revelation almost goes against conventional

[12]More precisely, 462 out of the 931 test images belong to subjects not in the training set.

[13]Notice that according to Table III, among all 504-D features, eigenfaces are in fact the worst for our algorithm. But we use it anyway as this gives a baseline performance for our algorithm.

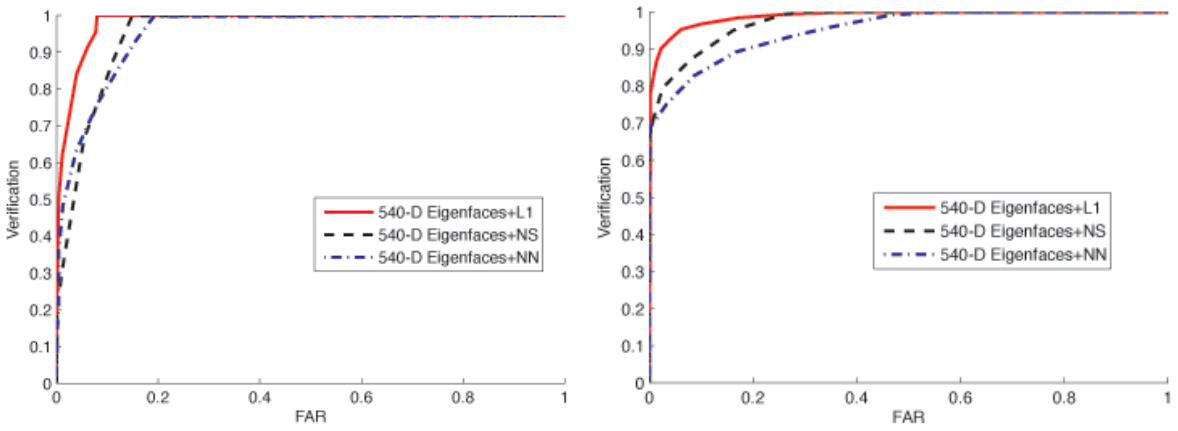

Fig. 11. The FAR-VR curves (in red color) of Algorithm 1 using Eigenfaces. They are compared with the curves of NS and NN using Eigenfaces. Left: 700 images for all 100 subjects in the training, no real outliers in the 700 test images. Right: 469 images for 67 subjects in the training, about half of the 931 test images are true outliers.

wisdom, but is strongly corroborated by the emerging mathematical theory of compressed sensing [15], [32].

Notice that in this paper, we never explicitly use any illumination model to reduce the effect of lighting on the faces, unlikely the methods that use quotient or self-quotient images [20], [41]. In a related study, we have verified empirically that using self-quotient images improves our algorithm only slightly, suggesting that our algorithm can already handle illumination without any preprocessing. The linear representation model (3) is the only assumption that we made about the relationship between the test image and the training images. As we see from the results on the AR database, this model apparently works well with variation in facial expression too.

The conclusions drawn in this paper apply to any object recognition problems where the linear feature model (5) is valid or approximately so. However, for face recognition with pose variation, the linear model may no longer be accurate. Most existing solutions have therefore relied on nonlinear kernel methods that render nonlinear face structures linearly separable [10], [26], [42]. Yet practitioners are faced with similar over-fitting problems in even higher-dimensional kernel spaces. We believe the proposed classification framework via ℓ^1-minimization may also provide new solutions for such kernel-based methods. Furthermore, in the companion paper [18], we have demonstrated how sparsity also plays a crucial role in face recognition when the test images are

severely corrupted and occluded.

REFERENCES

[1] P. Sinha, B. Balas, Y. Ostrovsky, and R. Russell, "Face recognition by humans: Nineteen results all computer vision researchers should know about," *Proceedings of the IEEE*, vol. 94, no. 11, pp. 1948–1962, 2006.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 1991.

[3] P. Belhumeur, J. Hespanda, and D. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[4] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, 2005.

[5] J. Kim, J. Choi, J. Yi, and M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1977–1981, 2005.

[6] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2001, pp. 1–6.

