Face Verification Using Sparse Representations

Huimin Guo[1], Ruiping Wang[1,2], Jonghyun Choi[1], Larry S. Davis[1]
[1]Institute for Advanced Computer Studies, University of Maryland, College Park, MD, 20742
[2]TNLIST, Department of Automation, Tsinghua University, Beijing, 100084, China
{hmguo, jhchoi, lsd}@umiacs.umd.edu, rpwang@tsinghua.edu.cn

Abstract

We propose a face verification framework using sparse representations that integrates two ways of employing sparsity. Given an image pair (A,B) and a dictionary $\boldsymbol{D}$, for image A(B), we generate two sparse codes, one by using the original dictionary and the other by adding B(A) into D as an augmented dictionary. Then the correlation of the sparse codes of A and B, both under the original dictionary $\boldsymbol{D}$, measuring how similar the pair is, is referred to as the similarity score. The dissimilarity of the sparse codes of A(B), respectively under $\boldsymbol{D}$ and $\boldsymbol{D}$+B(A), is referred to as the dissimilarity score. We exploit multiple feature transforms to obtain several scores using these two measures and fuse them by simple averaging for the situation where no training set is available or by an SVM when a training set is given. We evaluate our algorithm on the LFW dataset, where it is shown to outperform state-of-the-art methods in the unsupervised setting by a large margin and delivers very comparable performance to methods in the image restricted setting despite its simplicity.

1. Introduction

Face recognition (see [21, 29] for recent surveys) has long been an active research field in computer vision, driven by its wide range of practical applications in access control, identification systems, surveillance, pervasive computing, social networks, etc. Face recognition mainly involves the following three tasks: *identification* (*1:N matching*), *verification* (*1:1 matching*), and *watch-list*. In the *identification* task, the goal is to find a nearest neighbor of a given *probe* in a *gallery* face set. In the *verification* task, given a pair of face images, the goal is to determine whether they are coming from a single subject or not. In the *watch-list* task, the recognition system first determines the identity of the query face image in a given *watch-list*, and then identifies the individual. Here, we focus on face verification which is an important tool for authentication of an individual and widely used in security and e-commerce applications.

Face verification in unconstrained environments is a very challenging binary classification problem. Different from many classification problems where the specific class label of each image is given during training, only binary information of 'same' or 'different' label for pairs of images is given; this provides less specific information than known classes - category labels. Typically, a discriminative similarity measure is learned through metric learning [5,6,13,14] for a Mahalanobis distance to map samples from the feature space into a target space. Those approaches usually require a large training set that covers large and complex data variations. They are supervised and need an expensive training procedure. Most current state-of-the-art approaches [3,10,25,27] for face verification either use additional information or functionality, such as facial component detectors and high-level classifiers, or integrate many layers of information. Moreover, in the setting of no supervision, where no training information of same/not-same is given, the performance typically drops significantly.

In this paper, we propose a sparse representation [26] based face verification method that is simple yet achieves good performance on the LFW dataset [8] without a training set (unsupervised) and in the image restricted training setting. Sparse coding [26] approximates a signal $\boldsymbol{y}$ by a linear combination of a few atoms from a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{y} \approx \boldsymbol{D}\boldsymbol{x}$, and leads to good performance in various vision applications. Sparse coding can extract stable and discriminative face representations under challenging variations. Our method measures two models of image similarity via a dictionary (reference set). The intuition of the first model is very straightforward. Since sparse representations account for most or all information of a signal (a face) with a linear combination of a small number of elementary signals (reference set) called atoms, we would expect the sparse codes of two images from the same person to be similar. So, the similarity of the sparse codes can be a measure of similarity for the image pair. The other model measures the change of the sparse code of one image from the pair to be verified when the dictionary is expanded by adding the other image from the pair. Comparing the change of

the sparse codes before and after adding the extra face image also provides a measure of similarity for the pair. We integrate these two models and the scores are fused by averaging or training an SVM.

2. Related Work

In this section, we review related work on face verification that has been evaluated on the LFW benchmark [3, 6, 7, 10, 15–17, 20, 22, 24, 25, 27]. We also review works on sparse representations, which is widely used for face identification [9, 18, 26, 28].

One natural way of addressing face verification is to design robust face descriptors that are not only discriminative but also as insensitive to variations (pose, expressions, lighting, etc) as possible. [3, 12, 15, 22]. Pinto *et al*. [15] used V1-like and Gabor filter as a face representation. In [3], an unsupervised learning-based encoding method was presented to encode the local micro-structures of a face into a set of more uniformly distributed discrete codes. In [22], Patterns of Oriented Edge Magnitudes (POEM) was proposed. The POEM is an oriented spatial multi-resolution descriptor capturing rich information (self-similarity structure) about the original image. Recently, Liang *et al*. [12] used sparse representations [26] to select a feature for person specific verification.

