COMPACT IMAGE REPRESENTATION BY BINARY COMPONENT ANALYSIS

Zhaohui H. Sun and Anthony Hoogs

Kitware Inc., Clifton Park, NY, USA 12065

ABSTRACT

We propose binary component analysis (BCA) for compact image representation and apply it to fractional dimension reduction and Binaryface representation. BCA is similar to the widely used principal component analysis (PCA), but with the restriction of the base vectors taking binary values of +1 and -1, instead of real values. In a finite set of all binary vectors, the projection of the correlated data onto the binary components (BC) have the largest possible variances. BCA leads to fractional dimension reduction, where binary components successively reduce data variance. The top few BC's capture the largest variances, and each additional BC further reduces the variance by a fraction of the amount reduced in one dimension by PCA. BCA is applied to compact face representation as Binaryface and used for face classification.

1. INTRODUCTION

Principal component analysis (PCA) [1, 2, 3] is a multivariate statistical technique, which uses orthogonal transformation to convert correlated variables into linearly uncorrelated principal components (PC). The projection of the variables onto the PC's have the largest possible variances. PCA has been widely used in data analysis across all scientific disciplines. For example, it has been used for dimension reduction [2], as the data energy, or variance, is usually concentrated in the first few PC's, leading to compact data representation. In particular, Eigenface was proposed as a compact facial image representation for face classification and recognition [4, 5].

The focus of the paper is to extend the formulation of PCA and show the improvement over PCA-based methods. We propose binary component analysis (BCA), a data-driven analysis procedure similar to PCA, where the base vectors are restricted to binary values of +1 and -1. The primary binary component (PBC) is selected from all possible binary vectors, upon which the correlated variables have the largest variance. After removing the variance captured by the PBC, additional PBC's of the data residue are sought, thus successively removing the correlation in the data. All the PBC's on the data residue over iterations are collectively called the binary components (BC) of the original data. Note the solution space of PCA is an infinite number of unit vectors, whereas the solution space of BCA is limited to a finite number of binary vectors. Related prior works used three values of -1, +1, and 0 in [6] and binary box functions to span a subspace in [7]. The base vectors are a linear combination of Haar-like box functions and highly non-orthogonal. The difference becomes obvious by comparing Fig. 3(c) in [7] and Fig. 3.

In the first application, binary component analysis is used for fractional dimension reduction. Although each BC is suboptimal compared to PC, in terms of variance reduction, it only takes a single bit to code the binary value. As a comparison, the principal components take real values, which are usually stored in four bytes, or 32 bits, with sufficient precision. It is impossible to truncate the precision of principal components without breaking the correlation condition, i.e. $\mathbf{e}_i'\mathbf{e}_j = 0$ for $i \neq j$. When the PC's become linearly correlated due to significant precision loss, there is residue carrying over from dimension to dimension, and PCA just falls apart. The conventional approach is to truncate the number of PC's, not the precision of each PC. The experiment in Fig. 2(a) shows it takes 2.5 bits (not 32 bits) of BCA to reduce the amount of variance equivalent to one dimension of PC.

In the second application of face representation and recognition, we formulate the binary component as Binaryface, an alternative to the Eigenface representation. The Binaryfaces in Fig. 3 have great similarity to the Haar wavelets. The storage space for one Eigenface can hold up to 32 Binaryfaces. In addition, the binary coefficients of binary components are used as expressive features for face recognition. As shown by the five-fold cross validation using Support Vector Machine [8] in Fig. 2(b), the BCA-based feature significantly improves the cross validation precision over the PCA-based feature and other binary codes generated by Iterative Quantization (ITQ) [9] and Locality Sensitive Hashing (LSH) [10, 11].

Our main contributions include the concept and derivation of binary component analysis and the applications to fractional dimension reduction and Binaryface representation.

2. BINARY COMPONENT ANALYSIS

2.1. The Concept

Fig. 1 illustrates the principal component, the proposed binary component, and their differences. Let us consider a random point cloud centered around $(0, 0)$ on the 2D plane. The M points can be fit into a $M \times 2$ feature matrix $\mathbf{X}$, one point per row. The covariance matrix $\mathbf{C}$ has a dimension of $2 \times$

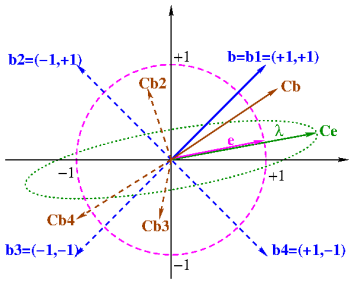

Fig. 1. Illustration of the principal component $\mathbf{e} \in \Re^2$ and the proposed primary binary component $\mathbf{b} \in \{-1,+1\}^2$.

