Robust Face Recognition via Sparse Representation

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

***Abstract*— We consider the problem of automatically recognizing human faces from frontal views with varying expression and illumination, as well as occlusion and disguise. We cast the recognition problem as one of classifying among multiple linear regression models, and argue that new theory from sparse signal representation offers the key to addressing this problem. Based on a sparse representation computed by ℓ^1-minimization, we propose a general classification algorithm for (image-based) object recognition. This new framework provides new insights into two crucial issues in face recognition: *feature extraction* and *robustness to occlusion*. For feature extraction, 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 sufficiently large and whether the sparse representation is correctly computed. Unconventional features such as downsampled images and random projections perform just as well as conventional features such as Eigenfaces and Laplacianfaces, as long as the dimension of the feature space surpasses certain threshold, predicted by the theory of sparse representation. This framework can handle errors due to occlusion and corruption uniformly, by exploiting the fact that these errors are often sparse w.r.t. to the standard (pixel) basis. The theory of sparse representation helps predict how much occlusion the recognition algorithm can handle and how to choose the training images to maximize robustness to occlusion. We conduct extensive experiments on publicly available databases to verify the efficacy of the proposed algorithm, and corroborate the above claims.**

***Index Terms*— Face Recognition, Feature Extraction, Occlusion and Corruption, Sparse Representation, Compressed Sensing, ℓ^1-Minimization, Validation and Outlier Rejection.**

I. Introduction

Parsimony has a rich history as a guiding principle for inference. One of its most celebrated instantiations, the principle of minimum description length in model selection [1], [2], stipulates that within a hierarchy of model classes, the model that yields the most compact representation should be preferred for decision-making tasks such as classification. A related, but simpler, measure of parsimony in high-dimensional data processing seeks models that depend on only a few of the observations, selecting a small subset of features for classification or visualization (e.g., Sparse PCA [3], [4] amongst others). Such sparse feature selection methods are, in a sense, dual to the support vector machine (SVM) approach of [5], [6], which instead selects a small subset of relevant training examples to characterize the decision boundary between classes. While these works comprise only a small fraction of the literature on parsimony for inference, they do serve to illustrate a common theme: all of them use parsimony as a principle for choosing a limited subset of features or models from the training data, rather than directly using the data for representing or classifying an input (test) signal.

The role of parsimony in human perception has also been strongly supported by studies of human vision. Investigators have recently revealed that in both low-level and mid-level human vision [7], [8], many neurons in the visual pathway are selective for a variety of specific stimuli, such as color, texture, orientation, scale, and even view-tuned object images. Considering these neurons to form an overcomplete dictionary of base signal elements at each visual stage, the firing of the neurons w.r.t. to a given input image is typically highly sparse.

In the statistical signal processing community, the algorithmic problem of computing sparse linear representations w.r.t. to an overcomplete dictionary of base elements or signal atoms has seen a recent surge of interest [9]–[12].[1] Much of this excitement centers around the discovery that whenever the optimal representation is *sufficiently sparse*, it can be efficiently computed by convex optimization [9], even though this problem can be extremely difficult in the general case [13]. The resulting optimization problem, similar to the Lasso in statistics [12], [14] penalizes the ℓ^1-norm of the coefficients in the linear combination, rather than the directly penalizing the number of nonzero coefficients (i.e., the ℓ^0-norm).

The original goal of these works was not inference or classification *per se*, but rather representation and compression of signals, potentially using lower sampling rates than the Shannon-Nyquist bound [15]. Algorithm performance was therefore measured in terms of sparsity of the representation and fidelity to the original signals. Furthermore, individual base elements in the dictionary were not assumed to have any particular semantic meaning – they are typically chosen from standard bases (e.g., Fourier, Wavelet, Curvelet, Gabor), or even generated from random matrices [11], [15]. Nevertheless, the sparsest representation *is* naturally discriminative: amongst all subsets of base vectors, it selects the subset which most compactly expresses the input signal and rejects all other possible but less compact representations.

In this paper, we exploit the discriminative nature of sparse representation to perform *classification*. Instead of using the generic dictionaries discussed above, we represent the test sample in an overcomplete dictionary whose base elements are *the training samples themselves*. If sufficient training samples are available

J. Wright, A. Ganesh, and Y. Ma are with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign. A. Yang and S. Sastry are with the Department of Electrical Engineering and Computer Science, University of California at Berkeley. Corresponding author: Yi Ma, 145 Coordinated Science Laboratory, 1308 W. Main St., Urbana, Illinois 61801. Email: yima@uiuc.edu. Tel: 1-217-244-0871. Fax: 1-217-244-2352.

[1] In the literature, the terms "sparse" and "representation" have been used to refer to a number of similar concepts. Throughout this paper, we will use the term "sparse representation" to refer specifically to an expression of the input signal as a linear combination of base elements in which many of the coefficients are zero. In most cases considered, the percentage of nonzero coefficients will vary between zero and $\approx 30\%$. However, in characterizing the breakdown point of our algorithms, we will encounter cases with up to 70% nonzeros.

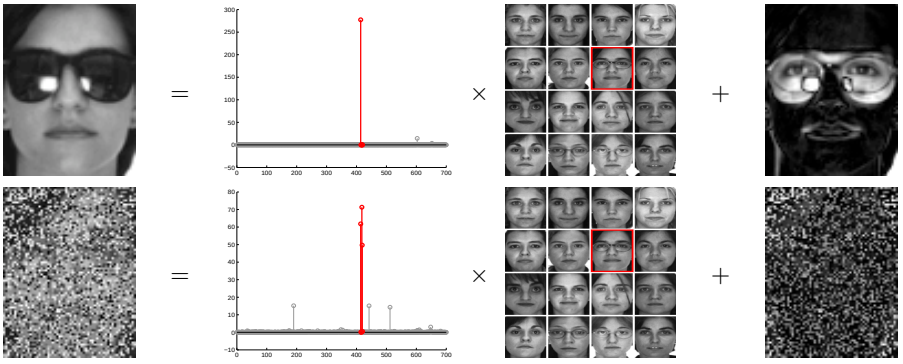


Fig. 1. **Overview of our approach.** Our method represents a test image (left), which is potentially occluded (top) or corrupted (bottom), as a sparse linear combination of all the training images (middle) plus sparse errors (right) due to occlusion or corruption. Red (darker) coefficients correspond to training images of the correct individual. Our algorithm determines the true identity (indicated with a red box at second row and third column) from 700 training images of 100 individuals (7 each) in the standard AR face database.

from each class,[2] it will be possible to represent the test samples as a linear combination of just those training samples from the same class. This representation is naturally sparse, involving only a small fraction of the overall training database. We argue that in many problems of interest, it is actually the *sparsest* linear representation of the test sample in terms of this dictionary, and can be recovered efficiently via ℓ^1-minimization. Seeking the sparsest representation therefore automatically discriminates between the various classes present in the training set. Figure 1 illustrates this simple idea using face recognition as an example. Sparse representation also provides a simple and surprisingly effective means of rejecting invalid test samples not arising from any class in the training database: these samples' sparsest representations tend to involve many dictionary elements, spanning multiple classes.

Our use of sparsity for classification differs significantly from the various parsimony principles discussed above. Instead of using sparsity to identify a relevant model or relevant features that can later be used for classifying *all* test samples, it uses the sparse representation of each individual test sample directly for classification, adaptively selecting the training samples that give the most compact representation. The proposed classifier can be considered a generalization of popular classifiers such as *nearest neighbor* (NN) [18] and *nearest subspace* (NS) [19] (i.e., minimum distance to the subspace spanned all training samples from each object class). Nearest neighbor classifies the test sample based on the best representation in terms of a single training sample, whereas nearest subspace classifies based on the best linear representation in terms of all the training samples in each class. The *nearest feature line* (NFL) algorithm [20] strikes a balance between these two extremes, classifying based on the best affine representation in terms of a pair of training samples. Our method strikes a similar balance, but considers all possible supports (within each class or across multiple classes) and adaptively chooses the minimal number of training samples needed to represent each test sample.[3]

We will motivate and study this new approach to classification within the context of automatic face recognition. 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 [21], and partly due to numerous important applications for face recognition technology [22]. In addition, technical issues associated with face recognition are representative of object recognition and even data classification in general. Conversely, the theory of sparse representation and compressed sensing yields new insights into two crucial issues in automatic face recognition: the role of feature extraction and the difficulty due to occlusion.

a) The Role of Feature Extraction: The question of *which low-dimensional features of an object image are the most relevant or informative for classification* is a central issue in face recognition, and in object recognition in general. An enormous volume of literature has been devoted to investigate various data-dependent feature transformations for projecting the high-dimensional test image into lower dimensional feature spaces: examples include Eigenfaces [23], Fisherfaces [24], Laplacianfaces [25], and a host of variants [26], [27]. With so many proposed features and so little consensus about which are better or worse, practitioners lack guidelines to decide which features to use. However, within our proposed framework, the theory of compressed sensing implies that *the precise choice of feature space is no longer critical*: even random features contain enough information to recover the sparse representation and hence correctly classify any test image. What is critical is that the dimension of the feature space is sufficiently large, and that the sparse representation is correctly computed.

b) Robustness to Occlusion: Occlusion poses a significant obstacle to robust, real-world face recognition [16], [28], [29]. This difficulty is mainly due to the unpredictable nature of the error incurred by occlusion: it may affect any part of the image, and may be arbitrarily large in magnitude. Nevertheless, this error typically corrupts only a fraction of the image pixels, and is therefore sparse in the standard basis given by individual pixels. When the error has such a sparse representation, it can be handled uniformly within our framework: the basis in which the error is sparse can be treated as a special class of training samples. The subsequent sparse representation of an occluded test image w.r.t. this expanded dictionary (training images plus error basis) naturally separates the component of the test image arising due to occlusion from the component arising from the identity of the test subject (see Figure 1 for an example). In this context, the theory of

[2] In contrast, methods such as [16], [17] that utilize only a single training sample per class face a more difficult problem and generally incorporate more explicit prior knowledge about the types of variation that could occur in the test sample.

[3] The relationship between our method and NN, NS, and NFL is explored more thoroughly in the supplementary appendix.

sparse representation and compressed sensing characterizes when such *source-and-error separation* can take place, and therefore how much occlusion the resulting recognition algorithm can tolerate.

c) Organization of this Paper: In Section II, we introduce a basic, general framework for classification using sparse representation, applicable to a wide variety of problems in image-based object recognition. We will discuss why the sparse representation can be computed by ℓ^1-minimization, and how it can be used for classifying and validating any given test sample. Section III shows how to apply this general classification framework to study two important issues in image-based face recognition: feature extraction and robustness to occlusion. In Section IV, we verify the proposed method with extensive experiments on popular face datasets, and comparisons with many other state-of-the-art face recognition techniques. Further connections between our method, nearest neighbor, and nearest subspace are discussed in the supplementary appendix.

While the proposed method is of broad interest to object recognition in general, the studies and experimental results in this paper are confined to human frontal face recognition. We will deal with illumination and expressions but we do not explicitly account for object pose, nor rely on any 3-D model of the face. The proposed algorithm is robust to small variations in pose and displacement, for example, due to registration errors. However, we do assume that detection, cropping, and normalization of the face have been performed prior to applying our algorithm.

II. Classification Based on Sparse Representation

A basic problem in object recognition is to use labeled training samples from k distinct object classes to correctly determine the class to which a new test sample belongs. We arrange the given n_i training samples from the i-th class as columns of a matrix $A_i \doteq [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$. In the context of face recognition, we will identify a $w \times h$ grayscale image with the vector $\boldsymbol{v} \in \mathbb{R}^m$ ($m = wh$) given by stacking its columns; the columns of A_i are then the training face images of the i-th subject.

A. Test Sample as a Sparse Linear Combination of Training Samples

An immense variety of statistical, generative or discriminative, models have been proposed for exploiting the structure of the A_i for recognition. One particularly simple and effective approach models the samples from a single class as lying on a linear subspace. Subspace models are flexible enough to capture much of the variation in real datasets, and are especially well-motivated in the context of face recognition, where it has been observed that the images of faces under varying lighting and expression lie on a special low-dimensional subspace [24], [30], often called a *face subspace*. Although the proposed framework and algorithm can also apply to multimodal or nonlinear distributions (see the supplementary appendix for more detail), for ease of presentation, we shall first assume that the training samples from a single class do lie on a subspace. This is the only prior knowledge about the training samples we will be using in our solution.[4]

[4]In face recognition, we actually do not need to know whether the linear structure is due to varying illumination or expression, since we do not rely on domain-specific knowledge such as an illumination model [31] to eliminate the variability in the training and testing images.

Given sufficient training samples of the i-th object class, $A_i = [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,n_i}] \in \mathbb{R}^{m \times n_i}$, any new (test) sample $\boldsymbol{y} \in \mathbb{R}^m$ from the same class will approximately lie in the linear span of the training samples[5] associated with object 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 $\alpha_{i,j} \in \mathbb{R}, j = 1, 2, \ldots, n_i$.

Since the membership i of the test sample is initially unknown, we define a new matrix A for the entire training set as the concatenation of the n training samples of all k object classes:

$$A \doteq [A_1, A_2, \ldots, A_k] = [\boldsymbol{v}_{1,1}, \boldsymbol{v}_{1,2}, \ldots, \boldsymbol{v}_{k,n_k}]. \tag{2}$$

Then the linear representation of $\boldsymbol{y}$ can be rewritten in terms of all training samples 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 zero except those associated with the i-th class.

As the entries of the vector $\boldsymbol{x}_0$ encode the identity of the test sample $\boldsymbol{y}$, it is tempting to attempt to obtain it by solving the linear system of equations $\boldsymbol{y} = A\boldsymbol{x}$. Notice, though, that using the entire training set to solve for $\boldsymbol{x}$ represents a significant departure from one sample or one class at a time methods such as nearest neighbor (NN) and nearest subspace (NS). We will later argue that one can obtain a more discriminative classifier from such a global representation. We will demonstrate its superiority over these local methods (NN or NS) both for identifying objects represented in the training set and for rejecting outlying samples that do not arise from any of the classes present in the training set. These advantages can come without an increase in the order of growth of the computation: as we will see, the complexity remains linear in the size of training set.

