

A Genetic Programming-PCA Hybrid Face Recognition Algorithm

Behzad Bozorgtabar, Gholam Ali Rezai Rad

School of Electrical Engineering, Iran University of Science and Technology, Tehran, Iran.
Email: b_bozorgtabar@elec.iust.ac.ir, Rezai@iust.ac.ir

Received June 11th, 2011; revised July 24th, 2011; accepted August 5th, 2011.

ABSTRACT

Increasing demand for a fast and reliable face recognition technology has obliged researchers to try and examine different pattern recognition schemes. But until now, Genetic Programming (GP), acclaimed pattern recognition, data mining and relation discovery methodology, has been neglected in face recognition literature. This paper tries to apply GP to face recognition. First Principal Component Analysis (PCA) is used to extract features, and then GP is used to classify image groups. To further improve the results, a leveraging method is also utilized. It is shown that although GP might not be efficient in its isolated form, a leveraged GP can offer results comparable to other Face recognition solutions.

Keywords: Face Recognition, Principal Component Analysis, Genetic Programming, Leveraging Algorithm

1. Introduction

Face recognition has become one of the most active research areas of pattern recognition since the early 1990s. In the past 20 years, significant advances have been made in design of successful classifier for face recognition [1]. However the diversity of the face patterns makes it difficult to create robust recognition systems and the complexity of the algorithms makes them hard to implement.

Principal components analysis (PCA) method [2], which is the base of well-known face recognition algorithm, Eigenfaces [3,4], is an appearance-based technique used widely for the feature extraction and has recorded a great performance in face recognition. PCA based approaches typically include two phases: training and classification. In the training phase, an eigenspace is established from the training samples using PCA and the training face images are mapped to the eigenspace for classification. In the classification phase, an input face is projected to the same eigenspace and classified by an appropriate classifier, such as Support Vector Machines (SVMs) or Neural Networks [5].

Genetic programming is an evolutionary algorithm methodology inspired by biological evolution [6]. Evolutionary algorithms create a population of abstract representations of candidate solutions, which is evolved using biology inspired operators such as selection, cross-over

and mutation towards better solutions.

In recent years, Genetic Programming and other evolutionary algorithms has been used in classification and pattern recognition problems [7,8], although to the authors' knowledge, Genetic Programming has never been used in Face Recognition Domain.

In many applications, Genetic programming yields simplified symbolical representation of the underlying system it tries to model. This leads to efficient checking of a new sample [9]. On the other hand the complexity and the time needed to find such representation discourages its use in many applications.

Leveraging algorithms are a group of deterministic algorithms where a set of weak learners are used to create a strong learner [10]. While it is not algorithmically constrained, most leveraging algorithms iteratively employ weak learners based on a distribution and combine them with weighting to form a final strong learner.

In this paper, Genetic Programming is utilized to classify face images. As images are usually large, PCA is used to extract image features and thus reduce data dimension. The Genetic Programming is then applied to the extracted features. Using a training group, Genetic Programming discovers possible relationship between the extracted features, which is in turn used to classify new images. To improve results, a leveraging scheme is introduced, which employs Genetic Programming as a weak learner, and combine results of several Genetic

Programming classifications as a single strong classifier.

The rest of paper is organized as follows: in Sections 2 and 3, PCA and Genetic Programming are introduced respectively. Section 4 presents the introduced algorithm, where Genetic Programming is used with and without leveraging. In Section 5, simulations are done on a selected face database and results are compared to previous studies.

2. Feature Extraction

Principal Component Analysis

Let there be R face images in the training set, where each image X_i is a 2-dimensional array of size $m \times n$ of intensity values. The image X_i can be converted into a vector of D (where $D = m \times n$) pixels. The rows of pixels of the image are placed one after another to form the vector.

If the training set of R images is defined by $X = (X_1, X_2, \dots, X_R)$, then the covariance matrix is defined as:

$$\begin{aligned} \Gamma &= \frac{1}{R} \sum_{i=1}^R (X_i - \bar{X})(X_i - \bar{X})^T \\ &= \Phi \Phi^T \end{aligned} \quad (1)$$

where $\Phi = (\Phi_1, \Phi_2, \dots, \Phi_R) \subset \mathfrak{R}^{D \times R}$ and

$$\bar{X} = \frac{1}{R} \sum_{i=1}^R x_i \quad (2)$$

is the mean image of the training set. Also the dimension of the covariance matrix is $D \times D$.

