Learning Matrix Space Image Representations

Anand Rangarajan[1]

Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611-6120, US
anand@cise.ufl.edu

Abstract. When we seek to directly learn basis functions from natural scenes, we are confronted with the problem of simultaneous estimation of these basis functions and the coefficients of each image (when projected onto that basis). In this work, we are mainly interested in learning matrix space basis functions and the projection coefficients from a set of natural images. We cast this problem in a joint optimization framework. The Frobenius norm is used to express the distance between a natural image and its matrix space reconstruction. An alternating algorithm is derived to simultaneously solve for the basis vectors and the projection coefficients. Since our fundamental goal is classification and indexing, we develop a matrix space distance measure between images in the training set. Results are shown on face images and natural scenes.

1 Introduction

In recent years, there has been a lot of progress in the mathematical representation of natural scenes [7]. Once it became clear that coding methods using the discrete cosine transform (DCT) [9], Gabor expansions [4], wavelets [3], etc. were quite successful in generating compact codes for images, there was an increased interest in directly *learning* such bases from natural images. Directly learning the basis vectors presents a more challenging computational problem than the usual coding case, because in the former, *both* the bases and the coefficients have to be estimated from the natural images.

The most comprehensive work in this general area of simultaneous estimation of both basis vectors and coefficients is the work of Olshausen and his collaborators [7, 6, 8]. In this body or work, the main interest is in learning compact and (usually) *overcomplete* representations of natural scenes. In this approach, a joint optimization problem is typically constructed on both the basis vectors and the coefficients. A regularization prior is always added to enforce prior knowledge of *sparsity* of the coefficients. Recently, this work has evolved toward adding a mixture of Gaussians prior [8] in order to enforce the constraints of sparsity and distributivity of the coefficients.

In this paper, we are mainly interested in recasting this problem of simultaneous estimation of bases and coefficients in a matrix space. We wish to point out that matrix space representations of natural images has been mostly ignored

despite the fact that it is mostly common knowledge that the singular value decomposition (SVD) [5] is sometimes a good choice for image representation and compression [10]. However, all the previous work (that we are aware of) on SVD-based image representation is focused on single images. There is no previous work on estimating *common matrix space bases* from a set of natural images. To state the goal of this paper, we wish to simultaneously learn a matrix basis set and the projection coefficients of each image. Given a set of 2D grayscale natural images, we represent each of them as matrics and then formulate the learning problem as a *joint optimization* of matrix basis vectors and projection coefficients.

In contrast to earlier work by Olshausen and his collaborators [7], our primary motivation is not to develop a compact *code* based on the statistics of natural scenes. Instead, we are more interested in common matrix space representations for the purposes of recognition, classification and indexing. That is, we believe that matrix space representations can be used to develop new matrix space distance measures for the purposes of classification and indexing. Some of the experimental results presented in the paper aim toward this goal.

2 Grayscale Intensity Images as Matrices

As mentioned in the Introduction, we are mainly motivated by the overall success of image coding. Our principal aim in this paper is to learn a common matrix space representation from a set of images without any *a priori* knowledge of the basis set or of the projection coefficients. The fundamental idea is to express each grayscale intensity image as a matrix and then investigate the extent to which compact coding allows us to extract a common matrix basis from the set of pregiven images. Since coding works and works well, we expect that each image is expressible by a compact set of basis vectors and coefficients. For example, it is well known though not so widely exploited that the SVD representation of a 2D grayscale intensity image has a spectrum that rapidly falls to zero [10].

Denote by X, the matrix corresponding to a grayscale intensity image. The SVD of the image is given by

$$X = USV^T \tag{1}$$

where $U^TU = V^TV = I$. Since the image matrix need not be square, note that the dimensions of the U and V matrices can be different. Reconstructions of the original image using a set of D components can be written as

$$\hat{X} = \sum_{i=1}^{D} s(i,i)u_iv_i^T \tag{2}$$

where u_i and v_i are the ith eigenvectors.

In Figures 1 and 2, we show the spectrum of the well known Lenna image and the SVD-based reconstructions respectively. The first 5 SVD component images (computed as $u_iv_i^T$) are also shown in Figure 2. It is evident from both the

spectrum plots and the component images that the spatial frequencies tend to vary in inverse proportion to the SVD coefficients; higher the spatial frequencies in a component image, the smaller the SVD coefficient and vice versa. The rapidity with which the SVD coefficients fall off is evident from Figure 2.

3 Learning a Common Reference Frame

We now embark upon the formulation of the problem. Since we wish to represent 2D grayscale intensity images as matrices, the technical challenge is one of finding a common eigen reference basis set of vectors to represent all of the images in the chosen set. Since we have decided to use a matrix space representation for the non-square images, the common eigen reference frame consists of a set of row eigen vectors, each of which has a corresponding counterpart in the set of column eigen vectors. Insofar as only the eigen reference frame is deemed common, the projections of each image in the chosen set onto the basis will be different *and* unknown *a priori*. Consequently, not only do we have to learn the common eigen reference frame, in addition, we have to compute the projections of each image onto the reference in order to determine the representation error.