2. Among all the unit vectors $\mathbf{u}$ on the pink dashed circle in Fig. 1, $\|\mathbf{u}\| = 1$, $\mathbf{u} \in \Re^{2\times 1}$, one is very special, $\mathbf{e} = \arg\max_{\mathbf{u}}\{\mathbf{u}'\mathbf{C}\mathbf{u}\}$, and called the principal component. The projection of the point cloud onto the unit vectors reach the maximal variance $\mathrm{Var}[\mathbf{X}\mathbf{u}]$ at $\mathbf{e}$. In addition, the projection of $\mathbf{C}$ on $\mathbf{u}$ has the same direction at $\mathbf{e}$, $\mathbf{C}\mathbf{e} = \lambda\mathbf{e}$, $\lambda \geq 0$, and $\lambda \in \Re$. After removing the component with the largest uncertainty, $\mathbf{X} \leftarrow \mathbf{X} - \lambda\mathbf{e}\mathbf{e}'$, PCA can be carried out on the following dimensions, subject to the orthogonal constraints of $\mathbf{e}_i'\mathbf{e}_j = 0$, for $i \neq j$.

Now consider the four binary vectors, $\mathbf{b}_1 = (1,1)'$, $\mathbf{b}_2 = (-1,1)'$, $\mathbf{b}_3 = (-1,-1)'$, and $\mathbf{b}_4 = (1,-1)'$ on the 2D plane. The projections of $\mathbf{C}$ on these binary vectors are depicted as $\mathbf{C}\mathbf{b}$, $\mathbf{C}\mathbf{b}_2$, $\mathbf{C}\mathbf{b}_3$, and $\mathbf{C}\mathbf{b}_4$ in Fig. 1. The one yields the maximum $\mathbf{b}'\mathbf{C}\mathbf{b}$ is coined as the primary binary component. In other words, when the random points are projected on these four vectors, the projection on the PBC has the largest variance $\mathrm{Var}[\mathbf{X}\mathbf{b}]$. Similarly, after removing the largest uncertainty, $\mathbf{X} \leftarrow \mathbf{X} - 1/2\mathbf{X}\mathbf{b}\mathbf{b}'$, the so called binary component analysis can be carried out on the next bits, while minimizing the correlation between the binary components $(\mathbf{b}_i'\mathbf{b}_j)^2$ for $i \neq j$.

The primary binary component is a binary quantization of the principal component subject to the correlation constraints (6). Projecting the random points onto the fixed vectors, such as the primary binary component, is sub-optimal in the sense of maximizing the projection variance, i.e. $\frac{1}{N}\mathbf{b}'\mathbf{C}\mathbf{b} \leq \mathbf{e}'\mathbf{C}\mathbf{e}$. However, it significantly reduced the number of bits needed to encode each base vector, reducing the infinite candidates on the unit circle to just four binary vectors in Fig. 1. Each element of the principal component is a real number, and is usually represented by four bytes or 32 bits in modern computers with sufficient precision. On the other hand, each element of the binary component requires only one bit to represent +1 or -1. Each bit of BCA reduces a fraction of the variance by one dimension of PCA, leading to fractional dimension reduction.

2.2. Primary Binary Component

Definition 1. Primary binary component *(PBC) of a zero-centered matrix* $\mathbf{X} \in \Re^{M\times N}$ *is a binary vector* $\mathbf{b} \in \{-1,+1\}^N$ *which*

$$\textit{maximizes } \mathbf{b}'\mathbf{X}'\mathbf{X}\mathbf{b}, \textit{ subject to } b_i^2 = 1,\ i = 1,\ldots,N. \quad (1)$$

The primary binary component is one of the finite binary vectors $\{-1,+1\}^{N\times 1}$, upon which the projection of random variables have the largest variance or uncertainty.