Obviously, if $m > n$, the system of equations $\boldsymbol{y} = A\boldsymbol{x}$ is overdetermined and the correct $\boldsymbol{x}_0$ can usually be found as its unique solution. We will see in Section III, however, that in robust face recognition, the system $\boldsymbol{y} = A\boldsymbol{x}$ is typically underdetermined, and so its solution is not unique.[6] Conventionally, this difficulty is resolved by choosing the minimum ℓ^2-norm solution,

$$(\ell^2): \qquad \hat{\boldsymbol{x}}_2 = \arg\min \|\boldsymbol{x}\|_2 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}. \tag{4}$$

While this optimization problem can be easily solved (via the pseudoinverse of A), the solution $\hat{\boldsymbol{x}}_2$ is not especially informative for recognizing the test sample $\boldsymbol{y}$. As shown in Example 1, $\hat{\boldsymbol{x}}_2$ is generally *dense*, with large nonzero entries corresponding to training samples from many different classes. To resolve this difficulty, we instead exploit the following simple observation: A valid test sample $\boldsymbol{y}$ can be sufficiently represented using only the training samples from the same class. This representation is naturally *sparse* if the number of object classes k is reasonably large. For instance, if $k = 20$, only 5% of the entries of the desired $\boldsymbol{x}_0$ should be nonzero. The more sparse the recovered $\boldsymbol{x}_0$ is, the easier will it be to accurately determine the identity of the test

[5]One may refer to [32] for how to choose the training images to ensure this property for face recognition. Here, we assume such a training set is given.

[6]Furthermore, even in the overdetermined case, such a linear equation may not be perfectly satisfied in the presence of data noise (see Section II-B.2).

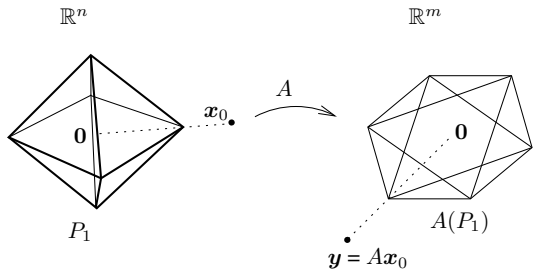


Fig. 2. **Geometry of sparse representation via ℓ^1-minimization.** The ℓ^1-minimization determines which facet (of the lowest-dimension) of the polytope $A(P_\alpha)$ the point $\boldsymbol{y}/\|\boldsymbol{y}\|_1$ lies in. The test sample vector $\boldsymbol{y}$ is represented as a linear combination of just the vertices of that facet, with coefficients $\boldsymbol{x}_0$.

sample $\boldsymbol{y}$.[7]

This motivates us to seek the sparsest solution to $\boldsymbol{y} = A\boldsymbol{x}$, solving the following optimization problem:

$$(\ell^0): \qquad \hat{\boldsymbol{x}}_0 = \arg\min \|\boldsymbol{x}\|_0 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}, \quad (5)$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector. In fact, if the columns of A are in general position, then whenever $\boldsymbol{y} = A\boldsymbol{x}$ for some $\boldsymbol{x}$ with less than $m/2$ nonzeros, $\boldsymbol{x}$ is the unique sparsest solution: $\hat{\boldsymbol{x}}_0 = \boldsymbol{x}$ [33]. However, the problem of finding the sparsest solution of an underdetermined system of linear equations is NP-hard, and difficult even to approximate [13]: That is, in the general case, no known procedure for finding the sparsest solution is significantly more efficient than exhausting all subsets of the entries for $\boldsymbol{x}$.

B. Sparse Solution via ℓ^1-Minimization

Recent development in the emerging theory of *sparse representation and compressed sensing* [9]–[11] reveals that if the solution $\boldsymbol{x}_0$ sought is *sparse enough*, the solution of the ℓ^0-minimization problem (5) is equal to the solution to the following ℓ^1-minimization problem:

$$(\ell^1): \qquad \hat{\boldsymbol{x}}_1 = \arg\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}. \quad (6)$$

This problem can be solved in polynomial time by standard linear programming methods [34]. Even more efficient methods are available when the solution is known to be very sparse. For example, homotopy algorithms recover solutions with t nonzeros in $O(t^3 + n)$ time, linear in the size of the training set [35].

1) Geometric Interpretation: Figure 2 gives a geometric interpretation (essentially due to [36]) of why minimizing the ℓ^1-norm correctly recovers sufficiently sparse solutions. Let P_α denote the ℓ^1-ball (or crosspolytope) of radius α:

$$P_\alpha \doteq \{\boldsymbol{x} : \|\boldsymbol{x}\|_1 \le \alpha\} \quad \subset \mathbb{R}^n. \quad (7)$$

In Figure 2, the unit ℓ^1-ball P_1 is mapped to the polytope $P \doteq A(P_1) \subset \mathbb{R}^m$ consisting of all $\boldsymbol{y}$ that satisfy $\boldsymbol{y} = A\boldsymbol{x}$ for some $\boldsymbol{x}$ whose ℓ^1-norm is ≤ 1.

The geometric relationship between P_α and the polytope $A(P_\alpha)$ is invariant to scaling. That is, if we scale P_α, its image under multiplication by A is also scaled by the same amount. Geometrically, finding the minimum ℓ^1-norm solution $\hat{\boldsymbol{x}}_1$ to (6) is equivalent to expanding the ℓ^1-ball P_α until the polytope $A(P_\alpha)$ first touches $\boldsymbol{y}$. The value of α at which this occurs is exactly $\|\hat{\boldsymbol{x}}_1\|_1$.

Now suppose that $\boldsymbol{y} = A\boldsymbol{x}_0$ for some sparse $\boldsymbol{x}_0$. We wish to know when solving (6) correctly recovers $\boldsymbol{x}_0$. This question is easily resolved from the geometry of Figure 2: Since $\hat{\boldsymbol{x}}_1$ is found by expanding both P_α and $A(P_\alpha)$ until a point of $A(P_\alpha)$ touches $\boldsymbol{y}$, the ℓ^1-minimizer $\hat{\boldsymbol{x}}_1$ must generate a point $A\hat{\boldsymbol{x}}_1$ on the boundary of P.

Thus $\hat{\boldsymbol{x}}_1 = \boldsymbol{x}_0$ if and only if the point $A(\boldsymbol{x}_0/\|\boldsymbol{x}_0\|_1)$ lies on the boundary of the polytope P. For the example shown in Figure 2, it is easy to see that the ℓ^1-minimization recovers all $\boldsymbol{x}_0$ with only one nonzero entry. This equivalence holds because all of the vertices of P_1 map to points on the boundary of P.

In general, if A maps all t-dimensional facets of P_1 to facets of P, the polytope P is referred to as *(centrally) t-neighborly* [36]. From the above, we see that the ℓ^1-minimization (6) correctly recovers all $\boldsymbol{x}_0$ with $\le t+1$ nonzeros iff P is t-neighborly, in which case it is equivalent to the ℓ^0-minimization (5).[8] This condition is surprisingly common: even polytopes P given by random matrices (e.g., uniform, Gaussian, and partial Fourier) are highly neighborly [15], allowing correct recover of sparse $\boldsymbol{x}_0$ 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 m is moderate [37]. When m is large, it is known that with overwhelming probability, the neighborliness of a randomly chosen polytope P is loosely bounded between:

$$c \cdot m < t < \lfloor (m+1)/3 \rfloor \quad (8)$$

for some small constant $c > 0$ (see [9], [36]). Loosely speaking, as long as the number of nonzero entries of $\boldsymbol{x}_0$ is a small fraction of the dimension m, ℓ^1-minimization will recover $\boldsymbol{x}_0$.

2) Dealing with Small, Dense Noise: So far, we have assumed that equation (3) holds exactly. Since real data are noisy, it may not be possible to express the test sample exactly as a sparse superposition of the training samples. The model (3) can be modified to explicitly account for small, possibly dense noise, by writing

$$\boldsymbol{y} = A\boldsymbol{x}_0 + \boldsymbol{z}, \quad (9)$$

where $\boldsymbol{z} \in \mathbb{R}^m$ is a noise term with bounded energy $\|\boldsymbol{z}\|_2 < \varepsilon$. The sparse solution $\boldsymbol{x}_0$ can still be approximately recovered by solving the following *stable* ℓ^1-minimization problem:

$$(\ell^1_s): \quad \hat{\boldsymbol{x}}_1 = \arg\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|A\boldsymbol{x} - \boldsymbol{y}\|_2 \le \varepsilon. \quad (10)$$

This convex optimization problem can be efficiently solved via second-order cone programming [34] (see Section IV for our algorithm of choice). The solution of (ℓ^1_s) is guaranteed to approximately recovery sparse solutions in ensembles of random matrices A [38]: There are constants ρ and ζ such that with overwhelming probability, if $\|\boldsymbol{x}_0\|_0 < \rho m$ and $\|\boldsymbol{z}\|_2 \le \varepsilon$, then the computed $\hat{\boldsymbol{x}}_1$ satisfies

$$\|\hat{\boldsymbol{x}}_1 - \boldsymbol{x}_0\|_2 \le \zeta\varepsilon. \quad (11)$$

[7]This intuition holds only when the size of the database is fixed. For example, if we are allowed to append additional irrelevant columns to A, we can make the solution $\boldsymbol{x}_0$ have a smaller fraction of nonzeros, but this does not make $\boldsymbol{x}_0$ more informative for recognition.

[8]Thus, neighborliness gives a necessary and sufficient condition for sparse recovery. The restricted isometry properties often used in analyzing the performance of ℓ^1-minimization in random matrix ensembles (e.g., [15]) give sufficient, but *not* necessary, conditions.

C. Classification Based on Sparse Representation

Given a new test sample $\boldsymbol{y}$ from one of the classes in the training set, we first compute its sparse representation $\hat{\boldsymbol{x}}_1$ via (6) or (10). Ideally, the nonzero entries in the estimate $\hat{\boldsymbol{x}}_1$ will all be associated with the columns of A from a single object class i, and we can easily assign the test sample $\boldsymbol{y}$ to that class. However, noise and modeling error may lead to small nonzero entries associated with multiple object classes (see Figure 3). Based on the global, sparse representation, one can design many possible classifiers to resolve this. For instance, we can simply assign $\boldsymbol{y}$ to the object class with the single largest entry in $\hat{\boldsymbol{x}}_1$. However, such heuristics do not harness the subspace structure associated with images in face recognition. To better harness such linear structure, we instead classify $\boldsymbol{y}$ based on how well the coefficients associated with all training samples of each object reproduce $\boldsymbol{y}$.

For each class i, let $\delta_i : \mathbb{R}^n \to \mathbb{R}^n$ be the characteristic function which selects the coefficients associated with the i-th class. 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 class i. Using only the coefficients associated with the i-th class, one can approximate the given test sample $\boldsymbol{y}$ as $\hat{\boldsymbol{y}}_i = A\delta_i(\hat{\boldsymbol{x}}_1)$. We then classify $\boldsymbol{y}$ based on these approximations by assigning it to the object class that minimizes the residual between $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}_i$:

$$\min_i \; r_i(\boldsymbol{y}) \doteq \|\boldsymbol{y} - A\,\delta_i(\hat{\boldsymbol{x}}_1)\|_2. \tag{12}$$

Algorithm 1 below summarizes the complete recognition procedure. Our implementation minimizes the ℓ^1-norm via a primal-dual algorithm for linear programming based on [39], [40].

Algorithm 1 : Sparse Representation-based Classification (SRC)

1: **Input:** a matrix of training samples $A = [A_1, A_2, \ldots, A_k] \in \mathbb{R}^{m\times n}$ for k classes, a test sample $\boldsymbol{y} \in \mathbb{R}^m$, (and an optional error tolerance $\varepsilon > 0$.)
2: Normalize the columns of A to have unit ℓ^2-norm.
3: Solve the ℓ^1-minimization problem:

$$\hat{\boldsymbol{x}}_1 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad A\boldsymbol{x} = \boldsymbol{y}. \tag{13}$$

(Or alternatively, solve

$$\hat{\boldsymbol{x}}_1 = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ subject to } \|A\boldsymbol{x} - \boldsymbol{y}\|_2 \le \varepsilon.)$$

4: Compute the residuals $r_i(\boldsymbol{y}) = \|\boldsymbol{y} - A\,\delta_i(\hat{\boldsymbol{x}}_1)\|_2$ for $i = 1, \ldots, k$.
5: **Output:** 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 subsample the images from the original 192×168 to size 12×10. The pixel values of the downsampled image are used as 120-D features – stacked as columns of the matrix A in the algorithm. Hence matrix A has size 120×1207, and the system $\boldsymbol{y} = A\boldsymbol{x}$ is underdetermined. Figure 3 left illustrates the sparse coefficients recovered by Algorithm 1 for a test image from the first subject. The figure also shows the features and the original images that correspond to the two largest coefficients. The two largest coefficients are both associated with training samples from subject 1. Figure 3 right shows the residuals w.r.t. the 38 projected coefficients $\delta_i(\hat{\boldsymbol{x}}_1)$, $i = 1, 2, \ldots, 38$. With 12×10 downsampled images as features, Algorithm 1 achieves an overall recognition rate of 92.1% across the Extended Yale B database. (See Section IV for details and performance with other features such as Eigenfaces and Fisherfaces, as well as comparison with other methods.) Whereas the more conventional minimum ℓ^2-norm solution to the underdetermined system $\boldsymbol{y} = A\boldsymbol{x}$ is typically quite dense, minimizing the ℓ^1-norm favors sparse solutions, and provably recovers the sparsest solution when this solution is sufficiently sparse. To illustrate this contrast, Figure 4 left shows the coefficients of the same test image given by the conventional ℓ^2-minimization (4), and Figure 4 right shows the corresponding residuals w.r.t. the 38 subjects. The coefficients are much less sparse than those given by ℓ^1-minimization (in Figure 3), and the dominant coefficients are not associated with subject 1. As a result, the smallest residual in Figure 4 does not correspond to the correct subject (subject 1).

D. Validation Based on Sparse Representation

Before classifying a given test sample, we must first decide if it is a valid sample from one of the classes in the dataset. The ability to detect and then reject invalid test samples, or "outliers," is crucial for recognition systems to work in real-world situations. A face recognition system, for example, could be given a face image of a subject that is not in the database, or an image that is not a face at all.

Systems based on conventional classifiers such as nearest neighbor (NN) or nearest subspace (NS), often use the residuals $r_i(\boldsymbol{y})$ for validation, in addition to identification. That is, the algorithm accepts or rejects a test sample based on how small the smallest residual is. However, each residual $r_i(\boldsymbol{y})$ is computed without any knowledge of images of other object classes in the training dataset and only measures similarity between the test sample and each individual class.

In the sparse representation paradigm, the coefficients $\hat{\boldsymbol{x}}_1$ are computed globally, in terms of images of all classes. In a sense, it can harness the joint distribution of all classes for validation. We contend that the coefficients $\hat{\boldsymbol{x}}$ are better statistics for validation than the residuals. Let us first see this through an example.