The similarity measure between a pair of images is a key component in face verification. Works including [6, 13, 14] focus on learning a more discriminative similarity measure for verification. Guillaumin *et al*. [6] presented two methods for learning robust distance measures: LDML and MkNN. LDML uses logistic discriminant analysis to learn a metric from a set of labeled image pairs. MkNN uses a set of labeled images to marginalize a k-nearest-neighbour (kNN) classifier for both images of a pair. Taigman *et al*. [19] applied the Information Theoretic Metric Learning (ITML) approach [5] to learn a Mahalanobis distance for face verification. The main idea of the cosine similarity metric learning (CSML) [13] is to learn a transformation matrix by minimizing the cross-validation error, where the distance measure used for optimization was cosine similarity rather than traditional Euclidean distance.

In addition, recently proposed face verification frameworks in [10, 19, 25, 27] achieve state-of-the-art accuracy in uncontrolled situations, and outperform both descriptor based methods and similarity measure based methods on the LFW dataset. Kumar *et al*. [10] presented two types of classifiers for face verification. 'Attribute Classifiers' were trained to recognize the presence or absence of a describable aspect of visual appearance. 'Simile Classifiers' were trained to recognize the similarities of parts of faces to specific reference people. Wolf *et al*. [19,25] proposed the one-shot similarity (OSS) kernel to learn discriminative models exclusive to the vectors being compared, by using a set of background samples. In [25], the OSS was extended to 'Two-Shot Similarity' (TSS) of a descriptor obtained from the ranking of images most similar to a query image. The best performance was obtained by adding SVM based OSS and TSS to LDA based OSS and TSS. Yin *et al*. [27] used extra generic identities ('memory': containing multiple images with large intra-personal variation) as a bridge and the 'associate-predict' model to handle intra-personal variation.

Most of the approaches mentioned above (especially the latter two categories) are supervised methods requiring a training set, which is referred to as the image-restricted setting in the LFW protocol. However, the training phase is burdensome and there are situations in which not providing training data is more practical. Some approaches design training-free face verification and are evaluated in the unsupervised setting on LFW dataset [16, 17]. In [16], the authors randomly selected 100 images from LFW as a reference set (without using label or pair-wise relationships of same or different) for the Borda count ranking between the Gabor Jet Descriptors. In another training-free approach, locally adaptive regression kernels (LARK) [17] were employed as visual descriptors, in conjunction with the matrix cosine similarity (MCS) measure.

Wright *et al*. [26] used sparse representations for face identification by relating the problem of finding the most similar face to a noiseless signal reconstruction. Since then, many other researchers have developed methods for face identification using sparse representations [9, 18, 28] and showed that such methods are robust to occlusion, expressions and disguise. The face identification problem is a multi-class problem naturally formulated by sparse coding since the goal of both problems is to obtain a noiseless signal reconstruction. To leverage the robustness of sparse coding for face verification, we formulate a sparse coding based face verification framework. It is, however, not trivial to extend the method to a binary classification problem of face verification.

3. Proposed Method

3.1. Overview of the Framework

The main idea to convert a multi-class classification problem into a binary one is by utilizing a set of arbitrary face images as dummy classes. With the help of the dummy classes, called *reference set*, we formulate a binary classification problem using sparse representation.

Figure 1 illustrates the proposed method. Three steps are involved: feature extraction, sparse coding and score fusion.

In feature extraction, a pair of images, A and B, are cropped and re-scaled to a fixed size. Then, feature extraction is performed to obtain the intensity (INT), HoG [2], LBP [1], and Gabor [4] features as image descriptors.

In the sparse coding step, we exploit two methods to ob-

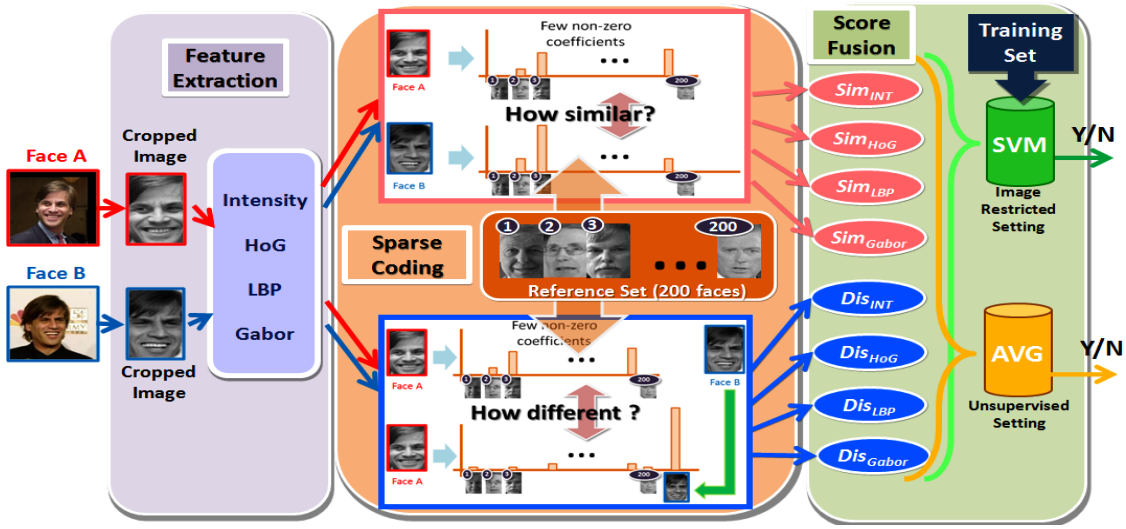


Figure 1. The proposed face verification framework based on sparse coding.