[7] B. Heisele, T. Serre, and T. Poggio, "A component-based framework for face detection and identification," *to appear in International Journal of Computer Vision*, 2007.

[8] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2001.

[9] J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman, "Clustering appearances of objects under varying illumination conditions," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2003, pp. 11–18.

[10] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 725–737, 2006.

[11] P. Phillips, W. Scruggs, A. O'Tools, P. Flynn, K. Bowyer, C. Schott, and M. Sharpe, "FRVT 2006 and ICE 2006 large-scale results," NIST, Tech. Rep. NISTIR 7408, 2007.

[12] J. Mutch and D. Lowe, "Multiclass object recognition with sparse, localized features," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 17–22.

[13] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–793, 1999.

[14] D. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ^1-norm near solution approximates the sparest solution," *preprint*, 2004.

[15] E. Candès, "Compressive sampling," in *Proceedings of the International Congress of Mathematicians*, 2006.

[16] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, pp. 237–260, 1998.

[17] D. Donoho and Y. Tsaig, "Fast solution of ℓ^1-norm minimization problems when the solution may be sparse," preprint, *http://www.stanford.edu/ tsaig/research.html*, 2006.

[18] J. Wright, A. Ganesh, A. Yang, and Y. Ma, "Robust face recognition via sparse representation," *Technical Report, University of Illinois, submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.

[19] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 2, pp. 218–233, 2003.

[20] H. Wang, S. Li, and Y. Wang, "Generalized quotient image," in *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, 2004, pp. 498–505.

[21] D. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, 2000.

[22] P. Bickel and B. Li, "Regularization in statistics," *Test*, vol. 15, no. 2, pp. 271–344, 2006.

[23] D. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Comm. on Pure and Applied Math*, vol. 59, no. 6, pp. 797–829, 2006.

[24] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. on Pure and Applied Math*, vol. 59, no. 8, pp. 1207–1223, 2006.

[25] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[26] M. Savvides, R. Abiantun, J. Heo, S. Park, C. Xie, and B. Vijayakumar, "Partial and holistic face recognition on FRGC-II data using support vector machine kernel correlation feature analysis," in *CVPR Workshop*, 2006.

[27] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, pp. 237–260, 1998.

[28] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[29] D. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," *preprint*, 2005.

[30] Y. Sharon, J. Wright, and Y. Ma, "Computation and relaxation of conditions for equivalence between ℓ^1 and ℓ^0 minimization," *submitted to IEEE Transactions on Information Theory*, 2007.

[31] K. Lee, J. Ho, and D. Kriegman, "Acuiqring linear subspaces for face recognition under variable lighting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[32] D. Donoho and J. Tanner, "Counting faces of randomly projected polytopes when the projection radically lowers dimension," preprint*, http://www.math.utah.edu/ tanner/*, 2007.

[33] S. Kaski, "Dimensionality reduction by random mapping," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, 1998, pp. 413–418.

[34] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the ACM Symposium on Principles of Database Systems*, 2001, pp. 274–281.

[35] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2001, pp. 245–250.

[36] R. Baraniuk and M. Wakin, "Random projections of smooth manifolds," *preprint*, 2006.

[37] R. Baraniuk, M. Davenport, R. de Vore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets compressed sensing," *to appear in Constructive Approximation*, 2007.

[38] X. Fern and C. Brodley, "Random projection for high dimensional data clustering: A cluster ensemble approach," in *Proceedings of the International Conference on Machine Learning*, 2003.

[39] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[40] A. Martinez and R. Benavente, "The AR face database," CVC Technical Report No. 24, Tech. Rep., 1998.

[41] A. Shashua and T. Riklin-Raviv, "The quotient image: Class-based re-rendering and recognition with varying illuminations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 129–139, 2001.

[42] M. Yang, "Kernel Eigenfaces vs. kernel Fisherfaces: face recognition using kernel methods," in *Proceedings of the Fifth International Conference on Automatic Face and Gesture Recognition*, 2002, pp. 215–220.