Example 2 (Concentration of Sparse Coefficients): We randomly select an irrelevant image from Google, and downsample it to 12×10. We then compute the sparse representation of the image against the same Extended Yale B training data as in Example 1. Figure 5 left plots the obtained coefficients, and right plots the corresponding residuals. Compared to the coefficients of a valid test image in Figure 3, notice that the coefficients $\hat{\boldsymbol{x}}$ 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 $\hat{\boldsymbol{x}}$ 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 class in the dataset:

Definition 1 (Sparsity Concentration Index): The *sparsity concentration index* (SCI) of a coefficient vector $\boldsymbol{x} \in \mathbb{R}^n$ is

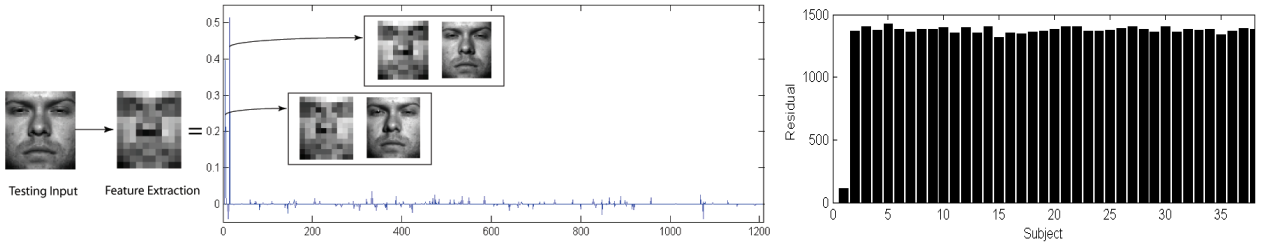


Fig. 3. **A valid test image.** Left: Recognition with 12×10 downsampled images as features. The test image $\boldsymbol{y}$ belongs to subject 1. The values of the sparse coefficients recovered from Algorithm 1 are plotted on the right together with the two training examples that correspond to the two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

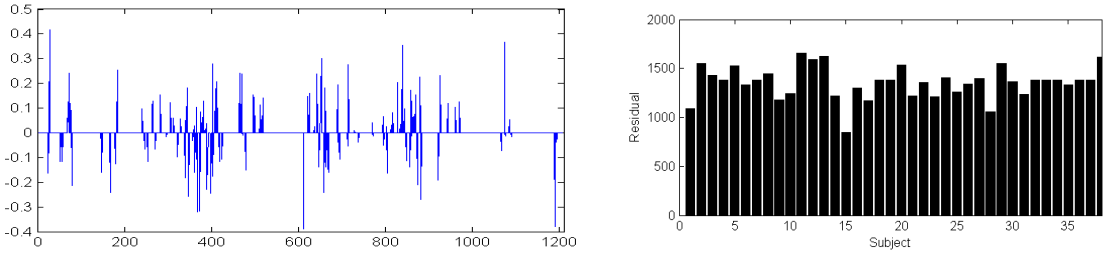


Fig. 4. **Non-sparsity of the ℓ^2-minimizer.** Left: Coefficients from ℓ^2-minimization, using the same test image as Figure 3. The recovered solution is not sparse and hence less informative for recognition (large coefficients do not correspond to training images of this test subject). Right: The residuals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

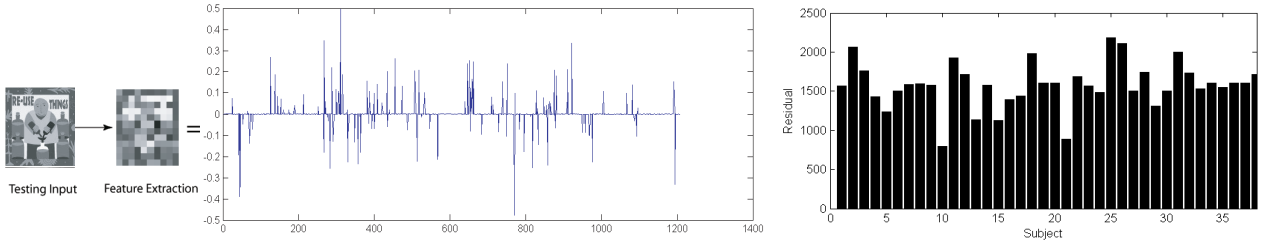


Fig. 5. **Example of an invalid test image.** Left: Sparse coefficients for the invalid test image w.r.t. the same training data set from Example 1. The test image is a randomly selected irrelevant image. Right: The residuals of the invalid test image w.r.t. the projection $\delta_i(\hat{\boldsymbol{x}})$ of the sparse representation computed by ℓ^1-minimization. The ratio of the two smallest residuals is about 1:1.2.

defined as

$$\text{SCI}(\boldsymbol{x}) \doteq \frac{k \cdot \max_i \|\delta_i(\boldsymbol{x})\|_1 / \|\boldsymbol{x}\|_1 - 1}{k-1} \quad \in [0,1]. \tag{14}$$

For a solution $\hat{\boldsymbol{x}}$ found by Algorithm 1, if $\text{SCI}(\hat{\boldsymbol{x}}) = 1$, the test image is represented using only images from a single object, and if $\text{SCI}(\hat{\boldsymbol{x}}) = 0$, the sparse coefficients are spread evenly over all classes.[9] We choose a threshold $\tau \in (0,1)$ and accept a test image as valid if

$$\text{SCI}(\hat{\boldsymbol{x}}) \geq \tau, \tag{15}$$

and otherwise reject as invalid. In step 5 of Algorithm 1, one may choose to output the identity of $\boldsymbol{y}$ only if it passes this criterion.

Unlike NN or NS, this new rule avoids the use of the residuals $r_i(\boldsymbol{y})$ for validation. Notice that in Figure 5, even for a non-face image, with a large training set, the smallest residual of the invalid test image is not so large. 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.[10] 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.

One benefit to this approach to validation is improved performance against generic objects that are similar to multiple object classes. For example, in face recognition, 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 IV-G we will demonstrate that the new validation rule outperforms the

[9]Directly choosing $\boldsymbol{x}$ to minimize the SCI might produce more concentrated coefficients; however, the SCI is highly non-convex and difficult to optimize. For valid test images, minimizing the ℓ^1-norm already produces representations that are well-concentrated on the correct subject class.

[10]We find empirically that this separation works well enough in our experiments with face images. However, it is possible that better validation and identification rules can be contrived from using the residual and the sparsity together.

NN and NS methods, with as much as 10–20% improvement in verification rate for a given false accept rate (see Figure 14 in Section IV or Figure 18 in the supplementary appendix).

III. Two Fundamental Issues in Face Recognition

In this section, we study the implications of the above general classification framework for two critical issues in face recognition: 1. The choice of feature transformation, and 2. Robustness to corruption, occlusion, and disguise.

A. The Role of Feature Extraction

In the computer vision literature, numerous feature extraction schemes have been investigated for finding projections that better separate the classes in lower-dimensional spaces, which are often referred to as *feature spaces*. One class of methods extracts holistic face features, such as Eigenfaces [23], Fisherfaces [24], and Laplacianfaces [25]. Another class of methods tries to extract meaningful partial facial features (e.g., patches around eyes or nose) [21], [41]. See Figure 6 for some examples. Traditionally, when feature extraction is used in conjunction with simple classifiers such as NN and NS, the choice of feature transformation is considered critical to the success of the algorithm. This has led to the development of a wide variety of increasingly complex feature extraction methods, including nonlinear and kernel features [42], [43]. In this section, we reexamine the role of feature extraction within the new sparse representation framework for face recognition.

One benefit of feature extraction, which carries over to the proposed sparse representation framework, is reduced data dimension and computational cost. For raw face images, the corresponding linear system $\boldsymbol{y} = A\boldsymbol{x}$ is very large. For instance, if the face images are given at the typical resolution, 640×480 pixels, the dimension m is on the order of 10^5. Although Algorithm 1 relies on scalable methods such as linear programming, directly applying it to such high-resolution images is still beyond the capability of regular computers.

Since most feature transformations involve only linear operations (or approximately so), 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:

$$\tilde{\boldsymbol{y}} \doteq R\boldsymbol{y} = RA\boldsymbol{x}_0 \quad \in \mathbb{R}^d. \tag{16}$$

In practice, the dimension d of the feature space is typically chosen to be much smaller than n. In this case, the system of equations $\tilde{\boldsymbol{y}} = RA\boldsymbol{x} \in \mathbb{R}^d$ is underdetermined in the unknown $\boldsymbol{x} \in \mathbb{R}^n$. Nevertheless, as the desired solution $\boldsymbol{x}_0$ is sparse, we can hope to recover it by solving the following *reduced* ℓ^1-minimization problem:

$$(\ell^1_r): \ \hat{\boldsymbol{x}}_1 = \arg\min \|\boldsymbol{x}\|_1 \quad \text{subject to} \quad \|RA\boldsymbol{x} - \tilde{\boldsymbol{y}}\|_2 \le \varepsilon, \tag{17}$$

for a given error tolerance $\varepsilon > 0$. Thus, in Algorithm 1, the matrix A of training images is now replaced by the matrix $RA \in \mathbb{R}^{d \times n}$ of d-dimensional features; the test image $\boldsymbol{y}$ is replaced by its features $\tilde{\boldsymbol{y}}$.

For extant face recognition methods, empirical studies have show that increasing the dimension d of the feature space generally improves the recognition rate, as long as the distribution of features RA_i does not become degenerate [42]. Degeneracy is not an issue for ℓ^1-minimization, since it merely requires that $\tilde{\boldsymbol{y}}$ be in or near the range of RA_i – it does not depend on the covariance $\Sigma_i = A_i^T R^T R A_i$ being nonsingular as in classical discriminant analysis. The stable version of ℓ^1-minimization (10) or (17) is known in statistical literature as the Lasso [14].[11] It effectively regularizes highly underdetermined linear regression when the desired solution is sparse, and has also been proven consistent in some noisy, overdetermined settings [12].

For our sparse representation approach to recognition, we would like to understand how the choice of the feature extraction R affects the ability of the ℓ^1-minimization (17) to recover the correct sparse solution $\boldsymbol{x}_0$. From the geometric interpretation of ℓ^1-minimization given in Section II-B.1, the answer to this depends on whether the associated new polytope $P = RA(P_1)$ remains sufficiently neighborly. It is easy to show that the neighborliness of the polytope $P = RA(P_1)$ increases with d [11], [15]. In Section IV, our experimental results will verify the ability of ℓ^1-minimization, in particular the stable version (17), to recover sparse representations for face recognition using a variety of features. This suggests that most data-dependent features popular in face recognition (e.g., Eigenfaces, Laplacianfaces) may indeed give highly neighborly polytopes P.

Further analysis of high-dimensional polytope geometry has revealed a somewhat surprising phenomenon: if the solution $\boldsymbol{x}_0$ is sparse enough, then with overwhelming probability, it can be correctly recovered via ℓ^1-minimization from *any* sufficiently large number d of linear measurements $\tilde{\boldsymbol{y}} = RA\boldsymbol{x}_0$. More precisely, if $\boldsymbol{x}_0$ has $t \ll n$ nonzeros, then with overwhelming probability,

$$d \ge 2t \log(n/d) \tag{18}$$

random linear measurements are sufficient for ℓ^1-minimization (17) to recover the correct sparse solution $\boldsymbol{x}_0$ [44].[12] This surprising phenomenon has been dubbed the "blessing of dimensionality" [15], [46]. Random features can be viewed as a less-structured counterpart to classical face features, such as Eigenfaces or Fisherfaces. Accordingly, we call the linear projection generated by a Gaussian random matrix *Randomfaces*:[13]

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. The row vectors of R can be viewed as d random faces in $\mathbb{R}^m$.

One major advantage of Randomfaces is that they are extremely efficient to generate, as the transformation R is independent of the training dataset. This advantage can be important for a face recognition system where we may not be able to acquire a complete database of all subjects of interest to precompute data-

[11] Classically, the Lasso solution is defined as the minimizer of $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$. Here, λ can be viewed as inverse of the Lagrange multiplier associated with a constraint $\|\boldsymbol{y} - A\boldsymbol{x}\|_2^2 \le \varepsilon$. For every λ there is an ε such that the two problems have the same solution. However, ε can be interpreted as a pixel noise level, and fixed across various instances of the problem, whereas λ cannot. One should distinguish the Lasso *optimization problem* from the LARS *algorithm*, which provably solves some instances of Lasso with very sparse optimizers [35].

[12] Strictly speaking, this threshold holds when random measurements are computed directly from $\boldsymbol{x}_0$, i.e., $\tilde{\boldsymbol{y}} = R\boldsymbol{x}_0$. Nevertheless, our experiments roughly agree with the bound given by (18). The case where $\boldsymbol{x}_0$ is instead sparse in some overcomplete basis A, and we observe random measurements $\tilde{\boldsymbol{y}} = RA\boldsymbol{x}_0$ has also been studied in [45]. While conditions for correct recovery have been given, the bounds are not yet as sharp as (18) above.

[13] Random projection has been previously studied as a general dimensionality-reduction method for numerous clustering problems [47]–[49], as well as for learning nonlinear manifolds [50], [51].

(a) original $\boldsymbol{y}$ (b) 120-D features $\tilde{\boldsymbol{y}} = R\boldsymbol{y}$ (c) eye feature $\tilde{\boldsymbol{y}}$

Fig. 6. **Examples of feature extraction.** (a). Original face image. (b). 120-D representations in terms of four different features (from left to right): Eigenfaces, Laplacianfaces, downsampled (12×10 pixel) image, and random projection. We will demonstrate that all these features contain almost the same information about the identity of the subject and give similarly good recognition performance. (c). The eye is a popular choice of feature for face recognition. In this case, the feature matrix R is simply a binary mask.

dependent transformations such as Eigenfaces, or the subjects in the database may change over time. In such cases, there is no need for recomputing the random transformation R.

As long as the correct sparse solution $\boldsymbol{x}_0$ can be recovered, Algorithm 1 will always give the same classification result, regardless of the feature actually used. Thus, when the dimension of feature d exceeds the above bound (18), one should expect that the recognition performance of Algorithm 1 with different features quickly converges, and the choice of an "optimal" feature transformation is no longer critical: Even random projections or downsampled images should perform as well as any other carefully engineered features. This will be corroborated by the experimental results in Section IV.