Let the set $\{X_k,\ k \in \{1,\ldots,K\}\}$ denote the collection of images. The common eigen reference frame is denoted by the pair (U,V) which is intended to invoke the SVD association. The projection of image X_k onto the basis set is denoted by Λ_k with the set $\{\Lambda_k,\ k \in \{1,\ldots,K\}\}$ denoting the set of projections. Each image X_k is of size $M \times N$ and the sizes of the basis sets U and V are $M \times D$ and $N \times D$ respectively. The size of each Λ_k is $D \times D$. Note that each $\Lambda_k,\ k \in \{1,\ldots,K\}$ is *diagonal* since it is a projection onto the basis set. Inequalities, such as $D < \min(M,N)$ hold as a consequence of orthogonality conditions on U and V.

The central technical contribution of this paper is a mathematical unpacking of the following intuition: A reasonable criterion for learning the basis set $(\hat{U},\hat{V})$ is to minimize the representation error of each image X_k in the set of K images when projected onto the *space* of basis sets (U,V). This criterion is somewhat complicated by the fact that the projections of the images onto the basis set are themselves computed after the basis set is fixed.

We use the Frobenius norm $||A||_F^2 = \sum_{ij} |a_{ij}|^2$ to mathematically characterize the representation error [5]. This norm is chosen because of its close relationship to the matrix spectra. The central cost function used in this paper is

$$(\hat{U},\hat{V}) = \min_{(U,V)} E_{\text{matrixbasis}}(U,V) = \min_{(U,V)} \sum_{k=1}^{K} ||X_k - U\Lambda_k V^T||_F^2 \tag{3}$$

with

$$\Lambda_k = \text{diag}\ (U^T X_k V),\ k \in \{1,\ldots,K\} \tag{4}$$

and the constraints

$$U^T U = I_D \text{ and } V^T V = I_D. \tag{5}$$

Equation (4) relates the projection Λ_k with the basis set (U, V). The **diag** operator emphasizes the fact that each Λ_k is diagonal. In this paper, we have elected not to treat the projections as quasi-independent variables. Following [8], it is certainly possible to treat Λ_k as a separate variable and associate a Bayesian regularization prior with it. We have not done so for reasons of simplicity at this preliminary stage. The constraints in (5) express the fact that U and V are orthonormal which in sync with their treatment as SVD-like bases. Note that we have also not chosen to enforce the constraint that each diagonal entry in Λ_k is strictly positive. For $U\Lambda_k V^T$ to be linked to an SVD representation, this constraint has to be active. However, we merely treat the reference basis as "SVD-like" in this paper. This constraint can also be enforced as a Bayesian prior if needed.

We now derive an alternating algorithm to minimize the energy function in (3). First, we enforce the orthogonality constraints in (5) using Lagrange parameters [1]. The Frobenius norm is also rewritten using matrix operators.

$$E_{\text{matrixbasis}}(U, V, \mu, \nu) = \sum_{k=1}^{K} \text{trace}\,[(X_k - U\Lambda_k V^T)^T (X_k - U\Lambda_k V^T)] \\ + \text{trace}\,[\mu(U^T U - I_D)] + \text{trace}\,[\nu(V^T V - I_D)] \quad (6)$$

with the understanding that $\Lambda_k = \text{diag}\,(U^T X_k V)$. The Lagrange parameter matrices μ and ν are both *symmetric* [1] $D \times D$ matrices. This energy function can be further transformed by dropping terms not involving the basis set (U, V) and by enforcing the orthogonality constraints in the matrix representation error norm.

$$E_{\text{matrixbasis}}(U, V, \mu, \nu) = -2 \sum_{k=1}^{K} \text{trace}\,[X_k^T U \Lambda_k V^T)] \\ + \text{trace}\,[\mu(U^T U - I_D)] + \text{trace}\,[\nu(V^T V - I_D)] \quad (7)$$

To derive the alternating algorithm, we first hold V fixed and solve for U in (7). Then we alternate by solving for V given U. Once U and V are updated, Λ_k is updated $\forall k \in \{1, \ldots, K\}$ in order to be in lockstep with the basis set (U, V). Differentiating (7) w.r.t. U and setting the result to zero, we get

$$\sum_{k=1}^{K} X_k V \Lambda_k = \hat{U}\mu \Rightarrow \hat{U} = \sum_{k=1}^{K} X_k V \Lambda_k \mu^{-1}. \quad (8)$$

Enforcing the orthogonality constraint for U, (8) is transformed into

$$\hat{U}^T \hat{U} = \mu^{-1} \left(\sum_{k=1}^{K} X_k V \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k V \Lambda_k \right) \mu^{-1} = I_D$$

$$\Rightarrow \mu^2 = \left(\sum_{k=1}^{K} X_k V \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k V \Lambda_k \right)$$

$$\Rightarrow \mu = \left[\left(\sum_{k=1}^{K} X_k V \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k V \Lambda_k \right) \right]^{\frac{1}{2}}$$

$$\Rightarrow \hat{U} = \sum_{k=1}^{K} X_k V \Lambda_k \left[\left(\sum_{k=1}^{K} X_k V \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k V \Lambda_k \right) \right]^{-\frac{1}{2}}. \quad (9)$$

The above solution for U is obtained relative to a fixed V. Completely analogous to (8) and (9), we can solve for V relative to a fixed U.

$$\hat{V} = \sum_{k=1}^{K} X_k^T U \Lambda_k \left[\left(\sum_{k=1}^{K} X_k^T U \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k^T U \Lambda_k \right) \right]^{-\frac{1}{2}}. \quad (10)$$

Once we've obtained a candidate basis set (U, V), we keep each Λ_k in lockstep with the basis by setting

$$\Lambda_k = \text{diag} \left(\hat{U}^T X_k \hat{V} \right). \quad (11)$$

The overall algorithm is summarized below.

