

A Direct Evolutionary Feature Extraction Algorithm for Classifying High Dimensional Data

Qijun Zhao and David Zhang

Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
{csqjzha, csdzhang}@comp.polyu.edu.hk

Hongtao Lu

Department of Computer Science and Engineering
Shanghai Jiao Tong University
Shanghai, 200030, China
lu-ht@cs.sjtu.edu.cn

Abstract

Among various feature extraction algorithms, those based on genetic algorithms are promising owing to their potential parallelizability and possible applications in large scale and high dimensional data classification. However, existing genetic algorithm based feature extraction algorithms are either limited in searching optimal projection basis vectors or costly in both time and space complexities and thus not directly applicable to high dimensional data. In this paper, a direct evolutionary feature extraction algorithm is proposed for classifying high-dimensional data. It constructs projection basis vectors using the linear combination of the basis of the search space and the technique of orthogonal complement. It also constrains the search space when seeking for the optimal projection basis vectors. It evaluates individuals according to the classification performance on a subset of the training samples and the generalization ability of the projection basis vectors represented by the individuals. We compared the proposed algorithm with some representative feature extraction algorithms in face recognition, including the evolutionary pursuit algorithm, Eigenfaces, and Fisherfaces. The results on the widely-used Yale and ORL face databases show that the proposed algorithm has an excellent performance in classification while reducing the space complexity by an order of magnitude.

I. Introduction

A central issue in pattern recognition is how to extract discriminating features for patterns, which can significantly affect the classification performance. Generally, samples are represented as vectors of feature values, and all of them lie in a vector space, whose basis consists of the feature vectors. Unfortunately, there is no guarantee that the original representation of samples is composed of the most discriminant features. More often than not, transforming the original features and discarding some redundant features can greatly improve the classification performance. These actually underlie all feature extraction algorithms.

Linear transformation is widely used for feature extraction in real-world applications. For example, Principal Component Analysis (PCA) and Linear Discriminant Analysis

(LDA), two classical feature extraction methods, both linearly transform the original features to get new representations (Fukunaga 1990). Mathematically, given an $n \times N$ sample matrix $X = [x_1 x_2 \cdots x_N]$ (n is the original dimension of samples, and N is the number of samples), linear feature extraction algorithms will acquire an $n \times m$ transform matrix W , with which the new representations of samples can be derived as $Y = W^T X = [y_1 y_2 \cdots y_N]$. Here, $0 < m \ll n$ is the dimension of the transformed feature space.

According to the latest Face Recognition Vendor Test 2002 (<http://www.frvt.org/>) and the Face Authentication Test on the ICPR'04 (Messer & others 2004), most state-of-the-art face recognition systems still build themselves on the statistics based feature extraction methods, although many new techniques, kernel methods (Scholkopf & others 1998), for instance, have been proposed in the last decade (Zhao & others 2003). Eigenfaces (Turk & Pentland 1991) and Fisherfaces (Belhumeur & others 1997) are among the classical statistic feature extraction methods in the literature of face recognition. Eigenfaces is based on the technique of PCA, which projects the original data onto the directions with maximum variances over the whole data set. On the other hand, Fisherfaces, based on the technique of LDA, projects the data from the original space to a lower-dimensional space such that, in the reduced space, the data from the same categories become much closer, whereas those from different categories become much further. These two methods are widely used. Their basic idea has a great impact on most popular methods in the face recognition area and they have become defacto benchmark methods in the performance evaluation. However, as statistic methods, they both suffer from the problem of small sample size, which refers to the performance degradation due to the relatively small sample size compared with the high dimensionality of data.

Feature extraction is essentially a kind of optimization problems. As for linear feature extraction, its task is to seek for m projection basis vectors $w_1, w_2, \cdots, w_m \in R^n$, i.e. the column vectors in the transform matrix W , such that the resulting transformed samples are most distinguishable. Therefore, feature extraction can acquire its solutions via such optimization techniques as Genetic Algorithms (GA) (Goldberg 1989). More specifically, the GA encodes a set of candidate transform matrixes, or features, in individuals.