B. Robustness to Occlusion or Corruption

In many practical face recognition scenarios, the test image $\boldsymbol{y}$ could be partially corrupted or occluded. In this case, the above linear model (3) should be modified as

$$\boldsymbol{y} \;=\; \boldsymbol{y}_0 + \boldsymbol{e}_0 \;=\; A\,\boldsymbol{x}_0 + \boldsymbol{e}_0, \tag{19}$$

where $\boldsymbol{e}_0 \in \mathbb{R}^m$ is a vector of errors – a fraction, ρ, of its entries are nonzero. The nonzero entries of $\boldsymbol{e}_0$ model which pixels in $\boldsymbol{y}$ are corrupted or occluded. The locations of corruption can differ for different test images and are not known to the computer. The errors may have arbitrary magnitude and therefore cannot be ignored or treated with techniques designed for small noise such as the one given in Section II-B.2.

A fundamental principle of coding theory [52] is that *redundancy* in the measurement is essential to detecting and correcting gross errors. Redundancy arises in object recognition because the number of image pixels is typically far greater than the number of subjects that have generated the images. In this case, even if a fraction of the pixels are completely corrupted by occlusion, recognition may still be possible based on the remaining pixels. On the other hand, feature extraction schemes discussed in the previous section would discard useful information that could help compensate for the occlusion. In this sense, no representation is more redundant, robust, or informative than the original images. Thus, when dealing with occlusion and corruption, we should always work with the highest possible resolution, performing downsampling or feature extraction only if the resolution of the original images is too high to process.

Of course, redundancy would be of no use without efficient computational tools for exploiting the information encoded in the redundant data. The difficulty in directly harnessing the redundancy in corrupted raw images has led researchers to instead focus on *spatial locality* as a guiding principle for robust recognition. Local features computed from only a small fraction of the image pixels are clearly less likely to be corrupted by occlusion than holistic features. In face recognition, methods such as ICA [53] and LNMF [54] exploit this observation by adaptively choosing filter bases that are locally concentrated. Local Binary Patterns [55] and Gabor wavelets [56] exhibit similar properties, since they are also computed from local image regions. A related approach partitions the image into fixed regions and computes features for each region [16], [57]. Notice, though, that projecting onto locally concentrated bases transforms the domain of the occlusion problem, rather than eliminating the occlusion. Errors on the original pixels become errors in the transformed domain, and may even become less local. The role of feature extraction in achieving spatial locality is therefore questionable, since *no bases or features are more spatially localized than the original image pixels themselves.* In fact, the most popular approach to robustifying feature-based methods is based on randomly sampling individual pixels [28], sometimes in conjunction with statistical techniques such as multivariate trimming [29].

Now, let us show how the proposed sparse representation classification framework can be extended to deal with occlusion. Let us assume that the corrupted pixels are a relatively small portion of the image. The error vector $\boldsymbol{e}_0$, like the vector $\boldsymbol{x}_0$, then has sparse[14] nonzero entries. Since $\boldsymbol{y}_0 = A\boldsymbol{x}_0$, we can rewrite (19) as

$$\boldsymbol{y} = \begin{bmatrix} A, & I \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_0 \\ \boldsymbol{e}_0 \end{bmatrix} \doteq B\,\boldsymbol{w}_0. \tag{20}$$

Here, $B = [A,\ I] \in \mathbb{R}^{m \times (n+m)}$, so the system $\boldsymbol{y} = B\boldsymbol{w}$ is always underdetermined and does not have a unique solution for $\boldsymbol{w}$. However, from the above discussion about the sparsity of $\boldsymbol{x}_0$ and $\boldsymbol{e}_0$, the correct generating $\boldsymbol{w}_0 = [\boldsymbol{x}_0,\ \boldsymbol{e}_0]$ has at most $n_i + \rho m$ nonzeros. We might therefore hope to recover $\boldsymbol{w}_0$ as the sparsest solution to the system $\boldsymbol{y} = B\boldsymbol{w}$. In fact, if the matrix B is in general position, then as long as $\boldsymbol{y} = B\tilde{\boldsymbol{w}}$ for some $\tilde{\boldsymbol{w}}$ with less than $m/2$ nonzeros, $\tilde{\boldsymbol{w}}$ is the unique sparsest solution. Thus, if the occlusion $\boldsymbol{e}$ covers less than $\frac{m-n_i}{2}$ pixels, $\approx 50\%$ of the image, the sparsest solution $\tilde{\boldsymbol{w}}$ to $\boldsymbol{y} = B\boldsymbol{w}$ is the true generator, $\boldsymbol{w}_0 = [\boldsymbol{x}_0,\ \boldsymbol{e}_0]$.

More generally, one can assume that the corrupting error $\boldsymbol{e}_0$ has a sparse representation with respect to some basis $A_e \in \mathbb{R}^{m \times n_e}$. That is, $\boldsymbol{e}_0 = A_e\boldsymbol{u}_0$ for some sparse vector $\boldsymbol{u}_0 \in \mathbb{R}^m$. Here, we have chosen the special case $A_e = I \in \mathbb{R}^{m \times m}$ as $\boldsymbol{e}_0$ is assumed to be sparse with respect to the natural pixel coordinates. If the

[14]Here, "sparse" does not mean "very few." In fact, as our experiments will demonstrate, the portion of corrupted entries can be rather significant. Depending on the type of corruption, our method can handle up to $\rho = 40\%$ or $\rho = 70\%$ corrupted pixels.

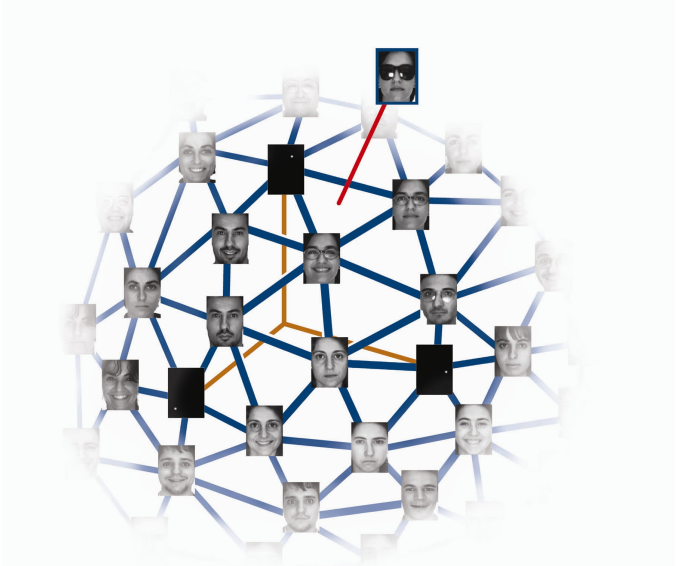


Fig. 7. **Face recognition with occlusion.** The columns of $\pm B = \pm[A, I]$ span a high-dimensional polytope $P = B(P_1)$ in $\mathbb{R}^m$. Each vertex of this polytope is either a training image or an image with just a single pixel illuminated (corresponding to the identity submatrix I). Given a test image, solving the ℓ^1-minimization problem essentially locates which facet of the polytope the test image falls on. The ℓ^1-minimization finds the facet with the fewest possible vertices. Only vertices of that facet contribute to the representation; all other vertices have no contribution.

error $\boldsymbol{e}_0$ is instead more sparse with respect to another basis, e.g., Fourier or Haar, we can simply redefine the matrix B by appending A_e (instead of the identity I) to A and instead seek the sparsest solution $\boldsymbol{w}_0$ to the equation:

$$\boldsymbol{y} = B\boldsymbol{w} \quad \text{with} \quad B = [A,\ A_e] \quad \in \mathbb{R}^{m\times(n+n_e)}. \tag{21}$$

In this way, the same formulation can handle more general classes of (sparse) corruption.

As before, we attempt to recover the sparsest solution $\boldsymbol{w}_0$ from solving the following *extended* ℓ^1-minimization problem:

$$(\ell_e^1): \qquad \hat{\boldsymbol{w}}_1 = \arg\min \|\boldsymbol{w}\|_1 \quad \text{subject to} \quad B\boldsymbol{w} = \boldsymbol{y}. \tag{22}$$

That is, in Algorithm 1, we now replace the image matrix A with the extended matrix $B = [A,\ I]$ and $\boldsymbol{x}$ with $\boldsymbol{w} = [\boldsymbol{x}, \boldsymbol{e}]$.

Clearly, whether the sparse solution $\boldsymbol{w}_0$ can be recovered from the above ℓ^1-minimization depends on the neighborliness of the new polytope $P = B(P_1) = [A,\ I](P_1)$. This polytope contains vertices from both the training images A and the identify matrix I, as illustrated in Figure 7. The bounds given in (8) imply that if $\boldsymbol{y}$ is an image of subject i, the ℓ^1-minimization (22) *cannot* guarantee to correctly recover $\boldsymbol{w}_0 = [\boldsymbol{x}_0,\ \boldsymbol{e}_0]$ if

$$n_i + |\text{support}(\boldsymbol{e}_0)| > d/3.$$

Generally, $d \gg n_i$, so (8) implies that the largest fraction of occlusion under which we can hope to still achieve perfect reconstruction is 33%. This bound is corroborated by our experimental results; see Figure 12.

To know exactly how much occlusion can be tolerated, we need more accurate information about the neighborliness of the polytope P than a loose upper bound given by (8). For instance, we would like to know for a given set of training images, what is the largest amount of (worst-possible) occlusion it can handle. While the best known algorithms for exactly computing the neighborliness of a polytope are combinatorial in nature, tighter upper bounds can be obtained by restricting the search for intersections between the nullspace of B and the ℓ^1-ball to a random subset of the t-faces of the ℓ^1-ball (see [37] for details). We will use this technique to estimate the neighborliness of all the training datasets considered in our experiments.

Empirically, we found that the stable version (10) is only necessary when we do not consider occlusion or corruption $\boldsymbol{e}_0$ in the model (such as the case with feature extraction discussed in the previous section). When we explicitly account for gross errors by using $B = [A,\ I]$ the extended ℓ^1-minimization (22) with the exact constraint $B\boldsymbol{w} = \boldsymbol{y}$ is already stable under moderate noise.

Once the sparse solution $\hat{\boldsymbol{w}}_1 = [\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{e}}_1]$ is computed, setting $\boldsymbol{y}_r \doteq \boldsymbol{y} - \hat{\boldsymbol{e}}_1$ recovers a clean image of the subject with occlusion or corruption compensated for. To identify the subject, we slightly modify the residual $r_i(\boldsymbol{y})$ in Algorithm 1, computing it against the recovered image $\boldsymbol{y}_r$:

$$r_i(\boldsymbol{y}) = \|\boldsymbol{y}_r - A\delta_i(\hat{\boldsymbol{x}}_1)\|_2 = \|\boldsymbol{y} - \hat{\boldsymbol{e}}_1 - A\delta_i(\hat{\boldsymbol{x}}_1)\|_2. \tag{23}$$

IV. Experimental Verification

In this section, we present experiments on publicly available databases for face recognition, which serve both to demonstrate the efficacy of the proposed classification algorithm, and to validate the claims of the previous sections. We will first examine the role of feature extraction within our framework, comparing performance across various feature spaces and feature dimensions, and comparing to several popular classifiers. We will then demonstrate the robustness of the proposed algorithm to corruption and occlusion. Finally, we demonstrate (using ROC curves) the effectiveness of sparsity as a means of validating test images, and examine how to choose training sets to maximize robustness to occlusion.

A. Feature Extraction and Classification Methods

We test our sparse representation-based classification (SRC) algorithm using several conventional holistic face features, namely, Eigenfaces, Laplacianfaces, and Fisherfaces, and compare their performance with two unconventional features: Randomfaces and downsampled images. We compare our algorithm with three classical algorithms, namely, *nearest neighbor* (NN), and *nearest subspace* (NS), discussed in the previous section, as well as linear *support vector machine* (SVM).[15] In this section, we use the stable version of SRC in various lower-dimensional feature spaces, solving the reduced optimization problem (17) with the error tolerance $\varepsilon = 0.05$. The MATLAB implementation of the reduced (feature space) version of Algorithm 1 takes only a few seconds per test image on a typical 3GHz PC.

1) Extended Yale B Database: The Extended Yale B database consists of 2,414 frontal-face images of 38 individuals [58]. The cropped and normalized 192×168 face images were captured under various laboratory-controlled lighting conditions [59]. 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.

[15]Due to the subspace structure of face images, linear SVM is already appropriate for separating features from different faces. The use of a linear kernel (as opposed to more complicated, nonlinear transformations) also makes it possible to directly compare between different algorithms working in the same feature space. Nevertheless, better performance might be achieved by using nonlinear kernels in addition to feature transformations.

Randomly choosing the training set ensures that our results and conclusions will not depend on any special choice of the training data.

We compute the recognition rates with the feature space dimensions 30, 56, 120, and 504. Those numbers correspond to downsampling ratios of 1/32, 1/24, 1/16, and 1/8, respectively.[16] Notice that Fisherfaces are different from the other features because the maximal number of valid Fisherfaces is one less than the number of classes k [24], 38 in this 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.[17] Figure 8 shows the recognition performance for the various features, in conjunction with four different classifiers: SRC, NN, NS, and SVM.

SRC achieves recognition rates between 92.1% and 95.6% for all 120 dimensional feature spaces, and a maximum rate of 98.1% with 504 dimensional randomfaces.[18] The maximum recognition rates for NN, NS, and SVM are 90.7%, 94.1%, and 97.7%, respectively. Tables with all the recognition rates are available in the supplementary appendix. The recognition rates shown in Figure 8 are consistent with those that have been reported in the literature, although some reported on different databases or with different training subsets. For example, He et. al. [25] 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 [32], 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 [60]. 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: the seven images from Session 1 for training, and the other seven from Session 2 for testing. The images are cropped with dimension 165×120 and converted to grayscale. We selected four feature space dimensions: 30, 54, 130, and 540, which correspond to the downsample ratios 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. 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.

Figure 9 shows the recognition rates for this experiment. With 540 dimensional features, SRC achieves a recognition rate between 92.0% and 94.7%. One the other hand, the best rates achieved by NN and NS are 89.7% and 90.3%, respectively. SVM slightly outperforms SRC on this dataset, achieving a maximum recognition rate of 95.7%. However, the performance of SVM varies more with the choice of feature space – the recognition rate using random features is just 88.8%. The supplementary appendix contains a table of detailed numerical results.