Learning Matrix Representations

Initialize U, V to random $M \times D$ and $N \times D$ matrices respectively.
Begin A: Do A until $\Delta E < \Delta E_{\text{thr}}$.
$U^{\text{old}} = U$, $V^{\text{old}} = V$, $\Lambda_k^{\text{old}} = \Lambda_k$, $\forall k \in \{1, \ldots, K\}$.
$\Lambda_k = \text{diag} \left(U^T X_k V \right)$, $\forall k \in \{1, \ldots, K\}$.
$U = \sum_{k=1}^{K} X_k V \Lambda_k \left[\left(\sum_{k=1}^{K} X_k V \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k V \Lambda_k \right) \right]^{-\frac{1}{2}}$.
$\Lambda_k = \text{diag} \left(U^T X_k V \right)$, $\forall k \in \{1, \ldots, K\}$.
$V = \sum_{k=1}^{K} X_k^T U \Lambda_k \left[\left(\sum_{k=1}^{K} X_k^T U \Lambda_k \right)^T \left(\sum_{k=1}^{K} X_k^T U \Lambda_k \right) \right]^{-\frac{1}{2}}$.
$\Delta E = \sum_{k=1}^{K} \left(||X_k - U^{\text{old}} \Lambda_k^{\text{old}} (V^{\text{old}})^T||_F^2 - ||X_k - U \Lambda_k V^T||_F^2 \right)$.
End A

A theoretical proof showing that the energy in (6) decreases in each step is not easily forthcoming. This is due to the obvious heuristic device we have employed for updating each Λ_k. In contrast, showing that the energy decreases due to the U and V updates is relatively straightforward due to the updates being constrained least-squares solutions. In all experiments, we have merely executed the above algorithm until an iteration cap is exceeded. We have not encountered any stability problems when executing the above algorithm. This is quite surprising as one would have expected numerical errors to create problems when taking the relevant matrix inverses and square roots. As mentioned in the next section, no effort was made to implement the matrix inverses and square roots in a computationally efficient manner. The matrix routines in Matlab$^{\text{TM}}$ 5.3 were used in all computations.

4 Results

Two face images are shown in Figure 3. The images have been registered using a non-rigid registration method [2]. Both images are originally 330 × 228 and have been down sampled to 150 × 114.

In Figures 4 through 8, we present the results of learning a common matrix space reference frame beginning with two components and moving up to 100 components. In each figure, the reconstructed faces are presented along with the projection coefficients [diag($\hat{U}^T X_k \hat{V}$)]. The first reconstruction with two components turned out to be identical for both faces, i.e.

$$\hat{U}\Lambda_1\hat{V}^T = \hat{U}\Lambda_2\hat{V}^T.$$

In the reconstructions (which are identical), the hair texture evident in face 1 is clearly discernible. In addition, the reconstruction has the appearance of a fuzzy face. As the number of basis components are increased, the quality of the reconstruction improves as can be seen in the progression from Figure 4 to Figure 8.

Examine the projection bar charts in Figure 4, Figure 5 and Figure 6. The bar charts are a visualization of the projection coefficients. Note the extensive similarities in the projection coefficients. They start out identical in Figure 4 and continue to be strongly related as can be seen in Figure 6. We return to this issue later.

After reconstructing faces 1 and 2 using 12 basis components, in Figure 9 we visualize six leading common components. This was done by only visualizing those components for which the projections in *both* face 1 and face 2 were large. Each image shown in Figure 9 corresponds to a component $u_k v_k^T$ where k is the component index. The first component image is not particularly informative but it does correspond to the largest projection value in both face images. The second component image is a different story. It bears a striking resemblence to the reconstruction using two components as shown in Figure 4. The remaining component images show increased spatial frequencies but do not have easily discernible patterns.

Next, we learn a common basis set for three very different images; a baboon, an outdoor scene with a boat and a girl. (Each image is 256 × 256). Prior to reconstruction, we normalized the spectra of the three images in the following manner. First, we evaluated the number of SVD components necessary to reconstruct each image with minimal loss. Then, we normalized the spectra of each image relative to the mean spectrum. The images before and after normalization are shown below in Figure 10. There are almost no visible perceptible differences.

After normalizing the spectra, we reconstructed the images using 12, 25, 50, 100 and 200 components. The results are shown in Figure 11. Somewhat to our surprise, the original images are discernible in the reconstructions using only 25 components. After about 50 components, there is no clear visual improvement which is also surprising. To take stock of what has been achieved, please note that the 25 component reconstructions use *a common matrix space basis for all*

three images. Since the original images are quite different, it is not immediately obvious why the reconstructions should so closely resemble the originals.