These individuals are evaluated according to the classification performance of the transformed features obtained from them. The GA then generates a new generation of individuals for the following iteration based upon the evaluation. As a result, the classification performance of candidate transform matrixes is improved along the GA iterations. When the GA converges, the optimal transform matrix is finally given by the fittest individual, i.e. the individual having the highest value in evaluation. We call these GA based feature extraction algorithms as evolutionary feature extraction algorithms (EFE). Compared with other feature extraction algorithms, EFE algorithms have the advantage of potential parallelizability and are thus expected to be more applicable for large-scale and high-dimensional data.

Existing EFE algorithms encode candidate transform matrixes, in other words generate candidate projection basis vectors, primarily in two ways. Some EFE algorithms generate the projection basis vectors via scaling the original features or the principal vectors of the sample data (Siedlecki & Sklansky 1989)(Pei & others 1995)(Vafaie & De Jong 1995)(Raymer & others 2000)(Zheng & others 2005) or via algebraic operations on them (Vafaie & De Jong 1998)(Smith & Bull 2003). The others construct the projection basis vectors through rotating the standard basis of the search space (Liu & Wechsler 2000)(Zhao & Lu 2005). The drawback of scaling and algebraic operations lies in their limited ability of searching for projection basis vectors. Although the rotation method has no such limitation, its computational cost is high due to the large amount of trigonometric operations it involved and its evaluation on the whole set of training samples. Recently, Zhao et al. proposed a fast EFE algorithm which successfully improved the efficiency of the EFE algorithm while not impairing its classification performance (Zhao & others 2006). However, both the rotation based EFE algorithm (REFE) and the fast EFE algorithm (FEFE) have a high space complexity. The individual defined in REFE has a length of $5n^2 - 4n$ and the length of the individual in FEFE is $11n^2 + n$. When the original dimension is high, these algorithms will not workable, in which case such techniques as PCA and whitened PCA (WPCA) are often used to reduce the dimension of data in advance (Liu & Wechsler 2000). However, another two problems arise when employing the pre-dimension reduction process. For one thing, some useful discriminant information could be lost in the process. For another, there is no systematic method to determine the pre-reduced dimension.

In this paper, we present a direct evolutionary feature extraction algorithm (DEFE). It generates the projection basis vectors by linearly combining the basis of the search space and using the orthogonal complement technique. The DEFE algorithm reduces the space complexity of EFE from $O(n^2)$ to $O(n)$, which enables it to be directly applicable for high dimensional data. Moreover, it improves the efficiency of EFE by the means of constraining the search space via incorporating the analytical results on LDA solution spaces into the construction of projection basis vectors. In the next section, we give the proposed DEFE algorithm in detail. We then show our face recognition experiments in section III and conclude this paper in the last section.

II. The Direct Evolutionary Feature Extraction Algorithm: DEFE

A. Constructing Projection Basis Vectors

The aim of constructing projection basis vectors is to generate candidate transform matrixes for the genetic algorithm. Generally, the whole set of candidate projection basis vectors are encoded in an individual. This is the reason why the space complexity of existing evolutionary feature extraction algorithms is high. Here, in order to reduce the space complexity and make the EFE algorithm more applicable for high dimensional data, we propose to construct projection basis vectors using the linear combination of the basis of the search space and the orthogonal complement technique. As a result, only one vector is needed to encode in the individual.