To maximize the cost subject to N constraints in (1), we use Lagrange multiplier to find the binary vector $\mathbf{b}$, which maximizes the cost function E,

$$E = \mathbf{b}'\mathbf{C}\mathbf{b} - \sum_{i=1}^{N}\gamma_i(b_i^2 - 1) = \mathbf{b}'\mathbf{C}\mathbf{b} - \mathbf{b}'\mathbf{\Gamma}\mathbf{b} + \mathrm{Tr}(\mathbf{\Gamma}), \quad (2)$$

where the multiplier, $\gamma_i \in \Re$, in (2) are the elements in a diagonal matrix $\mathbf{\Gamma} = diag\{\gamma_1,\ldots,\gamma_N\}$, and $\mathbf{C} = \mathbf{X}'\mathbf{X}$.

Take derivative of E in (2) with respect to $\mathbf{b}$ and set it to $\mathbf{0}$, $\partial E/\partial\mathbf{b} = \mathbf{C}\mathbf{b} - \mathbf{\Gamma}\mathbf{b} = \mathbf{0}$, yielding $\mathbf{C}\mathbf{b} = \mathbf{\Gamma}\mathbf{b}$. The real vector $\mathbf{v} = \mathbf{C}\mathbf{b}$, $\mathbf{v} \in \Re^{N\times 1}$ is the projection of a $N \times N$ matrix $\mathbf{C}$ on a binary vector $\mathbf{b}$. The magnitude $|v_i|$, or $-|v_i|$, is captured in γ_i and the diagonal elements of $\mathbf{\Gamma}$, while its sign $sgn\{v_i\}$, or $-sgn\{v_i\}$, is captured in b_i. Therefore the maximum of (2) is reached at $E_{max} = \mathrm{Tr}(\mathbf{\Gamma}) = \sum_{i=1}^{N}\gamma_i$.

2.3. Binary Components

Definition 2. Residue Matrices $\mathbf{X}_k$, $k \in \mathcal{N}$ *of a zero-centered matrix* $\mathbf{X} \in \Re^{M\times N}$, *are defined as*

$$\mathbf{X}_1 = \mathbf{X}, \quad (3)$$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \frac{1}{N}\mathbf{X}_k\mathbf{b}_k\mathbf{b}_k', \quad (4)$$

where $\mathbf{b}_k$ *is the primary binary component of* $\mathbf{X}_k$.

We recursively seek the primary binary component $\mathbf{b}_k$ of the residue matrix $\mathbf{X}_k$ until the residue norm $\|\mathbf{X}_k\|$ goes to zero. All the primary binary components of the residue matrices are called the binary components of the original matrix $\mathbf{X}$. In addition, the binary components are selected as uncorrelated as possible.

Definition 3. *Binary components. Given a zero-centered matrix* $\mathbf{X} \in \Re^{M\times N}$, *the binary components are the binary vector* $\mathbf{b}_k \in \{-1,+1\}^N$, $1 \leq k \leq K$, *satisfying the following conditions,*

$$\textit{maximize} \quad \mathbf{b}_\mathbf{k}'\mathbf{X}_\mathbf{k}'\mathbf{X}_\mathbf{k}\mathbf{b}_\mathbf{k} \quad (5)$$

$$\textit{and minimize} \quad (\mathbf{b}_l'\mathbf{b}_k)^2,\ 1 \leq l < k \quad (6)$$

$$\textit{subject to} \quad b_{ki}^2 = 1,\ i = 1,\ldots,N. \quad (7)$$

To solve the multi-objective optimization (5-7), we again use Lagrange multiplier to find the binary vector $\mathbf{b}_k$, which maximizes the cost function E_k, for $k = 1,\ldots,K$,

$$E_k = \mathbf{b}_\mathbf{k}'\mathbf{C}_\mathbf{k}\mathbf{b}_\mathbf{k} - \mathbf{b}_\mathbf{k}'\mathbf{\Gamma}_\mathbf{k}\mathbf{b}_\mathbf{k} + \mathrm{Tr}(\mathbf{\Gamma}_\mathbf{k}) - \Phi_k\sum_{l=1}^{k-1}(\mathbf{b}_l'\mathbf{b}_k)^2. \quad (8)$$

The first three terms are similar to (2). The last term requires $\mathbf{b}_k$ to be mostly uncorrelated to the previously selected BC's, $\mathbf{b}_l$, $l < k$, e.g., by having half (or close to half) elements with the same sign and the rest with opposite signs.