For a more quantitative understanding of the above reconstructions, we turn to Table 1. From the table, we see that the reconstruction error $||X_k - \hat{U}\Lambda_k V^T||_F^2$ for the girl image is worse than that of the baboon and the boat images. (Recall that the images $\{X_k\}$ have normalized spectra with the largest singular value set to one.) In addition to the reconstruction error, in Table 2, we compute the matrix space distances between the three images. The matrix space distance between image k and image l is defined as

$$D_{\text{matrixspace}}(X_k, X_l) \stackrel{\text{def}}{=} ||\hat{U}\Lambda_k\hat{V}^T - \hat{U}\Lambda_l V^T||_F^2 = ||\Lambda_k - \Lambda_l||_F^2 = \sum_{i=1}^{D}(\lambda_{ki} - \lambda_{li})^2 \tag{12}$$

where λ_{ki} is the ith projection coefficient of the kth reconstruction. Since the reconstructions are in a common matrix space, the image distances are naturally reduced to the distances between the projection coefficients. As before, the integer D denotes the number of components used for the reconstruction.

The next experiment further explores image distances when collapsed onto a single matrix space. We took six images—four faces, an outdoor scene and a fractal image—and reconstructed them using 100 components. Each image is 150×114. We first preprocess the images such that each image has its dominant singular value set to unity and with the rest of spectra normalized as previously explained. The reconstruction errors and the matrix space distances are shown in Tables 3 and 4 respectively. From the reconstruction error table, it is clear that face images 2 and 3 have the worst reconstruction errors. Given this empirical fact, we turn to the matrix space distance table in Table 4. The distances from all the face images to the outdoor scene and the Julia set are by far the largest. There is not a single face-face distance which is greater than a face-non face distance. Consequently, the reconstruction error may not be a suitable measure by which to gauge the degree of membership of an image to the estimated matrix space.

Finally, we describe initial experiments in automated filter design using the matrix space representation. The basic idea closely follows the model used in PCA filter design with one crucial difference. From the chosen image, we randomly choose N $K \times K$ blocks where K is the order of the filter mask. Given the N $K \times K$ "images", we learn a common matrix basis as before. Once U and V have been learned, we display $u_i v_i^T$ as a $K \times K$ filter mask. There are a total of K different such filters to choose from for which we implemented the following rank ordering scheme. For each filter, we evaluated the total response over the entire image. The filters were rank ordered using the total response as the metric. The learned filters and the corresponding filtered images are shown for 3×3, 7×7 and 15×15 masks in Figures 12, 13 and 14 respectively. We noticed that the first two filters always corresponded to an intensity blur and a first derivative operator respectively. Visual inspection reveals that the filters appear to be ordered according to increasing spatial frequency.

5 Discussion, Extensions and Conclusion

5.1 Isn't this just principal components (PCA)?

Not really. In most versions of PCA, the images are first converted into a vector followed by covariance matrix construction from the pattern vectors. In our approach, there is no covariance matrix formed. Instead, a common *matrix space* is estimated from the image intensity matrices. There is no denying the fact that matrix representations are at the heart of our approach as opposed to vector representations in PCA. Also, there is no statistical interpretation of matrix representations in terms of covariance matrices and dimensionality reduction as in PCA.

5.2 Matrix space distances

We view our initial experiments with matrix space distances as quite promising. Out of the six images in Table 4, two were non face images. The matrix space distances clearly and unambiguously group the face images together. The distances from each face image to the non-face images are always greater than the face-face image distances. If matrix space distance turns out to be a truly robust distance measure, it could have an impact in pattern clustering, object recognition and classification etc.

There are many ways in which the work presented here can be extended. We stress that the current algorithm is quite preliminary and one of our first goals is to add a regularization prior to better stabilize Λ_k. Once such priors are added, a more proper Bayesian justification (in terms of likelihood and prior) can be given. This would place this work in the context of earlier work by [8]. And there does not seem to be any obvious reason why mixtures of matrix bases cannot be considered in order to provide compact but *overcomplete* representations. Another promising direction of extension is the matrix space distance measure. We have only shown preliminary results of applying the distance measure on the training set. It is certainly possible to estimate basis vectors from different sets of training images and apply the matrix space distance measures to test sets as well.

In sum, we have derived a new learning algorithm for estimating matrix space representations of natural 2D grayscale images. Since the images use a common matrix basis, this allowed us to construct matrix space distances between them. Preliminary results indicate that matrix space image distances possess novel classification properties. While these initial results are promising, it remains to be seen if matrix space representations and distance measures are truly effective in image classification and recognition.

Acknowledgments

We thank Hemant Tagare, Baba Vemuri, Sudeep Sarkar and an anonymous reviewer for useful comments and criticisms. This work is partially supported by an NSF grant (IIS-9906081) to A.R.