First, we generate one vector via linearly combining the basis of the search space. Let $\{e_i \in R^n | i = 1, 2, \dots, n\}$ be the basis of the search space R^n , and $\{a_i \in R | i = 1, 2, \dots, n\}$ be the linear combination coefficients. Then we can get a vector as follows

$$v = \sum_{i=1}^n a_i e_i. \quad (1)$$

Second, we calculate a basis of the orthogonal complement space of $V = \text{span}\{v\}$, the space expanded by v , in R^n . Let $\{u_i \in R^n | i = 1, 2, \dots, n-1\}$ be the basis, and assume $U = \text{span}\{u_1, u_2, \dots, u_{n-1}\}$, then

$$R^n = V \oplus U, \quad (2)$$

where ' \oplus ' represents the direct sum of vector spaces, and

$$U = V^\perp, \quad (3)$$

where ' \perp ' denotes the orthogonal complement space. Finally, we randomly choose part of this basis as the projection basis vectors.

According to the above method of generating projection basis vectors, the information encoded in an individual includes the n combination coefficients and $n-1$ selection bits. Each coefficient is represented by 11 bits with the leftmost bit denoting its sign ('0' means negative and '1' positive) and the remaining 10 bits giving its value as a binary decimal. Figure 1 shows such individual, in which the selection bits b_1, b_2, \dots, b_{n-1} , taking the value of '0' or '1', indicate whether the corresponding basis vector is chosen as a projection basis vector. The individual under such definition has $12n-1$ bits. Apparently, it is much shorter than the one used by existing EFE algorithms, and thus the DEFE algorithm proposed here has a much lower space complexity.

B. Evaluating Individuals

We evaluate individuals based on both the performance accuracy on training samples and the generalization ability of the projection basis vectors represented by the individuals. In order to evaluate an individual D , we first retrieve the projection basis vectors from it and project all sample data onto them. Considering the high cost of classifying the whole

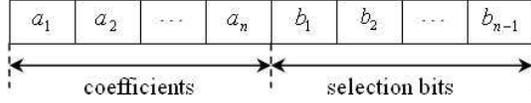


Figure 1: The individual defined in the DEFE algorithm. Each coefficient is represented with 11 bits.

set of training samples in evaluation, we measure the performance accuracy of D by the classification accuracy of a minimum distance classifier on a subset of the training samples instead of the whole training set. In this paper, we use the set of samples which are furthest from their cluster centers. We call this set as the boundary subset. Let $N_{BS}(D)$ and $N_{BS}^c(D)$ be the size of the boundary subset and the number of the correctly classified samples in it, the performance accuracy is then defined as

$$\zeta_a(D) = N_{BS}^c(D)/N_{BS}(D). \quad (4)$$

The generalization ability of the individual D is measured referring to the criterion of linear discriminant analysis. Specifically, we use the within-class and between-class scatter distances of samples, $d_w(D)$ and $d_b(D)$, to measure the generalization ability. Suppose the samples x_1, x_2, \dots, x_N belong to L clusters, and $I_j, j = 1, 2, \dots, L$, is the index set of the samples in the j_{th} cluster, which has N_j samples. Let $M_0 = \frac{1}{N} \sum_{i=1}^N y_i$ and $M_j = \frac{1}{N_j} \sum_{i \in I_j} y_i$ be the mean of all samples and those in the j_{th} cluster. Then

$$d_w(D) = \frac{1}{L} \sum_{j=1}^L \frac{1}{N_j} \sum_{i \in I_j} (y_i - M_j)^T (y_i - M_j), \quad (5)$$

and

$$d_b(D) = \frac{1}{N} \sum_{j=1}^L N_j (M_j - M_0)^T (M_j - M_0), \quad (6)$$

where ‘ T ’ is the transpose operator. The generalization ability of D is then measured by

$$\zeta_g(D) = d_b(D)/d_w(D). \quad (7)$$

And the total fitness of D is defined as the summation of the performance accuracy term and the generalization ability term, i.e.

$$\zeta(D) = \zeta_a(D) + \zeta_g(D). \quad (8)$$

C. Generating New Individuals

Three genetic operators are used to generate new individuals: selection, crossover, and mutation. The selection is based on the relative fitness of individuals, i.e. the proportion of the fitness of an individual to the total fitness of the population determines how many times the individual will be selected as parent individuals. After evaluating all individuals, we select $S-I$ pairs of parent individuals from them,

where S is the size of the GA population. $S-I$ new individuals generated from these parent individuals together with the individual with the highest fitness in current generation form the population of the next generation.