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

1) For both the Yale database and AR database, the best performances of SRC and SVM consistently exceed the best performances of the two classical methods NN and NS at each individual feature dimension. More specifically, the best recognition rate for SRC on the Yale database is 98.1%, compared to 97.7% for SVM, 94.0% for NS, and 90.7% for NN; the best rate for SRC on the AR database is 94.7%, compared to 95.7% for SVM, 90.3% for NS, and 89.7% for NN.
2) The performances of the other three classifiers depends strongly on a good choice of "optimal" features – Fisherfaces for lower feature space dimension and Laplacianfaces for higher feature space dimension. With NN and SVM, the performance of the various features does not converge as the dimension of the feature space increases.
3) The results corroborate the theory of compressed sensing: Equation (18) suggests that $d \approx 128$ random linear measurements should suffice for sparse recovery in the Yale database, while $d \approx 88$ random linear measurements should suffice for sparse recovery in the AR database [44]. Beyond these dimensions, the performances of various features in conjunction with ℓ^1-minimization converge, with conventional and unconventional features (e.g., Randomfaces and downsampled images) performing similarly. When the feature dimension is large, a single random projection performs the best (98.1% recognition rate on Yale, 94.7% on AR).

B. Partial Face Features

There have been extensive studies in both the human and computer vision literature about the effectiveness of partial features in recovering the identity of a human face, e.g., see [21], [41]. As a second set of experiments, we test our algorithm on the following three partial facial features: nose, right eye, and mouth & chin. We use the Extended Yale B database for the experiment, with the same training and test sets as in subsection IV-A.1. See Figure 10 for a typical example of the extracted features.

For each of the three features, the dimension d is larger than the number of training samples ($n = 1,207$), and the linear system (16) to be solved becomes overdetermined. Nevertheless, sparse approximate solutions $\boldsymbol{x}$ can still be obtain by solving the ε-relaxed ℓ^1-minimization problem (17) (here, again, $\varepsilon = 0.05$). The results in Figure 10 right again show that the proposed SRC algorithm achieves better recognition rates than NN, NS, and SVM. These experiments also show the scalability of the proposed algorithm in working with more than 10^4-dimensional features.

[16] We cut off the dimension at 504 as the computation 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.

[17] Subspace dimensions significantly greater or less than 9 eventually led to a decrease in performance.

[18] We also experimented with replacing the constrained ℓ^1-minimization in the SRC algorithm with the Lasso. For appropriate choice of regularization λ, the results are similar. For example, with downsampled faces as features and $\lambda = 1,000$, the recognition rates are $73.7\%, 86.2\%, 91.9\%, 97.5\%$, at dimensions $30, 56, 120, 504$, (within 1% of the results in Figure 8).

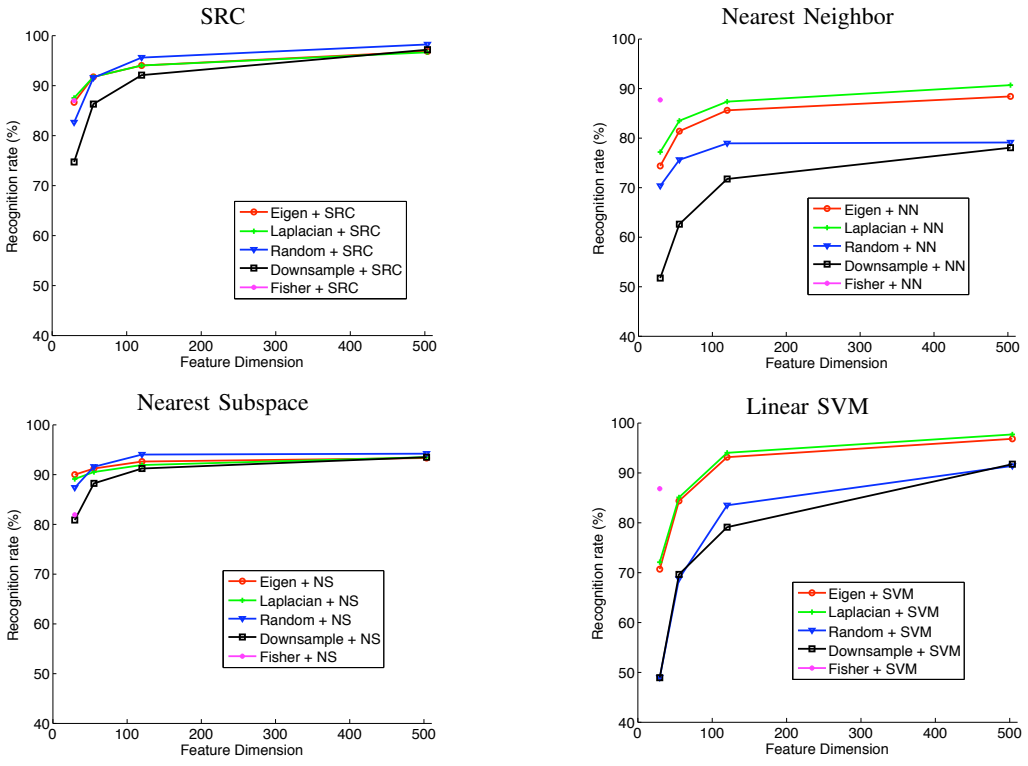


Fig. 8. **Recognition rates on Extended Yale B database**, for various feature transformations and classifiers. Top left: SRC (our approach). Top right: nearest neighbor. Bottom left: nearest subspace. Bottom right: support vector machine (linear kernel).

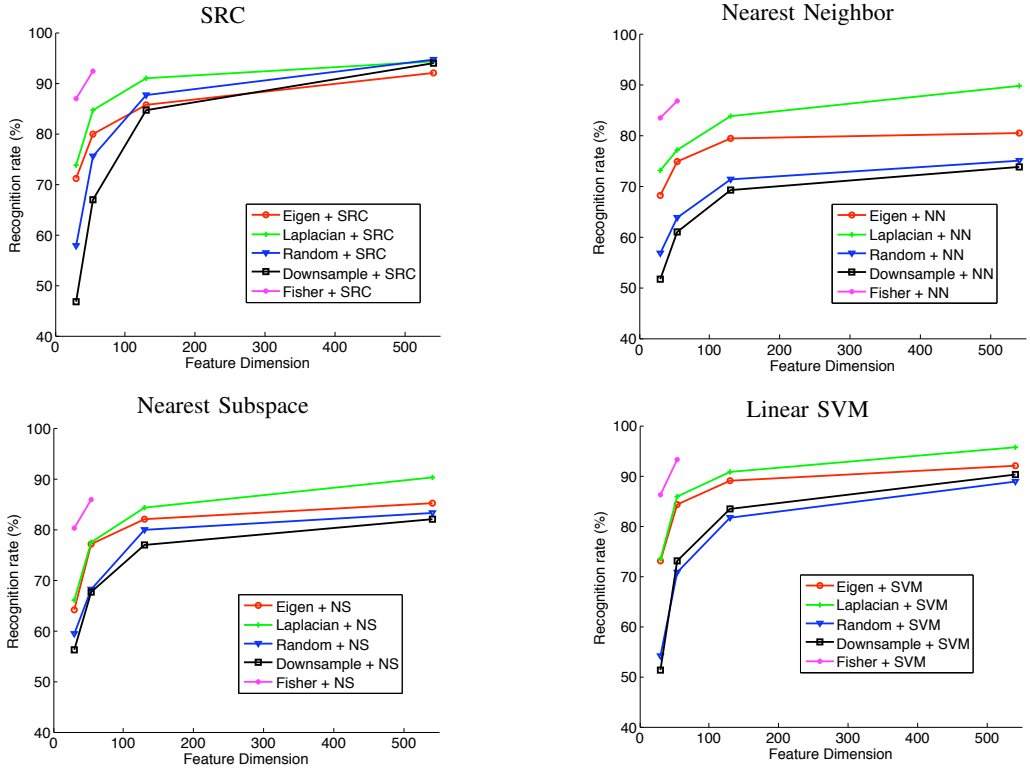
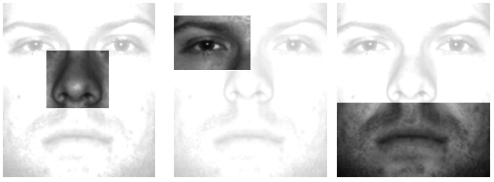


Fig. 9. **Recognition rates on AR database**, for various feature transformations and classifiers. Top left: SRC (our approach). Top right: nearest neighbor. Bottom left: nearest subspace. Bottom right: support vector machine (linear kernel).

C. Recognition Despite Random Pixel Corruption

For this experiment, we test the robust version of SRC, which solves the extended ℓ^1-minimization problem (22), using the Extended Yale B Face Database. We choose Subsets 1 and 2 (717 images, normal-to-moderate lighting conditions) for training, and Subset 3 (453 images, more extreme lighting conditions) for testing. Without occlusion, this is a relatively easy recognition



Features	Nose	Right Eye	Mouth & Chin
Dimension (d)	4,270	5,040	12,936
SRC	**87.3%**	**93.7%**	**98.3%**
NN	49.2%	68.8%	72.7%
NS	83.7%	78.6%	94.4%
SVM	70.8%	85.8%	95.3%

Fig. 10. **Recognition with partial face features.** Top: example features. Bottom: Recognition rates of SRC, NN, NS, and SVM on the Extended Yale B database.

problem. This choice is deliberate, in order to isolate the effect of occlusion. The images are resized to 96×84 pixels,[19] so in this case $B = [A,\ I]$ is an $8,064 \times 8,761$ matrix. For this dataset, we have estimated that the polytope $P = \text{conv}(\pm B)$ is approximately $1,185$-neighborly (using the method given in [37]), suggesting that perfect reconstruction can be achieved upto 13.3% (worst possible) occlusion.

We corrupt a percentage of randomly chosen pixels from each of the test images, replacing their values with iid samples from a uniform distribution[20]. The corrupted pixels are randomly chosen for each test image and the locations are unknown to the algorithm. We vary the percentage of corrupted pixels from 0% to 90%. Figure 11 (left) shows several example test images. To the human eye, beyond 50% corruption, the corrupted images (Figure 11(a) second and third rows) are barely recognizable as face images; determining their identity seems out of the question. Yet even in this extreme circumstance, SRC correctly recovers the identity of the subjects.

We quantitatively compare our method to four popular techniques for face recognition in the vision literature. The Principal Component Analysis (PCA) approach of [23] is not robust to occlusion. There are many variations to make PCA robust to corruption or incomplete data, and some have been applied to robust face recognition, e.g., [29]. We will later discuss their performance against ours on more realistic conditions. Here we use the basic PCA to provide a standard baseline for comparison[21]. The remaining three techniques are designed to be more robust to occlusion. Independent Component Analysis (ICA) architecture I [53] attempts to express the training set as a linear combination of statistically independent basis images. Local Nonnegative Matrix Factorization (LNMF) [54] approximates the training set as an additive combination of basis images, computed with a bias toward sparse bases.[22] Finally, to demonstrate that the improved robustness is really due to the use of the ℓ^1-norm, we compare to a least-squares technique that first projects the test image onto the subspace spanned by all face images, and then performs nearest subspace.

[19]The only reason for resizing the images is to be able to run all the experiments within the memory size of MATLAB on a typical PC. The algorithm relies on linear programming and is scalable in the image size.

[20]Uniform over $[0, y_{max}]$, where y_{max} is the largest possible pixel value.

[21]Following [58] we normalize the image pixels to have zero mean and unit variance before applying PCA.

[22]For PCA, ICA and LNMF, the number of basis components is chosen to give the optimal test performance over the range $\{100, 200, 300, 400, 500, 600\}$.

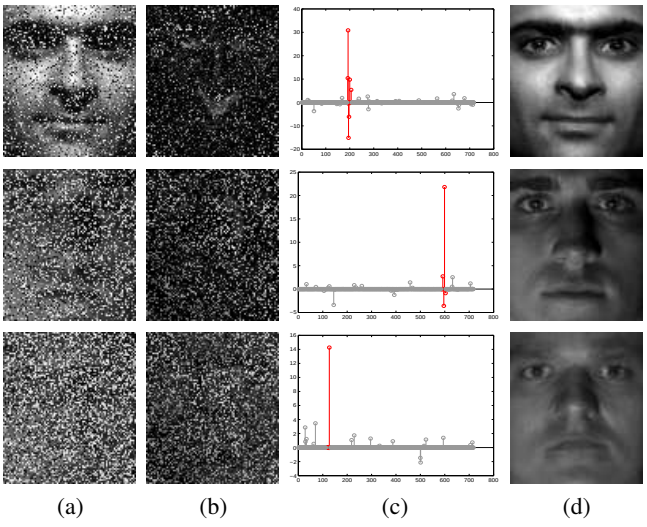
Figure 11 (right) plots the recognition performance of SRC and its five competitors, as a function of the level of corruption. We see that the algorithm dramatically outperforms others. From 0% upto 50% occlusion, SRC correctly classifies all subjects. At 50% corruption, none of the others achieves higher than 73% recognition rate, while the proposed algorithm achieves 100%. Even at 70% occlusion, the recognition rate is still 90.7%. This greatly surpasses the theoretical bound of worst-case corruption (13.3%) that the algorithm is ensured to tolerate. Clearly, the worst-case analysis is too conservative for random corruption.

D. Recognition Despite Random Block Occlusion

We next simulate various levels of contiguous occlusion, from 0% to 50%, by replacing a *randomly located* square block of each test image with an unrelated image, as in Figure 12(a). Again, the location of occlusion is randomly chosen for each image and is unknown to the computer. Methods that select fixed facial features or blocks of the image (e.g., [16], [57]) are less likely to succeed here, due to the unpredictable location of the occlusion. The top two rows of Figure 12 left shows two representative results of Algorithm 1 with 30% occlusion. Figure 12(a) is the occluded image. In the second row, the entire center of the face is occluded; this is a difficult recognition task even for humans. Figure 12(b) shows the magnitude of the estimated error $\hat{e}_1$. Notice that $\hat{e}_1$ compensates not only for occlusion due to the baboon, but also for the violation of the linear subspace model caused by the shadow under the nose. Figure 12(c) plots the estimated coefficient vector $\hat{x}_1$. The red entries are coefficients corresponding to test image's true class. In both examples, the estimated coefficients are indeed sparse, and have large magnitude only for training images of the same person. In both cases, the SRC algorithm correctly classifies the occluded image. For this dataset, our Matlab implementation requires 90 seconds per test image on a PowerMac G5.

The graph in Figure 12 (right) shows the recognition rates of all six algorithms. SRC again significantly outperforms the other five methods, for all levels of occlusion. Upto 30% occlusion, Algorithm 1 performs almost perfectly, correctly identifying over 98% of test subjects. Even at 40% occlusion, only 9.7% of subjects are misclassified. Compared to random pixel corruption, contiguous occlusion is certainly a worse type of errors for the algorithm. Notice, though, that the algorithm does not assume any knowledge about the nature of corruption or occlusion. In Section IV-F, we will see how prior knowledge that the occlusion is contiguous can be used to customize the algorithm and greatly enhance the recognition performance.