tain the sparse representations for face A and B using the fixed *reference set* as a dictionary that contains a number of, say N, faces (the reference set is chosen from the training set in the LFW protocol, and the identities in the training set are disjoint from those in the testing stage). The first method (top half of the figure) directly measures the correlation of the generated sparse codes, which we refer to as the *similarity score*. The second method (bottom half) measures the difference of the two sparse codes of face A. One is obtained on the original dictionary and the other is on an augmented dictionary by adding B to the original dictionary. Then we do the same for face B, by adding A to the original dictionary. We refer to this as the *dissimilarity score*. We compute both the *similarity score* and *dissimilarity score* for each type of feature descriptor. Sim_{INT}, Sim_{HoG}, Sim_{LBP}, and Sim_{Gabor} denote the similarity scores for each feature and Dis_{INT}, Dis_{HoG}, Dis_{LBP}, and Dis_{Gabor} denote the dissimilarity scores for each feature.

In the last stage, we fuse the eight scores obtained from different channels. We can either simply compute the average (AVG) of these eight scores in an unsupervised setting, or train an SVM to reduce the effect of overfitting to a particular score in a supervised setting.

3.2. Feature Extraction

After cropping and resizing the faces, each sample is decomposed into blocks and then a set of low-level feature descriptors is extracted from each block. The feature extraction methods capture information related to shape (histogram of oriented gradients (HOG)), texture (captured by local binary patterns (LBP)), color information (intensity) and salient visual properties (captured by Gabor filters).

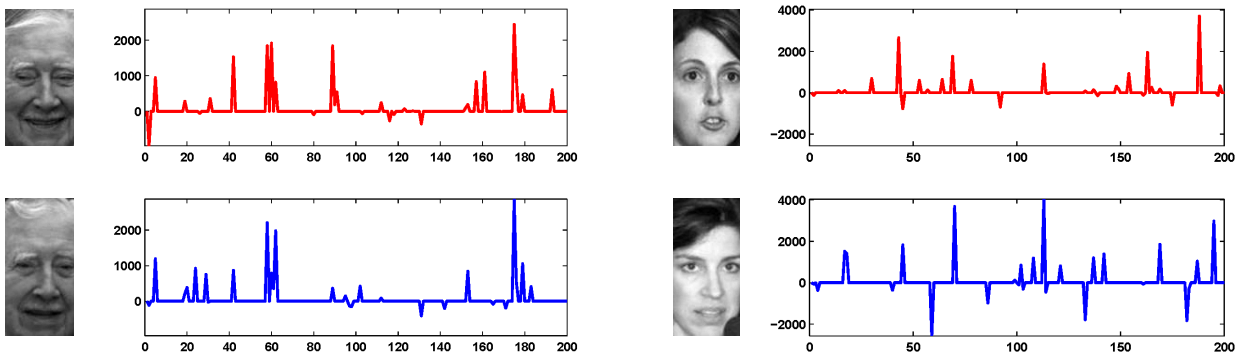
3.3. Sparse Representation

A sparse representation-based face recognition algorithm was proposed in [26] and demonstrated to have high performance on the face identification task. Given a dictionary $\boldsymbol{D} = \{\boldsymbol{d}_1, \boldsymbol{d}_2, ..., \boldsymbol{d}_N\}$ where $\boldsymbol{d}_i$ is the i-th dictionary atom (l_2-normalized) and a test sample $\boldsymbol{y}$, the sparse code of $\boldsymbol{y}$, $\hat{\boldsymbol{x}}$, can be obtained by solving the following l_1-minimization problem,

$$\hat{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \qquad (1)$$

Sparse representation is an intuitively appealing method for face identification. The dictionary typically contains multiple face images for each person to be subsequently recognized. However, it is not straightforward to be directly applied to face verification since verification is not a multi-class problem that can be solved by choosing a few atoms from the dictionary. In face verification, a similarity measure is typically learned from pairs of training images labeled 'same' or 'different'. This provides less specific information than known identities - image labels.

We instead use sparse representation for face verification problem in two different ways via a reference set, which we use as a dictionary: similarity score of two sparse codes and dissimilarity score of two sparse codes.



(a) (b) SimScore = 0.7534 (c) (d) SimScore = -0.0038

Figure 2. An example of sparse codes (intensity feature) for 'similarity score' denoted by SimScore. (a) Original faces of a 'same' pair. (b) Sparse codes for the 'same' pair. (c) Original faces of a 'different' pair. (d) Sparse codes for the 'different' pair.