We take the derivative of E_k over $\mathbf{b}_k$ and set it to $\mathbf{0}$,

$$\partial E_k / \partial \mathbf{b_k} = \mathbf{C_k b_k} - \mathbf{\Gamma_k b_k} - \Phi_k (\sum_{l=1}^{k-1} \mathbf{b}_l' \mathbf{b}_k \mathbf{b}_l) = \mathbf{0}, \quad (9)$$

and solve for the multiplier

$$\Phi_k = \frac{(\sum_{l=1}^{k-1} \mathbf{b}_l' \mathbf{b}_k \mathbf{b}_l')(\mathbf{C_k} - \mathbf{\Gamma_k})\mathbf{b_k}}{(\sum_{l=1}^{k-1} \mathbf{b}_l' \mathbf{b}_k \mathbf{b}_l')(\sum_{l=1}^{k-1} \mathbf{b}_l' \mathbf{b}_k \mathbf{b}_l)}. \quad (10)$$

The magnitude of $\mathbf{C_k b_k}$ is captured in $\mathbf{\Gamma}_k$ and the signs in $\mathbf{b}_k$. Having fixed the multipliers of $\mathbf{\Gamma}_k$ and Φ_k, the maximum of (8) is reached at $E_{k_{max}} = \mathrm{Tr}(\mathbf{\Gamma}_k) - \Phi_k \sum_{l=1}^{k-1} (\mathbf{b}_l' \mathbf{b_k})$.

2.4. Computation of Binary Components

We propose an exhaustive search algorithm and a successive PCA algorithm (BCA-PCA) to compute the binary components. To optimize (5-7), the exhaustive search algorithm goes through all binary vectors in the solution space and picks the one with the maximum value $E_{k_{max}}$ in (8). The solution space of the binary components is finite. For data in N dimensions, the size is 2^N. When N is small or moderate, it is possible to enumerate the solution space and find the optimal solution. However, the size grows exponentially and it is impractical for high dimension N. For large N, we resort to the following sub-optimal solution.

The BCA-PCA algorithm takes successive PCA on the residue matrices $\mathbf{X}_k$, and sets BC as $\mathbf{b}_k = sgn\{\mathbf{e_k}\}$, or $\mathbf{b}_k = -sgn\{\mathbf{e_k}\}$, where $\mathbf{e_k}$ is the most dominant principal component of $\mathbf{X}_k$. The correlation constraints between the BC's (6) are not explicitly enforced. Note the residue matrix is constructed on BC $\mathbf{b}_k$ in (4), not PC $\mathbf{e}_k$. This allows the quantization error to propagate to the next residue matrix. The solution is sub-optimal, as it can be shown that

$$\frac{1}{N}\mathbf{e}'\mathbf{C}\mathbf{e} \le \frac{1}{N}\mathbf{b}'\mathbf{C}\mathbf{b} \le \mathbf{e}'\mathbf{C}\mathbf{e}. \quad (11)$$

The first inequality in (11) indicates the variance of data projection on the selected PBC ($\mathbf{b}$) can be reduced, by at least a fraction of the variance reduction on the PC. The second inequality in (11) says the variance reduction is usually not as strong as the projection on the principal component ($\mathbf{e}$). In other words, the PBC captures some or most, but usually not all, of the uncertainty in one dimension. The quantization error is then propagated or diffused to the following bits.

3. FRACTIONAL DIMENSION REDUCTION

BCA can be used for dimension reduction. Each bit reduces the data variance by a fraction of that reduced by PCA in one dimension. It usually costs fewer than 32 bits to reduce the data variance in one dimension, resulting in fractional dimension reduction. In the example shown in Fig. 2, it costs about 2.5 bits to encode each additional dimension. In addition, the number of bits K is no longer limited by the data dimension N, so longer binary codeword ($K > N$) can be generated.

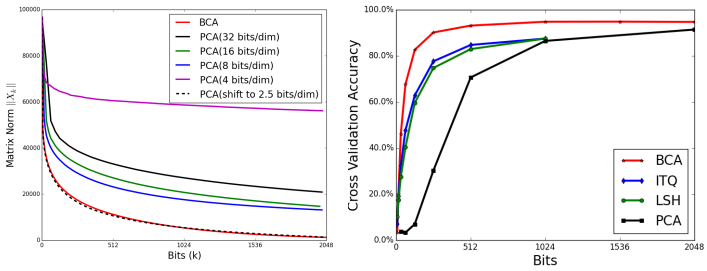

Fig. 2. (a) Matrix norm of $\|\mathbf{X}_k\|$ decreases monotonically over bit index k. (b) Comparison of face classification accuracy by five-fold cross validation of SVM.