The crossover operator is conducted under a given probability. If two parent individuals are not subjected to crossover, the one having higher fitness will be chosen into the next generation. Otherwise, two crossover points are randomly chosen, one of which is within the coefficient bits and the other is within the selection bits. These two points divide both parent individuals into three parts, and the second part is then exchanged between them to form two new individuals, one of which is randomly chosen as an individual in the next generation.

At last, each bit in the $S-I$ new individuals is subjected to mutation from ‘0’ to ‘1’ or reversely under a specific probability. After applying all the three genetic operators, we have a new population for the next GA iteration.

D. Constraining the Search Space

In existing EFE algorithms, no constraint is imposed upon the search space. Instead, they search blindly in the original space. This leads to the limitation of these EFE algorithms in search efficiency. Besides, recent researches on LDA show that the projection basis vectors containing rich discriminant information are very much likely to lie in a subspace of the original space (Yang & others 2005), such as the null space of $S_w = \frac{1}{N} \sum_{j=1}^L \sum_{i \in I_j} (x_i - M_j)(x_i - M_j)^T$, the within-class scatter matrix of samples, and the range space of $S_b = \frac{1}{N} \sum_{i=1}^L N_j (M_j - M_0)(M_j - M_0)^T$, the between-class scatter matrix of samples. Here, $M_j, j = 0, 1, \dots, L$, are calculated over the samples in the original space, i.e. x_1, x_2, \dots, x_N . Therefore, to further improve the search efficiency and the performance of the obtained projection basis vectors, some constraints are necessary for the search space.

Thanks to the linear combination mechanism used by DEFE, it is very easy for us to force the GA to search in a constrained space. Our method is to construct vectors by linearly combining the basis of the constrained search space, instead of the original space. Take $\text{null}(S_w)$, the null space of S_w , as an example. Suppose we want to constrain the GA to search in $\text{null}(S_w)$. Let $\{\alpha_i \in R^n | i = 1, 2, \dots, m\}$ be the eigenvectors of S_w associated with zero eigenvalues. They form a basis of $\text{null}(S_w)$. After obtaining a vector v via linearly combining the above basis, we have to calculate the basis of the orthogonal complement space of $V = \text{span}\{v\}$ in the constrained search space $\text{null}(S_w)$, but not the original space R^n (recall part A of this section). For this purpose, we first calculate the isomorphic space of V in R^m , denoted by $\hat{V} = \text{span}\{P^T v\}$, where $P = [\alpha_1 \alpha_2 \dots \alpha_m]$ is an isomorphic mapping. We then calculate a basis of the orthogonal complement space of \hat{V} in R^m . Let $\{\beta_i \in R^m | i = 1, 2, \dots, m-1\}$ be the obtained basis. Finally, we map this basis back into $\text{null}(S_w)$ through $\{\beta_i = P \beta_i \in R^n | i = 1, 2, \dots, m-1\}$. The below theorem demonstrates that $\{\beta_i | i = 1, 2, \dots, m-1\}$ comprise a basis of the orthogonal complement space of V in $\text{null}(S_w)$.

Theorem 1: Assume $A \subset R^n$ is an m dimensional space, $P = [\alpha_1 \alpha_2 \cdots \alpha_m]$ is an identity orthogonal basis of A , where $\alpha_i \in R^n, i = 1, 2, \dots, m$, and $P^T P = E_m$, the m dimensional identity matrix. For any $v \in A$, suppose $\hat{V} = \text{span}\{P^T v\} \subset R^m$. Let $\{\hat{\beta}_i \in R^m | i = 1, 2, \dots, m - 1\}$ be an identity orthogonal basis of the orthogonal complement space of \hat{V} in R^m . Then $\{\beta_i = P\hat{\beta}_i \in R^n | i = 1, 2, \dots, m - 1\}$ is a basis of the orthogonal complement space of $V = \text{span}\{v\}$ in A .