3.3.1 Similarity Score of Two Sparse Codes

A reference set is a set of images randomly selected from an image pool (e.g. training images) whose identities never appear in the test stage. We use the reference set as a dictionary $\boldsymbol{D}$ of size N to reconstruct the input image pairs (A,B). The feature vectors from the same individual are usually similar and more likely to have similar corresponding reconstructed signals by sparse coding, *i.e.* linear combination of dictionary atoms. We are interested in measuring the similarity of the sparse codes of A and B that approximates the similarity of the input images and let

$$\begin{aligned} \hat{\boldsymbol{x}}_A^N &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_A - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \\ \hat{\boldsymbol{x}}_B^N &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_B - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \end{aligned} \tag{2}$$

be the sparse codes of A and B, respectively. Here, $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$ are feature vectors of input faces A and B respectively, $\boldsymbol{D}$ is the given dictionary, and γ is a penalty weight on sparsity. We define the 'similarity score' of $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$, SimScore, by utilizing a similarity metric of $\hat{\boldsymbol{x}}_A^N$ and $\hat{\boldsymbol{x}}_B^N$,

$$\text{SimScore}(\boldsymbol{y}_A, \boldsymbol{y}_B) := \text{Similarity}(\hat{\boldsymbol{x}}_A^N, \hat{\boldsymbol{x}}_B^N) \tag{3}$$

We use the cosine similarity (CS) [13] as the similarity metric between two sparse codes. The CS of two vectors is defined as:

$$CS(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T\boldsymbol{y}}{\|\boldsymbol{x}\|\|\boldsymbol{y}\|} \tag{4}$$

Given a pair of feature vectors $(\boldsymbol{y}_A, \boldsymbol{y}_B)$, the 'similarity scores' (SimScore) of their sparse codes with the reference set from different feature channels are computed as:

$$\begin{aligned} Sim_{INT} &= CS(\hat{\boldsymbol{x}}_A^{N,INT}, \hat{\boldsymbol{x}}_B^{N,INT}) \\ Sim_{HoG} &= CS(\hat{\boldsymbol{x}}_A^{N,HoG}, \hat{\boldsymbol{x}}_B^{N,HoG}) \\ Sim_{LBP} &= CS(\hat{\boldsymbol{x}}_A^{N,LBP}, \hat{\boldsymbol{x}}_B^{N,LBP}) \\ Sim_{Gabor} &= CS(\hat{\boldsymbol{x}}_A^{N,Gabor}, \hat{\boldsymbol{x}}_B^{N,Gabor}) \end{aligned} \tag{5}$$

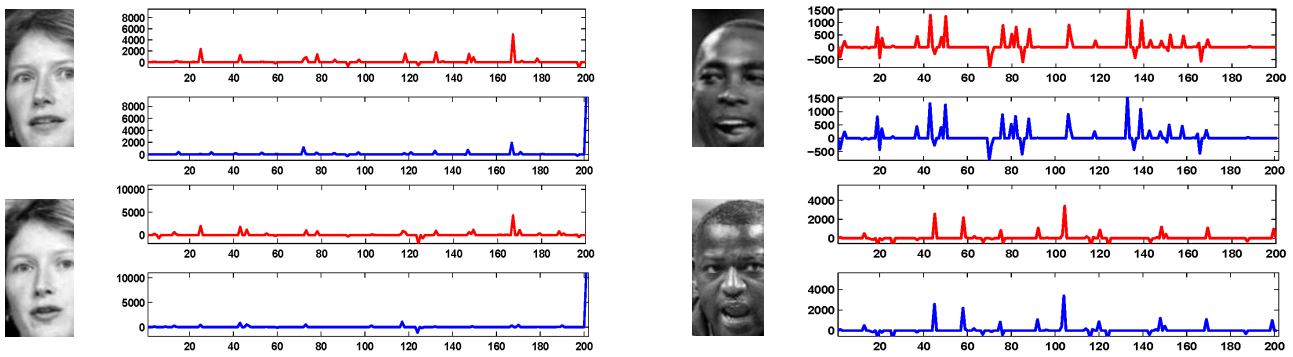
where $\hat{\boldsymbol{x}}_k^{N,feat}$ denotes the sparse code obtained from Eq. (2) using a dictionary with N atoms and $feat$ feature for face k, *e.g.*, $\hat{\boldsymbol{x}}_A^{N,INT}$ represents the N dimensional sparse codes with respect to the N dictionary atoms computed from the intensity features of face A.

Figure 2-(a) and (b) show an example of a pair of faces from the same individual (with slight expression change) and their corresponding sparse codes generated from a dictionary with N=200 atoms using intensity(INT). Figure 2-(c) and (d) show a pair of faces from different individuals and their corresponding sparse codes generated from the intensity. It can be seen that sparse codes from the same individual (left) have much higher correlation (the responses to the 200 dictionary atoms have similar trend) than the sparse codes of the pair from different individuals (right).