References

1. C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, 1995.
2. H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 44–51, 2000.
3. I. Daubechies. *Ten lectures on wavelets*. SIAM, 1992.
4. J. Daugman. Complete discrete 2D Gabor transforms by neural networks for image analysis and compression. *IEEE Trans. Acoustics, SPeech and Signal Proc.*, 7:1169–1179, July 1988.
5. G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 2nd edition, 1989.
6. M. Lewicki and B. Olshausen. Probabilistic framework for the adaptation and comparison of image codes. *J. Opt. Soc. America*, 16(7):1587–1601, 1999.
7. B. Olshausen and D. Field. Natural image statistics and efficient coding. *Network*, 7:333–339, 1996.
8. B. Olshausen and K. Millmann. Learning sparse codes with a mixture of Gaussians prior. In *Advances in Neural Information Processing Systems (NIPS) 12*, pages 841–847. MIT Press, Cambridge, MA, 2000.
9. K. Rao and P. Yip. *Discrete Cosine Transform—Algorithms, Advantages and Applications*. Academic Press, London, UK, 1990.
10. J. Yang and C. Lu. Combined techniques of singular value decomposition and vector quantization for image coding. *IEEE Trans. Image Proc.*, 4:1141–1146, 1995.

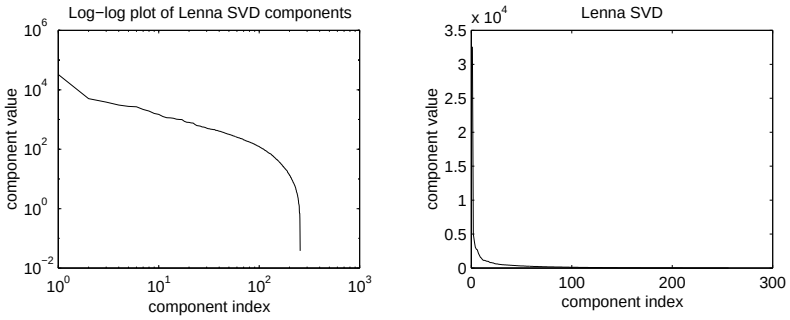

Fig. 1. Spectrum of the Lenna image (256×256). Left: Log-log plot of the spectrum. Right: Spectrum plot. Note the almost linear nature of the log-log spectrum for the dominant singular values.

Fig. 2. Lenna reconstructions and components: Top row: Reconstructions using 1, 5, 10, 15 and 20 components. Bottom row: The SVD component images $u_i v_i^T$ corresponding to the first five components.

Fig. 3. Original images. Left: Face 1. Right: Face 2.

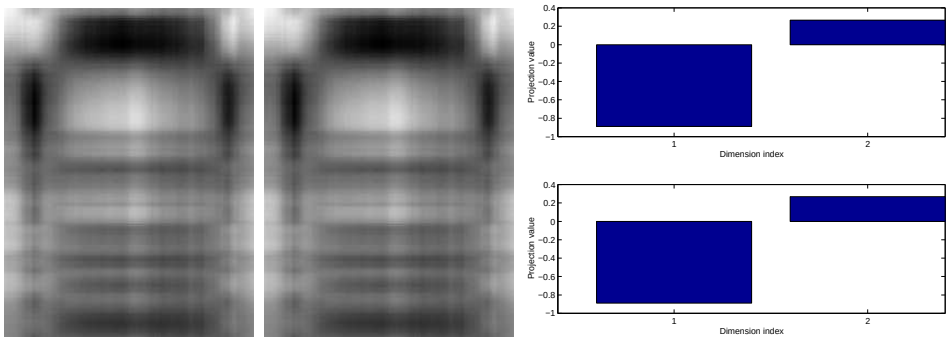

Fig. 4. Reconstructed faces using 2 components Left: Face 1. Middle: Face 2. Right: Projections.

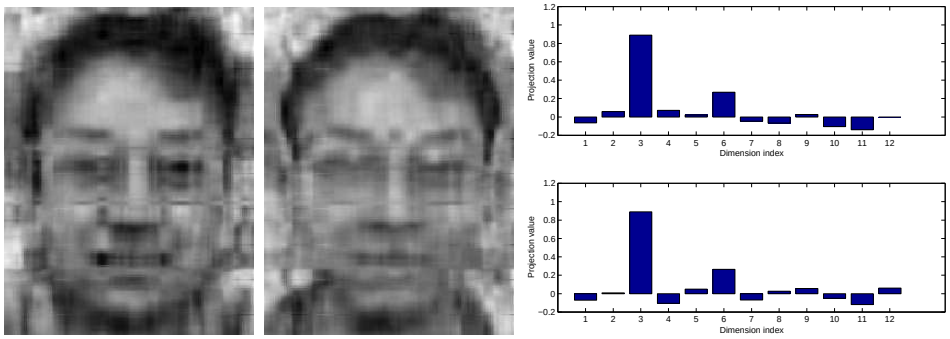

Fig. 5. Reconstructed faces using 12 components Left: Face 1. Middle: Face 2. Right: Projections.