Proof: First of all, let us prove $\forall i \in \{1, 2, \dots, m - 1\}, \beta_i \in A$.

Since $\beta_i = P\hat{\beta}_i = \sum_{j=1}^m \beta_{ij} \alpha_j$, β_i can be represented by a basis of A . Thus $\beta_i \in A$.

Secondly, let us prove $A = U \oplus V$, where $U = \text{span}\{\beta_1, \beta_2, \dots, \beta_{m-1}\}$. This is to prove $\{\beta_1, \beta_2, \dots, \beta_{m-1}, v\}$ is a linear independent bundle.

Since $\beta_i = P\hat{\beta}_i$ and $P^T P = E_m$, we have $\beta_i^T \beta_j = \hat{\beta}_i^T \hat{\beta}_j$. But $\{\hat{\beta}_i | i = 1, 2, \dots, m - 1\}$ is an identity orthogonal basis. Thus, $\beta_1, \beta_2, \dots, \beta_{m-1}$ are orthogonal to each other.

Furthermore, $\forall i \in \{1, 2, \dots, m - 1\}, \beta_i^T v = (P\hat{\beta}_i)^T v = \hat{\beta}_i^T P^T v$. Because $\{\hat{\beta}_i | i = 1, 2, \dots, m - 1\}$ is an identity orthogonal basis of the orthogonal complement space of $\hat{V} = \text{span}\{P^T v\}$ in R^m , $\hat{\beta}_i^T (P^T v)$ should be 0. Therefore, $\beta_i^T v = 0$, i.e. β_i is also orthogonal to v .

To sum up, we get that $\{\beta_1, \beta_2, \dots, \beta_{m-1}, v\}$ is a linear independent bundle containing m orthogonal vectors in the m dimensional space A . Thus, $A = U \oplus V$.

Now, Theorem 1 is proven. **END.**

III. Experiments

In this section, face recognition experiments are conducted to assess the performance of the proposed DEFE algorithm in comparison with two classical face recognition techniques, Eigenfaces (Turk & Pentland 1991) and Fisherfaces (Belhumeur & others 1997), as well as a representative EFE algorithm, Evolutionary Pursuit (EP) (Liu & Wechsler 2000).

A. Testing Methodology

After obtaining the optimal projection basis vectors and projecting all samples onto them, we employ a minimum distance classifier to classify novel samples. The classification performance is measured in terms of the Top 1, Top 2, Top 3, Top 4, and Top 5 recognition rates.

Two widely-used face databases are used. One is the Yale database, which consists of 165 face images of 15 subjects. All the images are first clipped to retain only the face area and calibrated according to the eye centers and the average face shape. The final size of the images is 121 by 168. We design two kinds of experiments on this database: the lighting test and the hybrid test. In the lighting test, which aims to evaluate the robustness of the algorithms to illumination variations, we select out the images taken under the same lighting condition to form test sets and use the remaining images for training. As a result, we get three pairs of training and test sets. The three test sets correspond to the left light, the frontal light, and the right light. Each test set has

11 images. In the hybrid test, which is designed to assess the performance of the algorithms over various recognition conditions, one image is randomly selected from the images of each subject to form a test set of 15 images. Five different pairs of training and test sets are constructed to assess the algorithms.

The other database is the ORL database. It contains 400 images of 40 subjects. One image of each subject is randomly chosen to form a test set and the remaining 360 images are used as a training set. Again, five different pairs of training and test sets are constructed.