3.3.2 Dissimilarity Score of Two Sparse Codes

Looking at only the similarity of the sparse codes is not making full use of the power of sparse coding. In face identification via sparse representation [26], the test face (probe) is represented as a sparse linear combination of the dictionary atoms. The coefficient of the most similar face in the dictionary to the test face is high while other coefficients are small or zero. We take advantage of this principle of the sparse coding in the following way.

For notation consistency, $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$ are feature vectors of input faces A and B respectively, $\boldsymbol{D}$ is the given original dictionary. We first compute the sparse coefficients of face



(a) (b) DisScore = 0.2653 (c) (d) DisScore = 5.56×10^{-17}

Figure 3. An example of sparse codes (intensity feature) for 'dissimilarity score' denoted by DisScore. (a) Original faces of a 'same' pair. (b) Sparse codes with and without adding the other face to dictionary for the 'same' pair. (c) Original faces of a 'different' pair. (d) Sparse codes with and without adding the other face to dictionary for the 'different' pair. **Note that the range of horizontal axes of blue plots is [1,201] while that of red plots is [1,200] and the scales of vertical axes of two sparse codes for an image are consistent for comparison.** In the blue plots, one can observe the peak at 201 in (b) but not in (d). Also note that SimScore of pair (a) is 0.8551 and SimScore of pair (c) is 0.0471.

A, $\hat{\boldsymbol{x}}_A^N$, using dictionary $\boldsymbol{D}$. Next, we add the l_2-normalized feature vector of face B, $\overline{\boldsymbol{y}_B}$, to the dictionary to construct a new augmented dictionary $\widetilde{\boldsymbol{D}}_B = [\boldsymbol{D}|\overline{\boldsymbol{y}_B}]$, of size $N+1$ and obtain another sparse code $\widetilde{\boldsymbol{x}}_A^{N+1}$ from the new dictionary $\widetilde{\boldsymbol{D}}_B$,

$$\begin{aligned} \hat{\boldsymbol{x}}_A^N &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_A - \boldsymbol{D}\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1 \\ \widetilde{\boldsymbol{x}}_A^{N+1} &= \arg\min_{\boldsymbol{x}} \|\boldsymbol{y}_A - \widetilde{\boldsymbol{D}}_B\boldsymbol{x}\|^2 + \gamma\|\boldsymbol{x}\|_1. \end{aligned} \tag{6}$$

Similarly, we can construct the augmented dictionary for face B, $\widetilde{\boldsymbol{D}}_A = [\boldsymbol{D}|\overline{\boldsymbol{y}_A}]$, by adding the l_2-normalized feature vector of face A to the original dictionary. Two sparse codes $\hat{\boldsymbol{x}}_B^N$ and $\widetilde{\boldsymbol{x}}_B^{N+1}$ are computed using the original dictionary $\boldsymbol{D}$ and the augmented dictionary $\widetilde{\boldsymbol{D}}_A$, respectively.

The motivation is, if two images are from the same individual, the $N+1$-th coefficient in the augmented dictionary will have a significantly high value and other coefficients will be diminished compared to the code obtained with the original dictionary. In contrast, when the two images are not from the same individual, the coefficients with respect to the original dictionary and the augmented dictionary do not significantly differ from each other. Thus, a higher dissimilarity of the two sparse codes obtained from the original dictionary and the augmented dictionary indicates a higher similarity of the pair being compared.

We compute the dissimilarity of the two sparse codes of face A before and after adding face B to the dictionary as follows,

$$\text{Dy}(\boldsymbol{y}_A) = 1 - \text{Similarity}(\hat{\boldsymbol{x}}_A^N, \widetilde{\boldsymbol{x}}_A^{N+1}(1{:}\text{N})) \tag{7}$$

Note that $\text{Dy}(\cdot)$ is defined on a single image in a given pair, whereas $\text{Similarity}(\cdot,\cdot)$ is defined with respect to two sparse codes. We can also obtain $\text{Dy}(\boldsymbol{y}_B)$, exchanging A and B. By averaging $\text{Dy}(\boldsymbol{y}_A)$ and $\text{Dy}(\boldsymbol{y}_B)$, we obtain the 'dissimilarity score' of $\boldsymbol{y}_A$ and $\boldsymbol{y}_B$, DisScore,

$$\text{DisScore}(\boldsymbol{y}_A, \boldsymbol{y}_B) := \frac{\text{Dy}(\boldsymbol{y}_A) + \text{Dy}(\boldsymbol{y}_B)}{2} \tag{8}$$

The higher the score, the more similar the pair is.

Figure 3-(a) and (b) show a pair of faces from the same individual and their corresponding sparse codes before (red) and after (blue) adding the other to the dictionary. Figure 3-(c) and (d) show a pair of faces from different individuals and their corresponding sparse codes before and after adding the other to the dictionary. We can observe that the sparse codes from the same individual (left) shows significant difference in the first N atoms than the pair from different individuals (right).