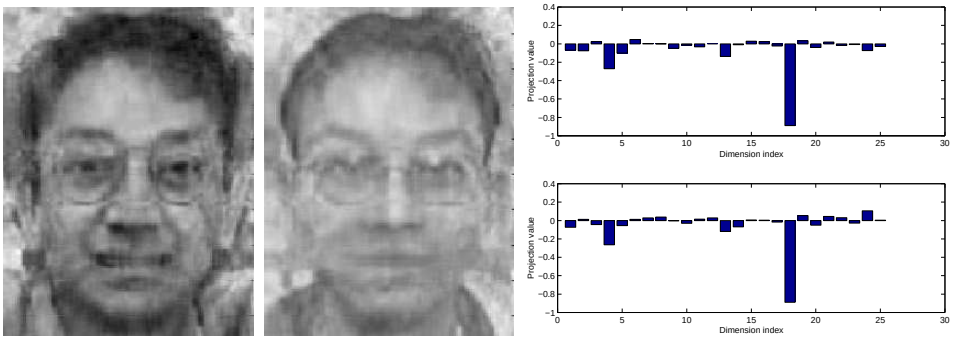

Fig. 6. Reconstructed faces using 25 components Left: Face 1. Middle: Face 2. Right: Projections.

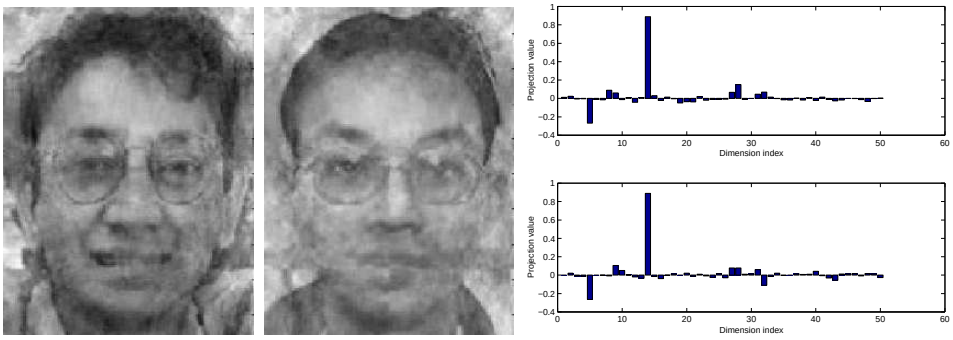

Fig. 7. Reconstructed faces using 50 components Left: Face 1. Middle: Face 2. Right: Projections.

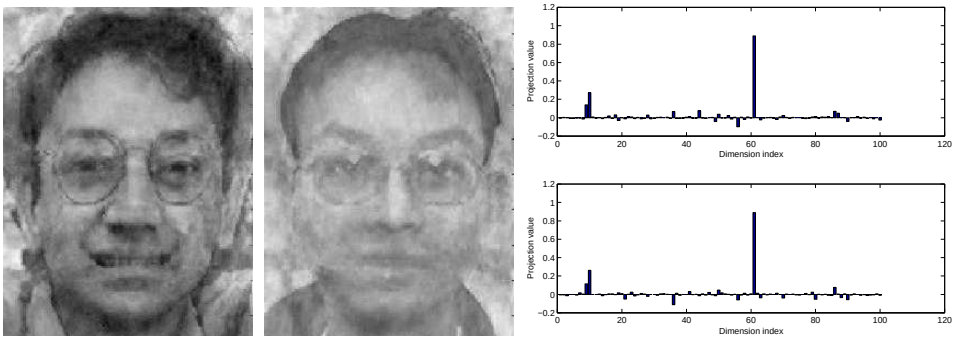

Fig. 8. Reconstructed faces using 100 components Left: Face 1. Middle: Face 2. Right: Projections.

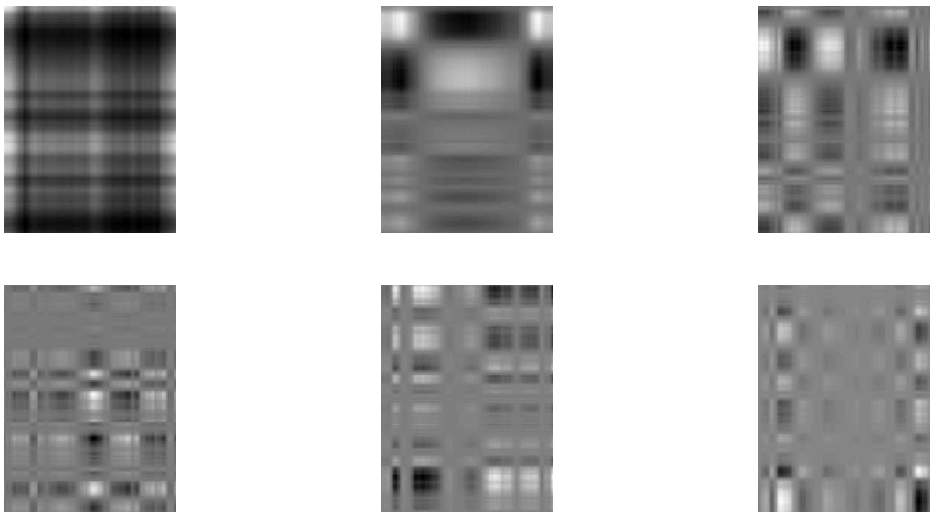

Fig. 9. Reconstruction using 12 components: First six components of the shared representation. Top: Components one through three. Bottom: Components four through six.