As for parameter settings, we set the GA parameters in experience learned from experiments. Specifically, the size of the GA population is 100, the crossover probability is 0.8, and the mutation probability is 0.01. Meanwhile, we constrain the GA to search in the null space of S_w , referring to (Yang & others 2005). In our experiments, the GA is set to evolve 50 generations because satisfactory results can be obtained already at that generation. This is also a tradeoff between the recognition performance and the time cost.

B. Experimental Results

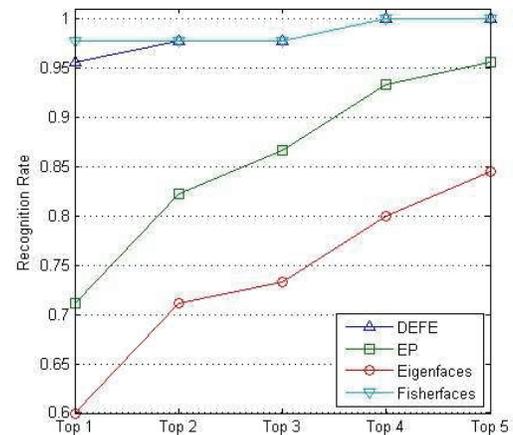


Figure 2: The recognition rates of DEFE, EP, Eigenfaces, and Fisherfaces in the lighting test on the Yale database.

We first evaluate the algorithms with the lighting test on the Yale database. In the experiments, we find that the EP algorithm can not be applied directly because of its high space complexity. Similarly with (Liu & Wechsler 2000), we use WPCA to reduce the data into a 30 dimensional space before applying the EP. As for Eigenfaces, we test its performance when using 20, 30, 40, 50, 60, 70, and 80 principal components and take the best result among them. Figure 2 shows the resulting recognition rates of the algorithms in the lighting test on the Yale database. The diagrams illustrate that the Top 1 recognition rate of the DEFE algorithm is a little

lower than that of Fisherfaces, but much higher than these of another two algorithms. As for the other recognition rates, the DEFE, the same as Fisherfaces, is better than the others.

We then conduct the hybrid test on the Yale database. The settings are the same as before and the results are given in Table 1. This time, both the DEFE and Fisherfaces achieve the best recognition rates of 100%, which are much better than these of Eigenfaces and the EP.

Table 1: The recognition rates of DEFE, EP, Eigenfaces, and Fisherfaces in the hybrid test on the Yale database.

Algorithm	Top 1	Top 2	Top 3	Top 4	Top 5
DEFE	100%	100%	100%	100%	100%
EP	91%	96%	97%	99%	99%
Eigenfaces	89%	91%	91%	93%	95%
Fisherfaces	100%	100%	100%	100%	100%

Finally, we compare the algorithms on the ORL database. The dimension of the WPCA transformed space in the EP is set to 60. In Eigenfaces, we use 20, 30, 40, 50, 60, 70, 80, 90, and 100 principal components respectively, and again, the best result among them is taken. Figure 3 displays the results. Apparently, the DEFE algorithm overwhelms all other three algorithms.

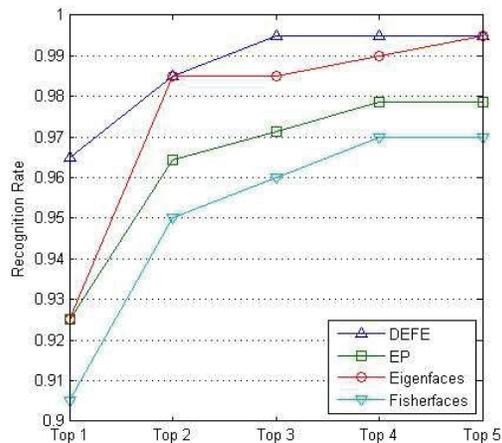


Figure 3: The recognition rates of DEFE, EP, Eigenfaces, and Fisherfaces on the ORL database.