As done for the similarity scores, we compute dissimilarity scores for four feature channels of intensity, HoG, LBP and Gabor to obtain Dis_{INT}, Dis_{HoG}, Dis_{LBP} and Dis_{Gabor}, respectively.

3.4. Score Fusion

Each feature descriptor and scoring method contains different discriminative power and should be aggregated in a reasonable way. According to [3, 19, 23, 25], combining multiple similarities from different descriptors usually boosts performance. We consider two simple approaches for fusing the eight scores (four feature channels $\times$ two scoring methods).

In the unsupervised setting, we simply average the eight scores from different feature channels to obtain the final similarity score of the given pair. The averaging weighs every score equally. For the image restricted setting, we can fuse the scores by training a linear SVM to obtain more discriminative weights on each score using the given training set.

4. Experimental Results

We evaluate the proposed algorithm on the LFW dataset and compare the results with previous approaches.

4.1. LFW Dataset

The Labeled Faces in the Wild (LFW) dataset was recently introduced as a benchmark for face verification in unconstrained environments [8]. Real-world images in the LFW dataset exhibit visual variations due to pose, facial appearance, age, lighting, expression, occlusion, scale, camera, misalignment, hairstyle, *etc*. Figure 4 shows some examples of image pairs, each pair corresponding to a single subject that differ in (partial) occlusion, lighting, facial expression and pose.

Figure 4. Some example images from the LFW dataset with variations in: Top row: (left) partial occlusion, (right) lighting and occlusion; bottom row: (left) expression, (right) expression and pose. Each corner shows a different subject. Note that each pair is from the same person.

The dataset comes with a division of 10 splits/folds (disjoint subject identities) for cross validation with three paradigms of evaluation protocols: unsupervised, image-restricted, and image-unrestricted protocols [8]. In the **unsupervised** protocol, there is no training information of same/not-same labels. It is the most challenging due to lack of training samples. The **image-restricted** protocol refers to the setting of using only the restricted number of given image pairs for training. In this setting, it is known whether an image pair belongs to the same person or not, while identity information of each image is not provided. The **unrestricted** protocol refers to the training setting that can use all available data, including the identity of the people in the images that allows one to generate as many training pairs as possible. The latter two settings allow us to utilize available image pair information in the training set. In this paper, we only focus on the first two protocols. The aligned version, lfw-a, was used in all experiments.

In our evaluations, for each fold, we randomly choose N=200 images (one image per individual) to construct a compact dictionary (reference set) from the training set without using their pair information. We have empirically tried varying dictionary size N from 200 to 500, and found that the size has only slight impact on the verification performance. For efficiency, we use a fixed size N=200 in the following experiments to report our result.

4.2. Experimental setup

To obtain a sparse solution to the least squares problem, we can choose either l_0 regularization or l_1 regularization in the least squares objective function (Eq.1). We choose the l_1 regularizer since it is hard to specify the number of nonzero coefficients, i.e., the hyper-parameter of the l_0 regularizer. We use the implementation of Lee *et al*. [11] due to its computational efficiency.

For the feature extraction step, we do not apply any photometric pre-processing. All the faces are cropped and rescaled to 80×148. For extracting HoG and LBP features, we divide each face into blocks of 20×20 size and extract 16-bin HoG feature and 59-bin uniform LBP feature for each block. For Gabor feature, we adopt five scales and eight orientations of the Gabor filters. The final Gabor feature vector is obtained by concatenating the responses at every five pixels in order to reduce the dimensionality of the feature vector to a manageable size.

4.3. Results from Different Feature Descriptors and Score Fusions

The performances of our method with individual feature and their fusion are shown in Table 1 (on fold 1 only). The first column shows the verification accuracy obtained by using the Euclidean distance of the original feature vector pairs as similarity measure. The second column shows the verification accuracy from the SimScore (Eq.3). The third column is from the DisScore (Eq.8). Both SimScore and DisScore for individual feature descriptors achieve significant improvements over the Euclidean distance. The 'Combined' scores are the results obtained by fusing the scores from all the four features by averaging (no training) or creating a vector of four scores and running an SVM on this vector. The **HybridSparse** scores are obtained by fusing the eight scores from both SimScore and DisScore. We can see that the **HybridSparse (Avg)**, obtained by simply averaging the eight scores with equal weight, achieves good verification accuracy (83.00%) and the **HybridSparse (SVM)** boosts the performance further to 84.67%. Generally, as we expect, score fusion can always achieve better result (as in [3, 10, 19, 23, 25, 27]) since there could be complimentary information across different scores.

Table 1. Verification accuracy at Equal Error Rate on LFW dataset (fold 1 only) under different similarity measures.

Descriptor	Euclidean	SimScore	DisScore
Intensity	0.7133	0.7533	0.7633
HoG	0.6767	0.7733	0.7467
LBP	0.6700	0.7633	0.7667
Gabor	0.6933	0.7700	0.7533
Combined (Avg)	0.7067	0.8167	0.8033
Combined (SVM)	0.7267	0.8333	0.7967
HybridSparse (Avg)	N/A	**0.8300**	
HybridSparse (SVM)	N/A	**0.8467**	

Table 2. Mean (± standard error) verification accuracy on the LFW dataset (Unsupervised protocol).