This result has interesting implications for the debate over the use of holistic versus local features in face recognition [22]. It has been suggested that both ICA I and LNMF are robust to occlusion: since their bases are locally concentrated, occlusion corrupts only a fraction of the coefficients. By contrast, if one uses ℓ^2-minimization (orthogonal projection) to express an occluded image in terms of a holistic basis such as the training images themselves, all of the coefficients may be corrupted (as in Figure 12 left third row). The implication here is that the problem is *not* the choice of representing the test image in terms of a holistic or local basis, but rather *how the representation is computed.* Properly harnessing redundancy and sparsity is the key to error correction and robustness. Extracting local or disjoint features can only reduce redundancy, resulting in inferior robustness.



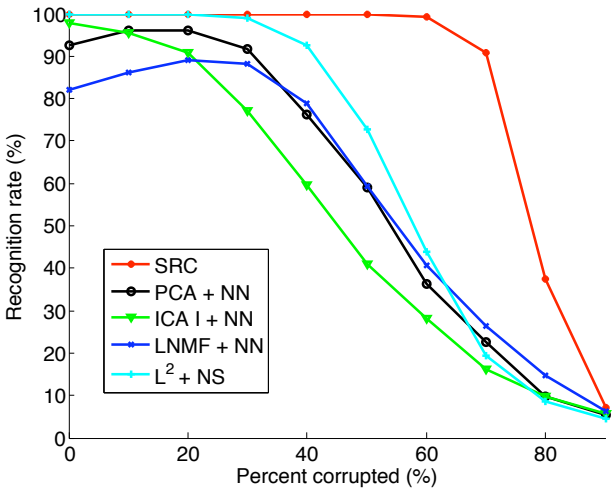
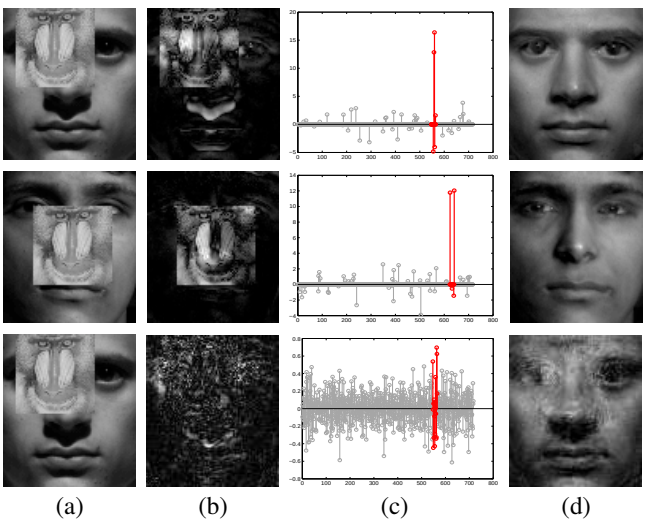


Fig. 11. **Recognition under random corruption.** Left: (a) Test images $\boldsymbol{y}$ from Extended Yale B, with random corruption. Top row: 30% of pixels are corrupted, Middle row: 50% corrupted, Bottom row: 70% corrupted. (b) Estimated errors $\hat{\boldsymbol{e}}_1$. (c) Estimated sparse coefficients $\hat{\boldsymbol{x}}_1$. (d) Reconstructed images $\boldsymbol{y}_r$. SRC correctly identifies all three corrupted face images. Right: The recognition rate across the entire range of corruption for various algorithms. SRC (red curve) significantly outperforms others, performing almost perfectly upto 60% random corruption (see table below).

Percent corrupted	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%
Recognition rate	100%	100%	100%	100%	100%	100%	99.3%	90.7%	37.5%	7.1%



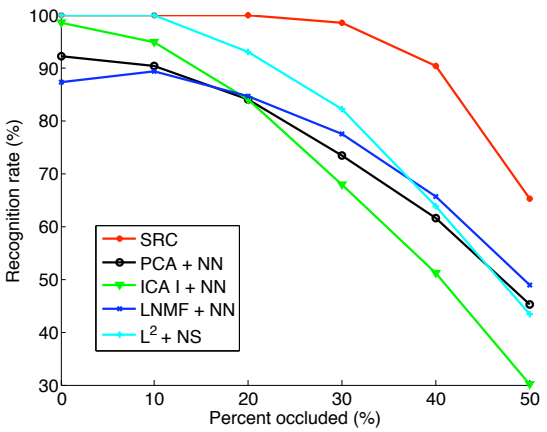


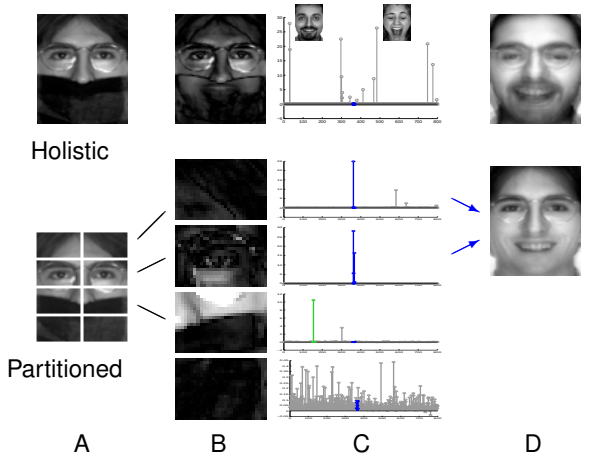
Fig. 12. **Recognition under varying level of contiguous occlusion.** Left, top two rows: (a) 30% occluded test face images $\boldsymbol{y}$ from Extended Yale B. (b) Estimated sparse errors, $\hat{\boldsymbol{e}}_1$. (c) Estimated sparse coefficients, $\hat{\boldsymbol{x}}_1$, red (darker) entries correspond to training images of the same person. (d) Reconstructed images, $\boldsymbol{y}_r$. SRC correctly identifies both occluded faces. For comparison, the bottom row shows the same test case, with the result given by least-squares (overdetermined ℓ^2-minimization). Right: The recognition rate across the entire range of corruption for various algorithms. SRC (red curve) significantly outperforms others, performing almost perfectly upto 30% contiguous occlusion (see table below).

Percent occluded	0%	10%	20%	30%	40%	50%
Recognition rate	100%	100%	99.8%	98.5%	90.3%	65.3%

E. Recognition Despite Disguise

We test SRC's ability to cope with real, possibly malicious, occlusions using a subset of the AR Face Database. The chosen subset consists of $1,399$ images (14 each, except for a corrupted image `w-027-14.bmp`) of 100 subjects, 50 male and 50 female. For training, we use 799 images (about 8 per subject) of unoccluded frontal views with varying facial expression, giving a matrix B of size $4,980 \times 5,779$. We estimate $P = \text{conv}(\pm B)$ is approximately 577-neighborly, indicating that perfect reconstruction is possible upto 11.6% occlusion. Our Matlab implementation requires about 75 seconds per test image on a PowerMac G5.

We consider two separate test sets of 200 images. The first test set contains images of the subjects wearing sunglasses, which occlude roughly 20% of the image. Figure 1 top shows a successful example from this test set. Notice that $\hat{\boldsymbol{e}}_1$ compensates for small misalignment of the image edges as well as occlusion due to sunglasses. The second test set considered contains images of the subjects wearing a scarf, which occludes roughly 40% of the image. Since the occlusion level is more than three times the maximum worst case occlusion given by the neighborliness of $\text{conv}(\pm B)$, our approach is unlikely to succeed in this domain. Figure 13 top shows one such failure. Notice that the largest coefficient corresponds to an image of a bearded man whose



Algorithms	Rec. rate sunglasses	Rec. rate scarves
SRC (partitioned)	**87.0% (97.5%)**	**59.5% (93.5%)**
PCA + NN	70.0%	12.0%
ICA I + NN	53.5%	15.0%
LNMF + NN	33.5%	24.0%
ℓ^2 + NS	64.5%	12.5%

Fig. 13. **Top: Partition scheme to tackle contiguous disguise.** The top row visualizes an example for which SRC failed with the whole image (holistic). The two largest coefficients correspond to a bearded man and a screaming woman, two images whose mouth region resembles the occluding scarf. If the occlusion is known to be contiguous, one can partition the image into multiple smaller blocks, apply the SRC algorithm to each of the blocks and then aggregate the results by voting. The second row visualizes how this partition-based scheme works on the same test image, but leading to a correct identification. (A) The test image, occluded by scarf. (B) Estimated sparse error $\hat{\boldsymbol{e}}_1$. (C) Estimated sparse coefficients $\hat{\boldsymbol{x}}_1$. (D) Reconstructed image. **Bottom: Table of recognition rates on the AR database.** The table shows the performance of all the algorithms for both types of occlusion. SRC, its holistic version (right top) and partitioned version (right bottom), achieves the highest recognition rate.

mouth region resembles the scarf.

The table in Figure 13 left compares SRC to the other five algorithms described in the previous section. On faces occluded by sunglasses, SRC achieves a recognition rate of 87%, more than 17% better than the nearest competitor. For occlusion by scarves, its recognition rate is 59.5%, more than double its nearest competitor but still quite poor. This confirms that although the algorithm is provably robust to *arbitrary* occlusions upto the breakdown point determined by the neighborliness of the training set, beyond that point it is sensitive to occlusions that resemble regions of a training image from a different individual. Because the amount of occlusion exceeds this breakdown point, additional assumptions, such as the disguise is likely to be contiguous, are needed to achieve higher recognition performance.

F. Improving Recognition by Block Partitioning

Thus far we have not exploited the fact that in many real recognition scenarios, the occlusion falls on some patch of image pixels which is a-priori unknown, but is known to be connected. A somewhat traditional approach (explored in [57] amongst others) to exploiting this information in face recognition is to partition the image into blocks and process each block independently. The results for individual blocks are then aggregated, for example, by voting, while discarding blocks believed to be occluded (using, say, the outlier rejection rule introduced in Section II-D). The major difficulty with this approach is that the occlusion cannot be expected to respect any fixed partition of the image; while only a few blocks are assumed to be completely occluded, some or all of the remaining blocks may be partially occluded. Thus, in such a scheme there is still a need for robust techniques *within each block*.

We partition each of the training images into L blocks of size $a \times b$, producing a set of matrices $A^{(1)}, \ldots, A^{(L)} \in \mathbb{R}^{p \times n}$, where $p \doteq ab$. We similarly partition the test image $\boldsymbol{y}$ into $\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(L)} \in \mathbb{R}^p$. We write the l-th block of the test image as a sparse linear combination $A^{(l)}\boldsymbol{x}^{(l)}$ of l-th blocks of the training images, plus a sparse error $\boldsymbol{e}^{(l)} \in \mathbb{R}^p$: $\boldsymbol{y}^{(l)} = A^{(l)}\boldsymbol{x}^{(l)} + \boldsymbol{e}^{(l)}$. We can recover can again recover a sparse $\boldsymbol{w}^{(l)} = [\boldsymbol{x}^{(l)}\ \boldsymbol{e}^{(l)}] \in \mathbb{R}^{n+p}$ by ℓ^1 minimization:

$$\hat{\boldsymbol{w}}_1^{(l)} \doteq \arg\min_{\boldsymbol{w} \in \mathbb{R}^{n+p}} \|\boldsymbol{w}\|_1 \quad \text{subject to} \quad \left[A^{(l)}\ I\right]\boldsymbol{w} = \boldsymbol{y}^{(l)}. \quad (24)$$

We apply the classifier from Algorithm 1 within each block[23] and then aggregate the results by voting. Figure 13 illustrates this scheme.

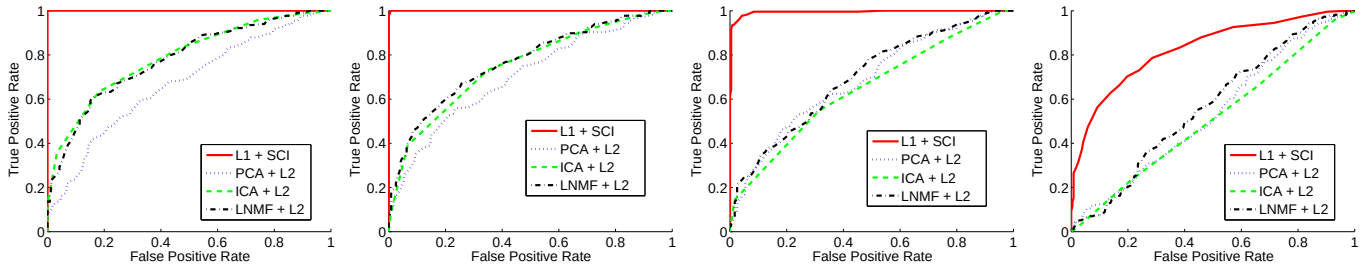
We verify the efficacy of this scheme on the AR database for faces disguised with sunglasses or scarves. We partition the images into eight (4×2) blocks of size 20×30 pixels. Partitioning increases the recognition rate on scarves from 59.5% to 93.5%, and also improves the recognition rate on sunglasses from 87.0% to 97.5%. This performance exceeds the best known results on the AR dataset [29] to date. That work obtains 84% on the sunglasses and 93% on the scarfs, on only 50 subjects, using more sophisticated random sampling techniques. Also noteworthy is [16], which aims to recognize occluded faces from only a single training sample per subject. On the AR database, that method achieves a lower combined recognition rate of 80%.[24]

G. Rejecting Invalid Test Images

We next demonstrate the relevance of sparsity for rejecting invalid test images, with or without occlusion. We test the outlier rejection rule (15) based on the Sparsity Concentration Index (14) on the Extended Yale B database, using Subsets 1 and 2 for training and Subset 3 for testing as before. We again simulate varying levels of occlusion (10%, 30%, and 50%) by replacing a randomly chosen block of each test image with an unrelated image. However, in this experiment, we include only half of the subjects in the training set. Thus, half of the subjects in the testing set are new to the algorithm. We test the system's ability to determine whether a given test subject is in the training database or not by sweeping the threshold τ through a range of values in $[0, 1]$, generating the receiver operator characteristic (ROC) curves in Figure 14. For comparison, we also considered outlier rejection by thresholding the Euclidean distance between (features of) the test image and (features of) the nearest training images within the PCA, ICA and LNMF feature spaces. These curves are also displayed in Figure 14. Notice that the simple rejection rule (15) performs nearly perfectly at 10% and 30% occlusion. At 50% occlusion, it still significantly outperforms the other three algorithms, and is the only one of the four algorithms

[23] Occluded blocks can also be rejected via (15). We find that this does not significantly increase the recognition rate.

[24] From our own implementation and experiments, we find their method does not generalize well to more extreme illuminations.