Theorem 1. *Given the primary binary component* $\mathbf{b} \in \{-1, +1\}^{N \times 1}$ *of a zero-centered real matrix* $\mathbf{X} \in \Re^{M \times N}$,

$$\|\mathbf{X}\|^2 - \|\mathbf{X} - \frac{1}{N}\mathbf{X}\mathbf{b}\mathbf{b}'\|^2 = \frac{1}{N} Tr(\mathbf{\Gamma}). \quad (12)$$

Proof.

$$\|\mathbf{X} - \frac{1}{N}\mathbf{X}\mathbf{b}\mathbf{b}'\|^2 = \mathrm{Tr}((\mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}')\mathbf{X}'\mathbf{X}(\mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}'))$$

$$= \mathrm{Tr}(\mathbf{X}'\mathbf{X}(\mathbf{I} - \frac{2}{N}\mathbf{b}\mathbf{b}' + \frac{1}{N^2}\mathbf{b}\mathbf{b}'\mathbf{b}\mathbf{b}')) \quad (13)$$

$$= \mathrm{Tr}(\mathbf{X}'\mathbf{X}) - \frac{1}{N}\mathrm{Tr}(\mathbf{X}'\mathbf{X}\mathbf{b}\mathbf{b}') \quad (14)$$

$$= \mathrm{Tr}(\mathbf{X}'\mathbf{X}) - \frac{1}{N}\mathrm{Tr}(\mathbf{\Gamma}\mathbf{b}\mathbf{b}') = \|\mathbf{X}\|^2 - \frac{1}{N}\mathrm{Tr}(\mathbf{\Gamma}). \quad (15)$$

The Frobenius norm is first expanded, where Tr denotes matrix trace and $\mathbf{I}$ is the $N \times N$ identity matrix. Term $(\mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}')(\mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}')$ is expanded in (13). Since $\mathbf{b}'\mathbf{b} = N$, it is simplified as $\mathbf{I} - \frac{1}{N}\mathbf{b}\mathbf{b}'$. The trace of matrix difference in written as the difference of traces in (14). The second term is further replaced by $\mathbf{\Gamma}\mathbf{b}\mathbf{b}'$ in (15). Recall $\mathbf{\Gamma}$ is a diagonal matrix, and $\mathbf{b}\mathbf{b}' \in \{-1, +1\}^{N \times N}$ has binary values of +1 and -1, with diagonal elements of 1. Therefore $\mathrm{Tr}(\mathbf{\Gamma}\mathbf{b}\mathbf{b}') = \mathrm{Tr}(\mathbf{\Gamma})$. □

Corollary 1.1. *The Frobenius norm of the residue matrices in (4) are bounded non-increasing monotonic function of the bit count,* $k, l \in \mathcal{N}$

$$0 \le \|\mathbf{X}^{k+l}\| \le \|\mathbf{X}^k\| \le \|\mathbf{X}\|. \quad (16)$$

The BC's successively reduce the data variance and the norm of residue matrices. In addition, the first few carry more weights as the variance reduction is more significant.

Fig. 3. From left to right columns, four face images from ExtYaleB, the top four Eigenfaces, and the top 32 Binaryfaces.

4. EXPERIMENTAL RESULTS

We use the extended Yale face database B (ExtYaleB) [1] [12, 13] to validate fractional dimension reduction, and binary face representation and classification. The dataset contain 2452 frontal face images of 39 human subjects under 9 poses and 64 illumination conditions. Four randomly selected images and the top four Eigenfaces are shown in the left column of Fig. 3. The images are resized to 32×32 pixels, vectorized and stacked into a data matrix $\mathbf{X}$, one face per row. After removing the mean, it is converted to a zero-centered matrix $\mathbf{X}$, with $M = 2452$ rows and $N = 1024$ columns.