Method	Accuracy
H-XS-40 [16]	0.6945±0.0048
GJD-BC-100 [16]	0.6847±0.0065
SD-MATCHES [16]	0.6410±0.0042
LARK [17]	0.7223±0.0049
HybridSparse (Avg)	**0.8377±0.0053**
HybridSparse (Avg, flip)	**0.8470±0.0047**

4.4. Comparison with the State-of-the-art Methods

Comparison on the Unsupervised protocol Our method can be compared with other methods using the unsupervised protocol, since we simply sample a very small number of images from the training set for the reference set without using any pair labels of same/different or identity information. Table 2 shows the comparison result at equal error rate. The 'flip' means that when comparing image pair A and B, we also compare A and the horizontally flipped image of B to reduce the effect of pose variation. Then the average of the two scores is taken as the final similarity score. Figure 5 presents the ROC curve of our approach (dotted red line), along with the ROC curves of previous methods. As shown, our approach significantly outperforms the other methods by a very large margin.

Comparison on the Image-Restricted protocol Table 3 shows the face verification accuracy of our method in comparison with state-of-the-art methods under the Image-Restricted protocol that allows using the training set with labels of same/different. Figure 6 shows the ROC curve of our approach (dotted red line), along with the ROC curves of selected recent state-of-the-art methods.

The results show that the verification accuracy of our approach is competitive to the state-of-the-art methods on the LFW benchmark in the challenging image-restricted protocol. It is worth noting that, methods marked by '$*$' (such as [3, 10, 27]) use training data outside of the LFW for facial point detection or pose/illumination classification and so on, which can have a significant impact on the verification accuracy, thus are not directly comparable to other methods including ours. Kumar *et al*. [10] achieves excellent results (however still marginally lower than ours) at the expense of an expensive training of high-level classifiers by incorporating a huge volume of images outside of the LFW dataset. The LE method [3] relies on facial feature point detectors. Predict-Associate [27] not only relies on facial feature point detectors, but also uses the Multi-PIE dataset with identities covering 7 poses and 4 illumination conditions. For other methods that we are in the same category with, [25] is the most comparable. Wolf *et al*. [25] also combines multiple

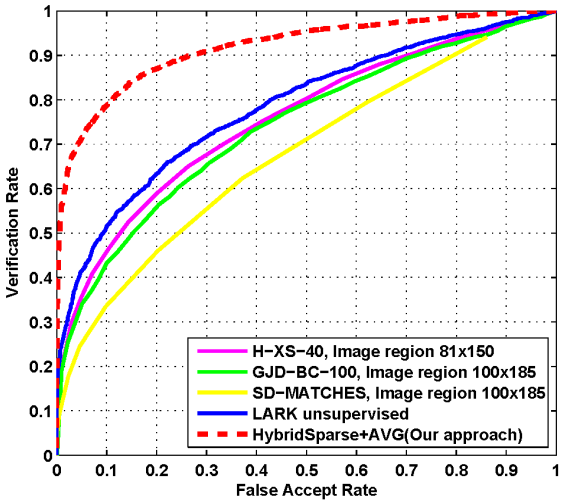


Figure 5. ROC curves on the LFW dataset (unsupervised protocol).

Table 3. Mean (± standard error) verification accuracy on the LFW dataset (Image-Restricted protocol). '$*$' denotes methods using outside training data.

Method	Accuracy
LDML, funneled [6]	0.7927±0.0060
POEM [22]	0.7542 ±0.0071
Hybrid [19]	0.8398±0.0035
Combined b/g samples based [25]	0.8683±0.0034
*Attribute and Simile classifiers [10]	0.8529± 0.0123
Single LE + holistic [3]	0.8122±0.0053
*Multiple LE + comp [3]	0.8445±0.0046
*Associate Predict [27]	0.9057 ±0.0056
LARK+OSS [17]	0.8512 ±0.0037
HybridSparse (SVM)	**0.8530 ±0.0040**
HybridSparse (SVM, flip)	**0.8624 ±0.0031**

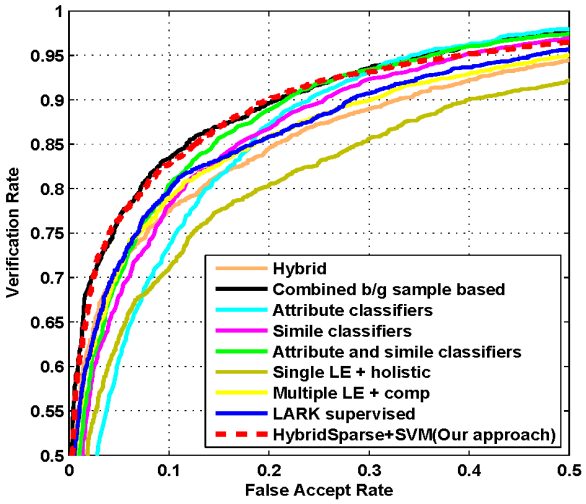


Figure 6. ROC curves on the LFW dataset (Image-Restricted protocol). Only shown with the selected **best** results that were recently reported for clarity.

descriptors, however, their method has complicated layers and leverages metric learning [27]. An additional disadvantage of this method is that it requires background samples (a fixed set of 'negative' examples) that have similar properties as the faces being compared. The background samples should not contain faces from any person who might subsequently appear in a pair to be compared. Overall, our simple approach achieves competitive accuracy without local feature detection or other additional information.