Reconstruction Error		
		
0.062	0.072	0.104

Table 1. Reconstruction errors for the baboon, boat and girl images using 50 components. The leading singular value for all three images is unity.

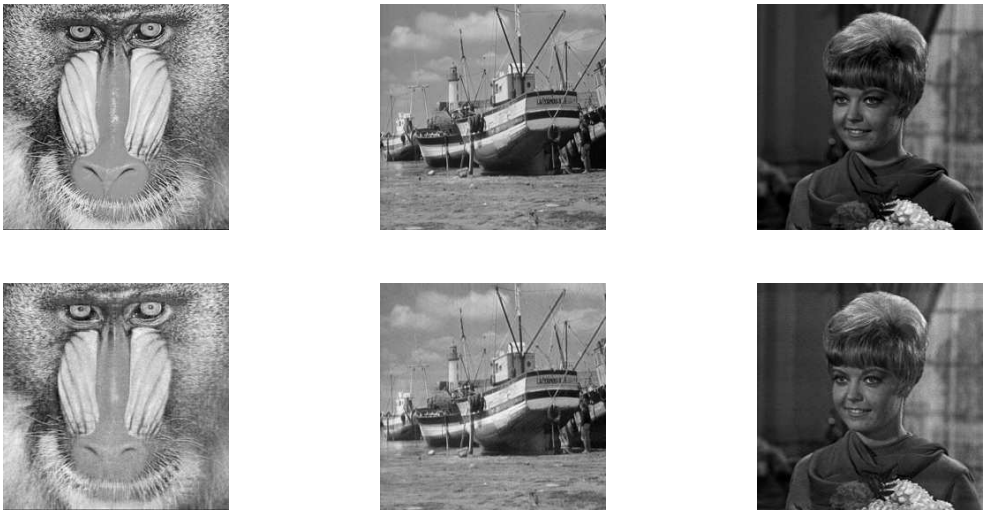

Fig. 10. Top row: Original baboon, boat and girl images. Bottom row: The same images after spectrum normalization.

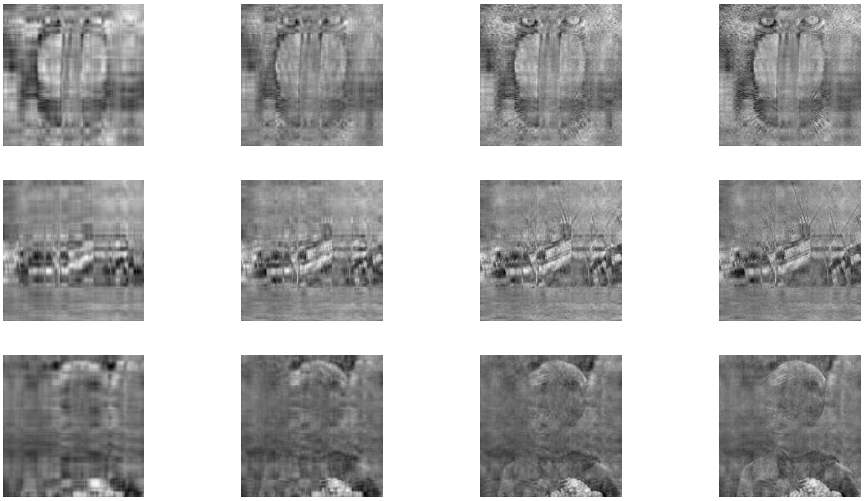

Fig. 11. Reconstructed images using 12, 25, 50, and 100 components

			
	0	0.079	0.115
	0.079	0	0.103
	0.115	0.103	0

Table 2. Matrix space distances between the three images using 50 component reconstructions.

					
0.058	0.153	0.217	0.078	0.079	0.048

Table 3. Reconstruction errors using 100 components.

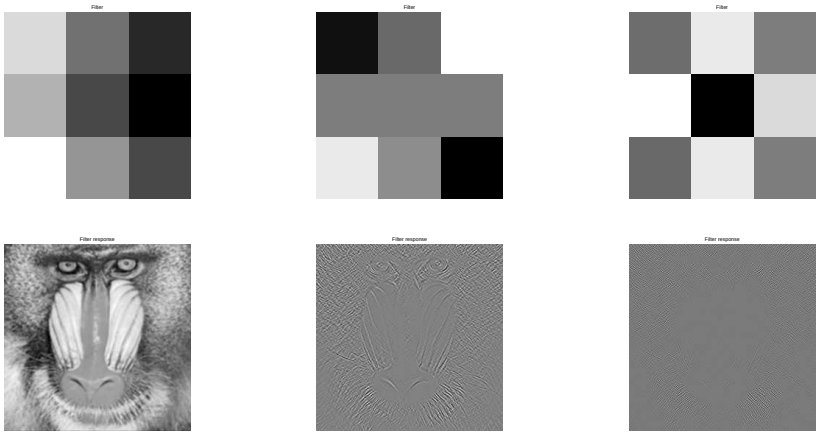

Fig. 12. Top: Learned 3×3 filters. Bottom: Corresponding filtered images. From left to right: Filters ranked according to strength of response.