According to above results, it is obvious that the proposed DEFE algorithm works excellently with high dimensional data, in which case the EP algorithm can not be directly applied at all. Moreover, the DEFE algorithm obtains the best recognition rates in almost all the tests, whereas the performance of the other three algorithms is not that stable. We suppose that it is the relatively small sample size compared with the high dimensionality of the data that leads to the

poor performance of the statistic methods, say Eigenfaces and Fisherfaces. All these results testify the feasibility and advantages of the DEFE algorithm proposed in this paper.

IV. Conclusions

With the aims of reducing the space complexity of evolutionary feature extraction algorithms and improving the search efficiency of them, we propose in this paper a direct evolutionary feature extraction algorithm. Not only can it be directly applied to high dimensional data, but also achieves an excellent classification performance. As a result, it successfully enhances the overall performance of EFE algorithms and provides a promising feature extraction method.

In Summary, the proposed direct evolutionary feature extraction algorithm mainly has the following two fine properties: (1) it reduces the space complexity of EFE algorithms by an order of magnitude, from $O(n^2)$ to $O(n)$; (2) it supplies a simple and effective scheme to constrain the search space of EFE algorithms.

Acknowledgements

We would like to thank Dr. Jian Yang, Wangmeng Zuo, and Zhenhua Guo as well as other colleagues from our team for their valuable suggestions and selfless help on the preparation of this paper. Our sincere thanks also go to the anonymous reviewers for their helpful suggestions on improving this paper.

References

- Belhumeur, P., et al. 1997. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 19(7):711–720.
- Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*. Academic Press, Inc., second edition.
- Goldberg, D. 1989. *Genetic Algorithm in Search, Optimization, and Machine Learning*. Adison-Wesley.
- Liu, C., and Wechsler, H. 2000. Evolutionary pursuit and its application to face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 22(6):570–582.
- Messer, K., et al. 2004. Face authentication test on the BANCA database. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, 523 – 532.
- Pei, M., et al. 1995. Genetic algorithms for classification and feature extraction. Presented at the Annual Meeting of the Classification Society of North America.
- Raymer, M., et al. 2000. Dimensionality reduction using genetic algorithms. *IEEE Transaction on Evolutionary Computation* 4(2):164–171.
- Scholkopf, B., et al. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation* 10(5):1299–1319.
- Siedlecki, W., and Sklansky, J. 1989. A note on genetic algorithms for large scale feature selection. *Pattern Recognition Letters* 10:335–347.

- Smith, M., and Bull, L. 2003. Feature construction and selection using genetic programming and a genetic algorithm. In *Lecture Notes on Computer Science 2610*, 229–237.
- Turk, M., and Pentland, A. 1991. Eigenfaces for recognition. *Journal of Cognitive Neuroscience* 13(1):71–86.
- Vafaie, H., and De Jong, K. 1995. Genetic algorithms as a tool for restructuring feature space representations. In *Proceedings of Seventh International Conference on Tools with Artificial Intelligence*, 8–11.
- Vafaie, H., and De Jong, K. 1998. Feature space transformation using genetic algorithms. *IEEE Intelligent Systems and Their Applications* 13(2):57–65.
- Yang, J., et al. 2005. KPCA plus LDA: A completed kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 27(2):230–244.
- Zhao, Q., and Lu, H. 2005. GA-driven LDA in KPCA space for facial expression recognition. In *Lecture Notes on Computer Science 3611*, 28–36.
- Zhao, W., et al. 2003. Face recognition: A literature survey. *ACM Computing Surveys* 35(5):399–458.
- Zhao, Q., et al. 2006. A fast evolutionary pursuit algorithm based on linearly combining vectors. *Pattern Recognition* 39:310–312.
- Zheng, W., et al. 2005. GA-Fisher: A new LDA-based face recognition algorithm with selection of principal components. *IEEE Transaction on Systems, Man, and Cybernetics - Part B: Cybernetics* 35(5):1065–1078.