5. Conclusions and Future Work

We have proposed a novel approach for face verification using sparse coding in two different yet complimentary ways with a fixed reference set as a dictionary. The evaluation on the very challenging LFW dataset both under the unsupervised setting and image restricted training setting shows competitive results. We demonstrated that sparse coding can be a promising direction for face verification since it extracts more stable and discriminative face representation under challenging variations. As a future work, we would explore pairwise dictionary learning for face verification applications.

6. Acknowledgements

This work was partly supported by MURI from the Office of Naval Research under the Grant N00014-08-I-0638.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face Description with Local Binary Patterns: Application to Face Recognition. *IEEE T. PAMI*, 28(12):2037–2041, 2006. 2

[2] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol. Face Recognition Using HOG-EBGM. *Pattern Recognition Letters*, 29(10):1537–1543, 2008. 2

[3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *CVPR*, pages 2707–2714, 2010. 1, 2, 5, 6, 7

[4] J. G. Daugman. Uncertainty Relation for Resolution in Space, Spatial Frequency, and Orientation Optimized by Two-Dimensional Visual Cortical Filters. *Journal of the Optical Society of America A*, 2:1160–1169, 1985. 2

[5] J. Davis, B. Kulis, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *ICML*, 2007. 1, 2

[6] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *ICCV*, pages 498 –505, 2009. 1, 2, 7

[7] H. Guo, W. R. Schwartz, and L. S. Davis. Face Verification using Large Feature Sets and One Shot Similarity. In *International Joint Conference on Biometrics*, 2011. 2

[8] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 1, 6

[9] Z. Jiang, Z. Lin, and L. S. Davis. Learning a Discriminative Dictionary for Sparse Coding via Label Consistent K-SVD. In *CVPR*, 2011. 2

[10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009. 1, 2, 6, 7

[11] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, pages 801–808, 2007. 6

[12] Y. Liang, S. Liao, L. Wang, and B. Zou. Exploring Regularized Feature Selection for Person Specific Face Verification. In *ICCV*, 2011. 2

[13] H. V. Nguyen and L. Bai. Cosine Similarity Metric Learning for Face Verification. In *ACCV*, 2010. 1, 2, 4

[14] E. Nowak. Learning visual similarity measures for comparing never seen objects. In *CVPR*, 2007. 1, 2

[15] N. Pinto, J. DiCarlo, and D. Cox. How far can you get with a modern face recognition test set using only simple features? In *CVPR'09*. 2

[16] J. Ruiz-del Solar, R. Verschae, and M. Correa. Recognition of faces in unconstrained environments: a comparative study. *EURASIP J. Adv. Signal Process*, 2009. 2, 7

[17] H. J. Seo and P. Milanfar. Face verification using the lark representation. In *IEEE Transactions on Information Forensics and Security*, 2011. 2, 7

[18] I. W.-H. T. Shenghua Gao and L.-T. Chia. Kernel sparse representation for image classification and face recognition. In *ECCV'10*. 2

[19] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *BMVC*, 2009. 2, 5, 6, 7

[20] X. Tan and B. Triggs. Fusing Gabor and LBP feature sets for kernel-based face recognition. In *AMFG'07*, pages 235–249, 2007. 2

[21] A. Tolba, A. El-Baz, and A. El-Harby. Face recognition: A literature review. *International Journal of Signal Processing*, 2, 2006. 1

[22] N.-S. Vu and A. Caplier. Face recognition with patterns of oriented edge magnitudes. In *ECCV*, 2010. 2, 7

[23] L. Wolf, T. Hassner, and Y. Taigman. Descriptor based methods in the wild. In *Workshop in ECCV. (2008)*, 2008. 5, 6

[24] L. Wolf, T. Hassner, and Y. Taigman. The one-shot similarity kernel. In *ICCV'09*, Sept. 2009. 2

[25] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *ACCV*, 2009. 1, 2, 5, 6, 7

[26] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust Face Recognition via Sparse Representation. *IEEE Trans. PAMI*, 31(2):210–227, 2009. 1, 2, 3, 4

[27] Q. Yin, X. Tang, and J. Sun. An associate-predict model for face recognition. In *CVPR*, 2011. 1, 2, 6, 7, 8

[28] Q. Zhang and B. Li. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2010. 2

[29] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4), 2003. 1