						
	0	0.0237	0.0430	0.0293	0.0619	0.0853
	0.0237	0	0.035	0.0212	0.0676	0.0832
	0.043	0.035	0	0.053	0.1146	0.1220
	0.0293	0.0212	0.053	0	0.0667	0.0794
	0.0619	0.0676	0.1146	0.0667	0	0.0817
	0.0853	0.0832	0.122	0.0794	0.0817	0

Table 4. Matrix space distances between the six images using 100 component reconstructions.

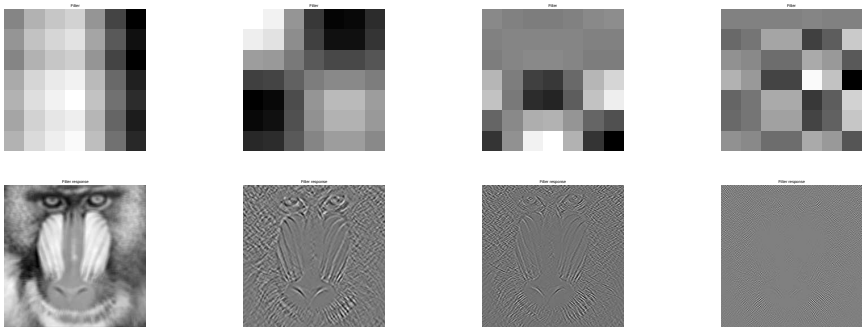

Fig. 13. Top: Learned 7×7 filters. Bottom: Corresponding filtered images. From left to right: Top 4 filters ranked according to strength of response.

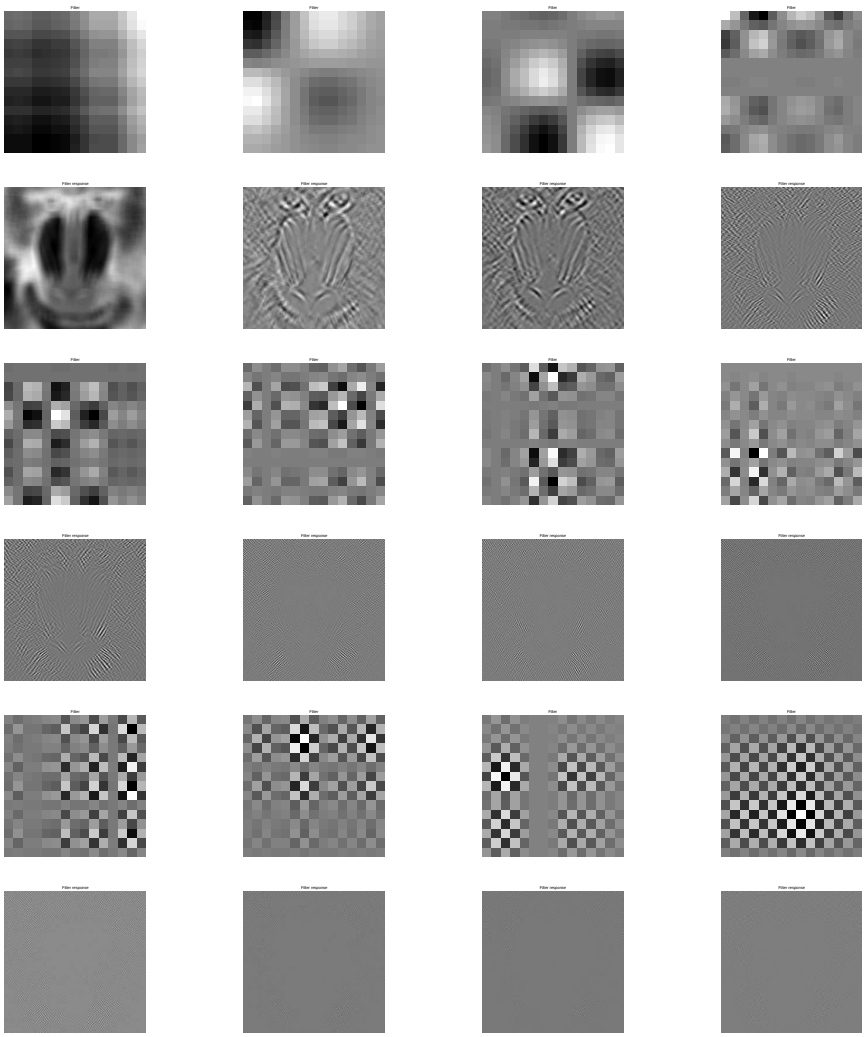

Fig. 14. Top Row: First four learned 15×15 filters. Second Row: Corresponding filtered images. Third Row: Filters 5,7,9 and 10 (15×15). Fourth Row: Corresponding filtered images. Fifth Row: Filters 12,13,14 and 15 (15×15). Sixth Row: Corresponding filtered images.