We apply BCA to the data matrix $\mathbf{X} \in \Re^{2452\times1024}$. The norm of the residue matrix $\|\mathbf{X}_k\|$ is computed for $k = 1, \ldots, = 2048$, and plotted as the red curve in Fig. 2(a). The matrix norm decreases monotonically and converges to 0 for large k. Most of the norm reduction happens at the first few bits. Note the bit count k goes up to $K = 2048$, way beyond the data dimension of $N = 1024$. Next we apply PCA to $\mathbf{X}$ and plot the norm of the residue matrices. The black, green, and blue lines show successive matrix norm reduction when each PC in [-1,1] is truncated to 32, 16, and 8 bits for the first 64, 128, and 256 dimensions. When it falls to 4 bits per PC over 512 dimensions (magenta), the residue norm is significantly increased. It is obvious that BCA (red line) consistently outperforms all PCA settings. When the black curve (32 bits per PC element over 64 dimensions) is artificially scaled by setting the X axis at $2.5 * k$ (instead of $32 * k$), the black dashed curve follows the red curve quite well. It shows there is quite a bit redundancy to use 32 bits per dimension. BCA can reduce matrix norm at the rate of 2.5 bits per dimension. In fact, to reduce $\|X\|$ by 95%, it takes BCA 1090 bits and PCA 445 dimensions.

Next we apply BCA to face representation, and call the binary components Binaryfaces. The decomposition as Binaryfaces leads to compact face representation, which have great similarity to the Haar wavelets. The top 32 Binaryfaces are shown in the right columns of Fig. 3. The first Binaryface captures mainly the background. The second Binaryface, with white right and black left, captures illumination in horizontal direction. The third Binaryface outlines the face region from the background. The fourth Binaryface captures the left half and the right half, but with opposite signs. The fifth Binaryface partitions the top half and the bottom half, capturing illumination in vertical direction. Each pixel on the Binaryfaces is either +1 or -1, and coded by one bit. Therefore, it takes the same amount of storage to capture the 32 Binaryfaces as one Eigenface highlighted with green border. Note only face encoding into binary code is studied here, and perfect face reconstruction from decoding is not always possible.

At last we evaluate the expressiveness of the binary features derived from BCA for face classification. Five-fold cross validation is carried out using linear SVM [8]. 80% of the binary features are randomly selected for training and the rest 20% for testing. The classification results are compared against the subject label, and the cross validation accuracy is evaluated. The experiment is repeated for code length $K = 8, 64, 32, 64, 128, 256, 512, 1024, 1536, 2048$, and the results are shown as the red curve in Fig. 2(b). In addition, we generate the binary codes using ITQ and LSH, and plot the cross validation accuracy as the blue and green curves. Since the methods cannot generate code longer than data dimension $N = 1024$, both curves stop at 1024 bits. Finally, the PCA coefficients of the top 64 Eigenfaces are scaled to [-1,+1] and used for training, testing, and face classification. The results are shown as the black curve, assuming each coefficient takes 32 bits. The accuracy by PCA keeps increasing after the first 64 dimensions. It is clear that the red curve by BCA consistently outperforms the other methods, indicating the binary feature by BCA is more expressive for classification.

5. CONCLUSIONS

We have proposed BCA as an alternative to PCA for general data analysis, where the base vectors are restricted to binary values of +1 and -1. BCA leads to fractional dimension reduction, a more compact representation using fewer bits per dimension. BCA is also applied to Binaryface representation yielding better classification performance.

[1] http://vision.ucsd.edu/∼iskwak/ExtYaleDatabase/ExtYaleB.html

6. REFERENCES

[1] Ian Jolliffe, *Principal component analysis*, Springer, 2002.

[2] G. H. Golub and C. F. Van Loan, *Matrix computations*, Johns Hopkins University Press, Baltimore, Maryland, 2013.

[3] W. H. Press, S. A. Teulolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes*, Cambridge University Press, 2007.

[4] Lawrence Sirovich and Michael Kirby, "Low-dimensional procedure for the characterization of human faces," *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.

[5] Matthew Turk and Alex P Pentland, "Face recognition using eigenfaces," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[6] Dianne O'Leary and Shmuel Peleg, "Digital image compression by outer product expansion," *IEEE Transactions on Communications*, vol. 31, no. 3, pp. 441–444, 1983.

[7] Feng Tang and Hai Tao, "Binary principal component analysis," in *British Machine Vision Conference*, 2006, pp. 377–386.

[8] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[9] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin., "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[10] Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *ACM symposium on Theory of computing*, 1998, pp. 604–613.

[11] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al., "Similarity search in high dimensions via hashing," in *VLDB*, 1999, vol. 99, pp. 518–529.

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

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