(a) No occlusion (b) 10% occlusion (c) 30% occlusion (d) 50% occlusion

Fig. 14. **ROC curves for outlier rejection.** Vertical axis: true positive rate. Horizontal axis: false positive rate. The solid red curve is generated by SRC with outliers rejected based on equation (15). The SCI-based validation and SRC classification together perform almost perfectly for upto 30% occlusion.

that performs significantly better than chance. The supplementary appendix contains more validation results on the AR database using Eigenfaces, again demonstrating significant improvement in the ROC.

H. Designing the Training Set for Robustness

An important consideration in designing recognition systems is selecting the number of training images as well as the conditions (lighting, expression, viewpoint etc.) under which they are to be taken. The training images should be extensive enough to span the conditions that might occur in the test set: they should be "sufficient" from a pattern recognition standpoint. For instance, [59] shows how to choose the fewest representative images to well-approximate the illumination cone of each face. The notion of neighborliness discussed in Section II provides a different, quantitative measure for how "robust" the training set is: the amount of worst-case occlusion the algorithm can tolerate is directly determined by how neighborly the associated polytope is. The worst case is relevant in visual recognition, since the occluding object could potentially be quite similar to one of the other training classes. However, if the occlusion is random and uncorrelated with the training images, as in Section IV-C, the average behavior may also be of interest.

In fact, these two concerns, sufficiency and robustness, are complementary. Figure 15 left shows the estimated neighborliness for the four subsets of the Extended Yale B database. Notice that the highest neighborliness, $\approx 1,330$, is achieved with Subset 4, the most extreme lighting conditions. Figure 15 right shows the breakdown point for subsets of the AR database with different facial expressions. The dataset contains four facial expressions, Neutral, Happy, Angry, and Scream, pictured in Figure 15 right. We generate training sets from all pairs of expressions and compute the neighborliness of each of the corresponding polytopes. The most robust training sets are achieved by the Neutral+Happy and Happy+Scream combinations, while the least robustness comes from Neutral+Angry. Notice that the Neutral and Angry images are quite similar in appearance, while (for example) Happy and Scream are very dissimilar.

Thus, both for varying lighting (Figure 15 left) and expression (Figure 15 right), training sets with wider variation in the images allow greater robustness to occlusion. Designing a training set that allows recognition under widely varying conditions does not hinder our algorithm; in fact it helps it. However, the training set should not contain too many similar images, as in the Neutral+Angry example of Figure 15 right. In the language of signal representation, the training images should form an *incoherent dictionary* [9].

V. Conclusions and Discussions

In this paper, we have contended both theoretically and experimentally that exploiting sparsity is critical for high-performance classification of high-dimensional data such as face images. 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). Moreover, occlusion and corruption can be handled uniformly and robustly within the same classification framework. One can achieve striking recognition performance for severely occluded or corrupted images by a simple algorithm with no special engineering.

An intriguing question for future work is whether this framework can be useful for object detection, in addition to recognition. The usefulness of sparsity in detection has been noticed in the work of [61] and more recently explored in [62]. We believe that the full potential of sparsity in robust object detection and recognition together is yet to be uncovered. From a practical standpoint, it would also be useful to extend the algorithm to less constrained conditions, especially variations in object pose. Robustness to occlusion allows the algorithm to tolerate small pose variation or misalignment. Furthermore, in the supplementary appendix, we discuss our algorithm's ability to adapt to nonlinear training distributions. However, the number of training samples required to directly represent the distribution of face images under varying pose may be prohibitively large. Extrapolation in pose, e.g., using only frontal training images, will require integrating feature matching techniques or nonlinear deformation models into the computation of the sparse representation of the test image. Doing so in a principled manner remains an important direction for future work.

Acknowledgments

We would like to thank Dr. Harry Shum, Dr. Xiaoou Tang and many others at Microsoft Research in Asia for helpful and informative discussions on face recognition, during our visit there in Fall 2006. We also thank Prof. Harm Derksen and Prof. Michael Wakin of the University of Michigan, Prof. Robert Fossum and Yoav Sharon of the University of Illinois for advice and discussions on polytope geometry and sparse representation. This work was partially supported by the grants ARO MURI W911NF-06-1-0076, NSF CAREER IIS-0347456, NSF CRS-EHS-0509151, NSF CCF-TF-0514955, ONR YIP N00014-05-1-0633, NSF ECCS07-01676, and NSF IIS 07-03756.

Training set	1	2	3	**4**
Neighborliness	1,124	1,122	1,190	**1,330**

Training set	**N+H**	N+A	N+S	H+A	H+S	A+S
Neighborliness	**585**	421	545	490	550	510

Fig. 15. **Robust training set design.** Left: varying illumination. Top left: four subsets of Extended Yale B, containing increasingly extreme lighting conditions. Bottom left: estimated neighborliness of the polytope $\text{conv}(\pm B)$ for each subset. Right: varying expression. Top right: four facial expressions in the AR database. Bottom right: estimated neighborliness of $\text{conv}(\pm B)$ when taking the training set from different pairs of expressions.

REFERENCES

[1] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471, 1978.

[2] M. Hansen and B. Yu, "Model selection and the minimum description length principle," *Journal of the American Statistical Association*, vol. 96, pp. 746–774, 2001.

[3] A. d'Aspremont, L. E. Ghaoui, M. Jordan, and G. Lanckriet, "A direct formulation of sparse PCA using semidefinite programming," *SIAM Review*, vol. 49, pp. 434–448, 2007.

[4] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Neural Information Processing Systems (NIPS)*, 2006.

[5] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2000.

[6] T. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. 14, no. 3, pp. 326–334, 1965.

[7] B. Olshausen and D. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vision Research*, vol. 37, pp. 3311–3325, 1997.

[8] T. Serre, "Learning a dictionary of shape-components in visual cortex: Comparison with neurons, humans and machines," Ph.D. dissertation, MIT, 2006.

[9] 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.

[10] 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.

[11] 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.

[12] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, no. 7, pp. 2541–2567, 2006.

[13] 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.

[14] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society B*, vol. 58, no. 1, pp. 267–288, 1996.

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

[16] A. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748–763, 2002.

[17] B. Park, K. Lee, and S. Lee, "Face recognition using face-ARG matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1982–1988, 2005.

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

[19] 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.

[20] S. Li and J. Lu, "Face recognition using the nearest feature line method," *IEEE Trans. Neural Networks*, vol. 10, no. 2, pp. 439–443, 1999.

[21] 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.

[22] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, pp. 399–458, 2003.

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

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

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

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

[27] 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.

[28] A. Leonardis and H. Bischof, "Robust recognition using eigenimages," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 99–118, 2000.

[29] F. Sanja, D. Skocaj, and A. Leonardis, "Combining reconstructive and discriminative subspace methods for robust classification and regression by subsampling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, 2006.

[30] R. Basri and D. Jacobs, "Lambertian reflection and linear subspaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 3, pp. 218–233, 2003.

[31] 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.

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

[33] D. Donoho and M. Elad, "Optimal sparse representation in general (nonorthogonal) dictionaries via ℓ^1 minimization," *Proceedings of the National Academy of Sciences of the United States of America*, pp. 2197–2202, March 2003.

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

[35] 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.

[36] D. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," *Dept. of Statistics TR 2005-4, Stanford University*, 2005.

[37] Y. Sharon, J. Wright, and Y. Ma, "Computation and relaxation of conditions for equivalence between ℓ^1 and ℓ^0 minimization," *CSL Tech. Report UILU-ENG-07-2208, University of Illinois at Urbana-Champaign*, 2007.

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

[39] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

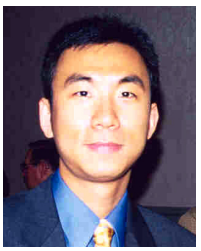
[40] E. Candes and J. Romberg, "ℓ^1-magic: Recovery of sparse signals via convex programming," *http://www.acm.caltech.edu/l1magic/*, 2005.

[41] 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.

[42] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Trans. on*

Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 725–737, 2006.
[43] 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.
[44] 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.
[45] H. Rauhut, K. Schnass, and P. Vandergheynst, "Compressed sensing and redundant dictionaries," to appear in *IEEE Transactions on Information Theory*, 2007.
[46] D. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, 2000.
[47] S. Kaski, "Dimensionality reduction by random mapping," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 1, 1998, pp. 413–418.
[48] D. Achlioptas, "Database-friendly random projections," in *Proceedings of the ACM Symposium on Principles of Database Systems*, 2001, pp. 274–281.
[49] 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.
[50] R. Baraniuk and M. Wakin, "Random projections of smooth manifolds," to appear in *Foundations of Computational Mathematics*, 2007.
[51] R. Baraniuk, M. Davenport, R. de Vore, and M. Wakin, "The Johnson-Lindenstrauss lemma meets compressed sensing," *to appear in Constructive Approximation*, 2007.
[52] F. Macwilliams and N. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1981.
[53] J. Kim, J. Choi, J. Yi, and M. Turk, "Effective representation using ICA for face recognition robust to local distortion and partial occlusion," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1977–1981, 2005.
[54] S. Li, X. Hou, H. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 1–6.
[55] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.
[56] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Transactions on Computers*, vol. 42, pp. 300–311, 1993.
[57] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 1994.
[58] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.
[59] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.
[60] A. Martinez and R. Benavente, "The AR face database," *CVC Tech. Report No. 24*, 1998.
[61] D. Geiger, T. Liu, and M. Donahue, "Sparse representations for image decompositions," *International Journal of Computer Vision*, vol. 33, no. 2, 1999.
[62] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *Proc. Neural Information and Processing Systems*, 2006.

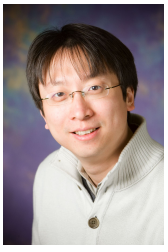
John Wright received his BS in Computer Engineering and MS in Electrical Engineering from the University of Illinois at Urbana-Champaign. He is currently a PhD candidate in the Decision and Control Group at the University of Illinois. His research interests included automatic face and object recognition, sparse signal representation, and minimum description length techniques in supervised and unsupervised learning, and has published more than 10 papers on these subjects. He has been the recipient of several awards and fellowships, including the UIUC ECE Distinguished Fellowship and a Carver Fellowship. Most recently, in 2008 he received a Microsoft Research Fellowship, sponsored by Microsoft Live Labs.



Allen Y. Yang is a postdoctoral researcher in the department of EECS at UC Berkeley. His primary research is on pattern analysis of geometric or statistical models in very high-dimensional data space, and applications in motion segmentation, image segmentation, face recognition, and signal processing in heterogeneous sensor networks. He has co-authored five journal papers and more than 10 conference papers. He is also the co-inventor of two US patent applications. He received his Bachelor's degree in Computer Science from the University of Science and Technology of China in 2001, and his PhD in Electrical and Computer Engineering from the University of Illinois at Urbana-Champaign in 2006.

Arvind Ganesh received the Bachelor's and Master's degrees, both in Electrical Engineering, from the Indian Institute of Technology, Madras, India, in 2006. He is currently pursuing the PhD degree in Electrical Engineering at the University of Illinois at Urbana-Champaign. His research interests include computer vision, machine learning, and signal processing.

S. Shankar Sastry received his Ph.D. degree in 1981 from the University of California, Berkeley. He was on the faculty of MIT as Assistant Professor from 1980-82 and Harvard University as a chaired Gordon McKay professor in 1994. He served as Chairman of the EECS Department, UC Berkeley from 2001 to 2004. In 2000, he served as Director of the Information Technology Office at DARPA. He is currently the Roy W. Carlson Professor of Electrical Engineering and Computer Science, Bioengineering and Mechanical Engineering, as well as the Dean of the College of Engineering at UC Berkeley. He also serves as the Director of the Blum Center for Developing Economies. Shankar Sastry has coauthored over 300 technical papers and 9 books. He received the President of India Gold Medal in 1977, the IBM Faculty Development award for 1983-1985, the NSF Presidential Young Investigator Award in 1985, the Eckman Award of the American Automatic Control Council in 1990, an M. A. (honoris causa) from Harvard in 1994, Fellow of the IEEE in 1994, the distinguished Alumnus Award of the Indian Institute of Technology in 1999, the David Marr prize for the best paper at the International Conference in Computer Vision in 1999, and the Ragazzini Award for Excellence in Education by the American Control Council in 2005. He is a member of the National Academy of Engineering and the American Academy of Arts and Sciences. He is on the Air Force Science Board and is Chairman of the Board of the International Computer Science Institute. He is also a member of the boards of the Federation of American Scientists and ESCHER (Embedded Systems Consortium for Hybrid and Embedded Research).



Yi Ma (SM'06) received his two Bachelors' degrees in Automation and Applied Mathematics from Tsinghua University, Beijing, China in 1995. He received an M.S. degree in Electrical Engineering and Computer Science (EECS) in 1997, an M.A. degree in Mathematics in 2000, and a PhD degree in EECS in 2000 all from UC Berkeley. Since 2000, he has been on the faculty of the Electrical & Computer Engineering Department of the University of Illinois at Urbana-Champaign, where he now holds the rank of associate professor. His main research areas are in systems theory and computer vision. Yi Ma was the recipient of the David Marr Best Paper Prize at the International Conference on Computer Vision in 1999 and Honorable Mention for the Longuet-Higgins Best Paper Award at the European Conference on Computer Vision in 2004. He received the CAREER Award from the National Science Foundation in 2004 and the Young Investigator Program Award from the Office of Naval Research in 2005. He is a senior member of IEEE and a member of ACM.

APPENDIX
I. RELATIONSHIPS TO NEAREST NEIGHBOR AND NEAREST SUBSPACE

One may notice that the use of *all* the training samples of *all* classes to represent the test sample goes against the conventional classification methods popular in face recognition literature and existing systems. These methods typically suggest using residuals computed from "one sample at a time" or "one class at a time" to classify the test sample. The representative methods include:

1) The *nearest neighbor* (NN) classifier: Assign the test sample $\boldsymbol{y}$ to class i if the smallest distance from $\boldsymbol{y}$ to the nearest training sample of class i
$$r_i(\boldsymbol{y}) = \min_{j=1,\ldots,n_i} \|\boldsymbol{y} - \boldsymbol{v}_{i,j}\|_2 \tag{25}$$
is the smallest among all classes.[25]
2) The *nearest subspace* (NS) classifier (e.g., [32]): Assign the test sample $\boldsymbol{y}$ to class i if the distance from $\boldsymbol{y}$ to the subspace spanned by all samples $A_i = [\boldsymbol{v}_{i,1}, \ldots, \boldsymbol{v}_{i,n_i}]$ of class i:
$$r_i(\boldsymbol{y}) = \min_{\alpha_i \in \mathbb{R}^{n_i}} \|\boldsymbol{y} - A_i\alpha_i\|_2 \tag{26}$$
is the smallest among all classes.

Clearly, NN seeks the best representation in terms of just a single training sample,[26] while NS seeks the best representation in terms of all the training samples of each class. The *nearest feature line* (NFL) algorithm [20] strikes a balance between these two by considering the distance of $\boldsymbol{y}$ to the line spanned by any pair of training samples. As NN and NS represent the two extreme cases, we will compare our method with them and see how enforcing sparsity can strike a better balance than methods like NFL.

A. Relationship to Nearest Neighbor

Let us first assume that a test sample $\boldsymbol{y}$ can be well-represented in terms of one training sample, say $\boldsymbol{v}_i$ (one of the columns of A):
$$\boldsymbol{y} = \boldsymbol{v}_i + \boldsymbol{z}_i \tag{27}$$
where $\|\boldsymbol{z}_i\|_2 \leq \varepsilon$ for some small $\varepsilon > 0$. As discussed in Section II-B.2, the recovered sparse solution $\hat{\boldsymbol{x}}$ to (10) satisfies
$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2 \leq \zeta\varepsilon$$
where $\boldsymbol{x}_0 \in \mathbb{R}^n$ is the vector whose i-th entry is 1 and others are all zero, and ζ is a constant that depends on A. Thus, in this case, the ℓ^1-minimization based classifier will give the same identification for the test sample as NN.

On the other hand, in face recognition, 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 face images (as we see in the experimental section, 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 3: Figure 16 left shows the ℓ^2-distances between the downsampled 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 IV, this inevitably leads to inferior recognition performance when using NN (only 71.6% in this case, comparing to 92.1% of Algorithm 1).[27]

B. Relationship to Nearest Subspace

Let us now assume that the test sample $\boldsymbol{y}$ can be represented uniquely as a linear combination of the training samples A_i of class i:
$$\boldsymbol{y} = A_i\alpha_i + \boldsymbol{z}_i \tag{28}$$
where $\|\boldsymbol{z}_i\|_2 \leq \varepsilon$ for some small $\varepsilon > 0$. Then again according to equation (11), the recovered sparse solution $\hat{\boldsymbol{x}}$ to (10) satisfies
$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}_0\|_2 \leq \zeta\varepsilon$$
where $\boldsymbol{x}_0 \in \mathbb{R}^n$ is a vector of the form $[0, \ldots, 0, \alpha_i^T, 0, \ldots, 0]^T$. That is,
$$\delta_i(\hat{\boldsymbol{x}}) \approx \boldsymbol{x}_0 \;\text{ and }\; \|\delta_j(\hat{\boldsymbol{x}})\| < \zeta\varepsilon \quad \text{for all } j \neq i. \tag{29}$$
We have
$$\|\boldsymbol{y} - A\delta_i(\hat{\boldsymbol{x}})\|_2 \approx \|\boldsymbol{z}_i\|_2 \leq \varepsilon, \quad \text{and} \tag{30}$$
$$\|\boldsymbol{y} - A\delta_j(\hat{\boldsymbol{x}})\|_2 \approx \|\boldsymbol{y}\|_2 \gg \varepsilon \quad \text{for all } j \neq i. \tag{31}$$
Thus, in this case, the ℓ^1-minimization based classifier will give the same identification for the test sample as NS. Notice that for $j \neq i$, $\delta_j(\hat{\boldsymbol{x}})$ is rather different from α_j computed from $\min_{\alpha_j} \|\boldsymbol{y} - A_j\alpha_j\|_2$. The norm of $\delta_j(\hat{\boldsymbol{x}})$ is bounded by the approximation error (29) when $\boldsymbol{y}$ is represented just within class j, whereas the norm of α_j can be very large as face images of different subjects are highly correlated. Further notice that each of the α_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 $\{\delta_j(\hat{\boldsymbol{x}})\}_{j=1}^k$ 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 $\{\|\boldsymbol{y} - A\delta_j(\hat{\boldsymbol{x}})\|_2\}_{j=1}^k$.

Example 4: Figure 16 right shows the residuals of the downsampled 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 16 right is only 1:3. In other words, the solution from Algorithm 1 is more discriminative than that from NS. As we will see Section IV, for the 12×10 downsampled images, the recognition rate of NS is lower than that of Algorithm 1 (91.1% versus 92.1%).

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 sample is generated from a simple statistical model: $\boldsymbol{y} = A_i\alpha_i + \boldsymbol{z}_i$ with α_i and $\boldsymbol{z}_i$ independent Gaussians, any sufficient statistic (for the

[25] 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.

[26] Alternatively, a similar classifier K-NN considers K nearest neighbors.

[27] Other commonly used distance metrics in NN such as ℓ^1-distance give results similar to Figure 16 left.

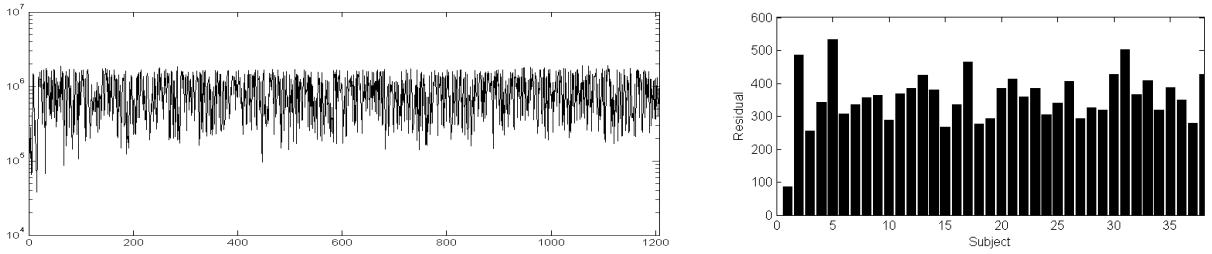


Fig. 16. Left: The ℓ^2-distances (logarithmic scale) between the test image and the training images in Example 1 (as used by nearest neighbor). Right: The residuals of the test image in Example 1 w.r.t. the 38 face subspaces (as used by nearest subspace).

optimal classifier) depends on both $\|\alpha_i\|_2$ and $\|\boldsymbol{z}_i\|_2$, not just the residual $\|\boldsymbol{z}_i\|_2$. While the ℓ^1-minimization based classifier is also suboptimal under this model, it does implicitly use the information in α_i as it penalizes α_i that has a large norm – the ℓ^1-minimization based classifier favors small $\|\boldsymbol{z}_i\|_2$ as well as small $\|\alpha_i\|_1$ in representing the test sample with the training data.

Furthermore, using all the training samples in each class may over-fit the test sample In the case when the solution α_i to

$$\boldsymbol{y} = A_i\alpha_i + \boldsymbol{z}_i \quad \text{subject to} \quad \|\boldsymbol{z}_i\|_2 < \varepsilon$$

is *not unique*, the ℓ^1-minimization (6) will find the sparsest $\alpha_{i0} \in \mathbb{R}^{n_i}$ instead of the least ℓ^2-norm solution $\alpha_{i2} = (A_i^T A_i)^\dagger \boldsymbol{y} \in \mathbb{R}^{n_i}$. That is, the ℓ^1-minimization will use the smallest number of samples necessary in each class to represent the test sample, subject to a small error. To see why the sparse solution α_{i0} respects better the actual distribution of the training samples (inside the subspace spanned by all samples), consider the two situations illustrated in Figure 17.

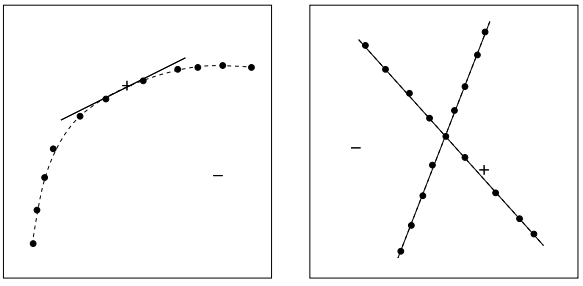


Fig. 17. A sparse solution within the subspace spanned by all training samples of one class. Left: the samples exhibit a nonlinear distribution within the subspace. Right: the samples lie on two lower-dimensional subspaces within the subspace spanned by all the samples.

In the figure on the left, the training samples have a nonlinear distribution within the subspace, say due to pose variation. For the given positive test sample "+," only two training samples are needed to represent it well 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 class are distributed on two lower-dimensional subspaces. This could represent the situation in face recognition 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 class. For example, as we have shown in Figure 3, although subject 1 has 32 training samples, 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 the sample distribution of each class.

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 samples in each class to linearly extrapolate the test sample, but it uses only the smallest necessary number of them to avoid over-fitting. For each test sample, the number of samples needed is automatically determined by the ℓ^1-minimization, because in terms of finding the sparse solution $\boldsymbol{x}_0$, 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 samples of each class and is therefore likely to be more discriminative among multiple classes. These advantages of Algorithm 1 are corroborated by experimental results presented in Section IV as well as the additional experimental results given below.

C. Experimental Comparison

In this subsection, we provide more detailed numerical results, for easy comparison of Algorithm 1 with NN, NS and SVM, in terms of both recognition and validation.

a) Comparison of Recognition Performance: The tables below contain the numerical values plotted in the graphs in Sections IV-A.1 and IV-A.2. Table I gives the performance of our sparse representation based classification (SRC) algorithm on the Extended Yale B database, across different feature transformations and feature dimensions. Here, "E-Random" refers to a variant of the algorithm that uses an *ensemble* of multiple random projections to compute averaged residuals r_i (here, 5 different random projections are used). Aggregating multiple random projections improves the stability of the algorithm, leading to better classification performance. Table II gives the corresponding results for NN and NS. Similarly, using the same experimental setup in Section IV-A.2, Table III gives the result for Algorithm 1, and Table IV for NN, NS and SVM.

b) Comparison of Validation Performance: In Section IV-G, we have demonstrated the ability of the robust version of Algorithm 1 to reject invalid test images, in the presence of occlusion. Here, we present further experimental results comparing the algorithm's outlier rejection capability to that of nearest neighbor and nearest subspace, this time without occlusion, working with features rather than the raw image itself. Conventionally, the two major indices used to measure the accuracy of outlier rejection are

TABLE I

RECOGNITION RATES OF SRC ON THE EXTENDED YALE B DATABASE.

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), NEAREST SUBSPACE (CENTER) AND SUPPORT VECTOR MACHINE (RIGHT) ON THE EXTENDED YALE B DATABASE.

	NN				NS				SVM			
Dimension (d)	30	56	120	504	30	56	120	504	30	56	120	504
Eigen [%]	74.3	81.4	85.5	88.4	**89.9**	**91.1**	92.5	93.2	70.6	84.3	93.1	96.8
Laplacian [%]	77.1	**83.5**	**87.2**	**90.7**	89.0	90.4	91.9	93.4	72.0	**85.0**	**94.0**	**97.7**
Random [%]	70.3	75.6	78.8	79.0	87.3	91.5	**93.9**	**94.1**	48.8	68.6	83.4	91.4
Downsample [%]	51.7	62.6	71.6	78.0	80.8	88.2	91.1	93.4	48.9	69.5	79.0	91.6
Fisher [%]	**87.6**	N/A	N/A	N/A	81.9	N/A	N/A	N/A	**86.7**	N/A	N/A	N/A

TABLE III

RECOGNITION RATES OF SRC ON THE AR DATABASE.

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), NEAREST SUBSPACE (CENTER) AND SUPPORT VECTOR MACHINE (RIGHT) ON THE AR DATABASE.

	NN				NS				SVM			
Dimension (d)	30	54	130	540	30	54	130	540	30	54	130	540
Eigen [%]	68.1	74.8	79.3	80.5	64.1	77.1	82.0	85.1	73.0	84.3	89.0	92.0
Laplacian [%]	73.1	77.1	**83.8**	**89.7**	65.9	77.5	**84.3**	**90.3**	73.4	85.8	**90.8**	**95.7**
Random [%]	56.6	63.7	71.4	75.0	59.2	68.2	80.0	83.3	54.1	70.8	81.6	88.8
Downsample [%]	51.6	60.9	69.2	73.7	56.2	67.7	77.0	82.1	51.4	73.0	83.4	90.3
Fisher [%]	**83.4**	**86.8**	N/A	N/A	**80.3**	**85.8**	N/A	N/A	**86.3**	**93.3**	N/A	N/A

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 IV-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.[28] We compare three algorithms: Algorithm 1, NN, and NS. To be fair, all three algorithms use exactly the same features, 504-dimensional eigenfaces.[29]

Figure 18 shows the FAR-VR curves obtained under the two settings. Notice that Algorithm 1 significantly outperforms NS and NN, as expected. Compared to the performance in Section IV-G, 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.

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

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

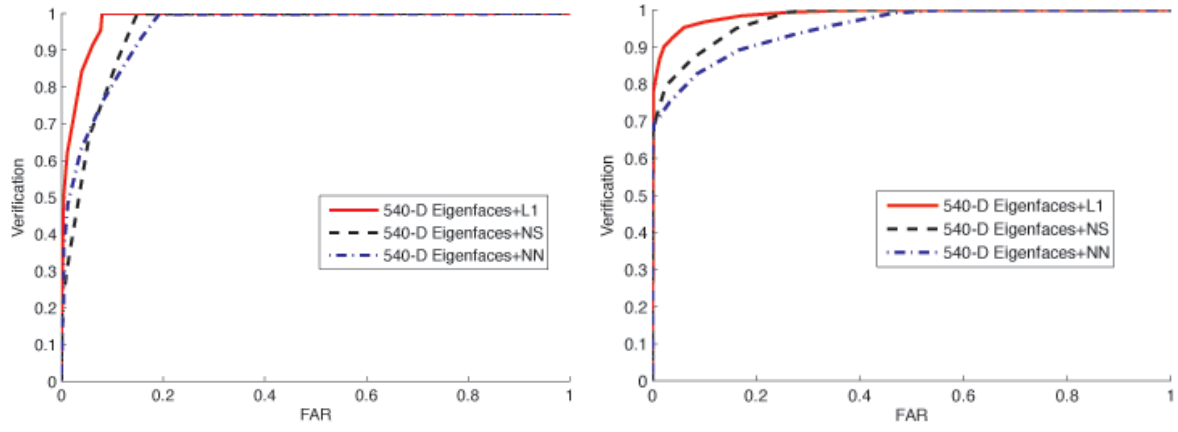


Fig. 18. The FAR-VR curves (solid, red) for SRC using Eigenfaces, compared with the curves of NS and NN using Eigenfaces. Left: 700 images for all 100 subjects in the training, no real outliers in the 700 test images. Right: 469 images for 67 subjects in the training, about half of the 931 test images are true outliers.