The eigenvalues and eigenvectors are then calculated from the covariance matrix.

Let $Q = (Q_1, Q_2, \dots, Q_r) \subset \mathfrak{R}^{D \times R}$ (generally, $r < R$) be the r normalized eigenvectors corresponding to r largest eigenvalues. Each of the r eigenvectors is called an Eigenface. Now, each of the face images of the training set X is projected into the Eigenface space to obtain its corresponding Eigenface based feature $Z_i \subset \mathfrak{R}^{D \times R}$, which is defined as:

$$Z_i = QTY_i, i = 1, 2, \dots, R \quad (3)$$

where Y_i is the mean-subtracted image of X_i [11].

3. Genetic Programming

Genetic programming is a methodology inspired by biological evolution to find equations, computer programs, analog circuits or in general any suitable structure for a predefined problem [9]. Genetic programming's general mechanisms are almost identical to genetic algorithms, as genetic programming is considered either a specialized form of genetic algorithms or an expansion of it [6]. Genetic programming is usually implemented similar to the following algorithm:

1) Create initial population. Individual solutions (call-

ed chromosomes) are usually generated randomly.

2) Evaluate the fitness of each individual in the population.

3) Select best-ranking individuals to reproduce.

4) Breed new generation through crossover and/or mutation (genetic operations) and give birth to offspring.

5) Repeat from step 2 until a termination condition is reached (time limit or sufficient fitness achieved).

Figure 1 illustrates the general Genetic programming algorithm.

In GP individual population members (chromosomes) are not fixed length linear character strings that encode possible solution to the problem like in GA, but they are programs that, when executed, provide solution to the problem.

These programs are expressed in GP as parse trees of varying sizes and shapes, what makes these methods flexible in their application to the wide range of problems.

The difference in chromosomes representation is the main and almost the only difference between this method and GA. The overall Darwinian idea of survival stays the same, but there are changes in mutation and crossover operators and in fitness function calculation.

Each node of tree can be function, operator, variable or constant number. Trees can be evaluated in a recursive manner, in which each node's operator or function is executed up on the results of its children's evaluation. Tree structure can easily represent a mathematical equation or a Turing complete program.

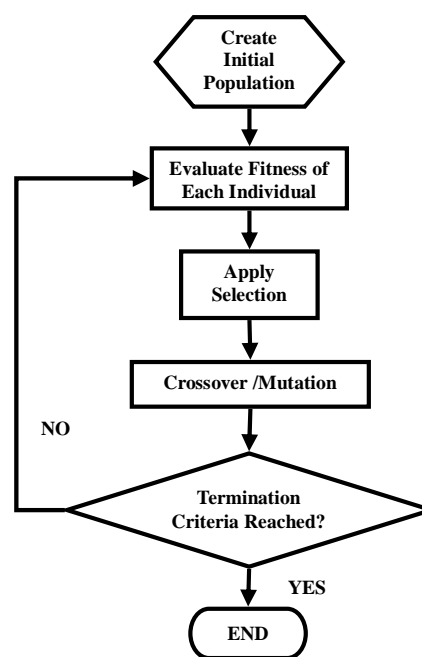


Figure 1. Genetic programming's flowchart.

4. Classification Algorithm

4.1. Using Genetic Programming

To classify a given dataset, it is usually enough to find a way for differentiating classes. Using genetic programming, this translates to finding a function which outputs a unique value for each different class:

$$f(X) = \begin{cases} 0 & X \in C_0 \\ 1 & X \in C_1 \\ \dots & \dots \\ n & X \in C_n \end{cases} \quad (4)$$

This is proven to be difficult. As a result, genetic programming is used to find a function per class that can discriminate only a certain class from others:

$$f_i(X) = \begin{cases} 1 & X \in C_i \\ 0 & X \notin C_i \end{cases} \quad (5)$$

This method creates N different functions for a total of N classes. Test images are tested one by one against the functions, and the first function to return a non-zero value is used to determine the image's class.

4.2. Leveraging Algorithm

Leveraging is a method of using multiple results to improve detection. A leveraging algorithm employs multiple weak classifiers to create a strong classifier. The following leveraging algorithm is used in this paper: Instead of using all training images as input, the whole group is partitioned to k different groups. **Figure 2** shows sample face images which are partitioned to three various clusters.

Detector function $f_{i,j}$ is then obtained as a function which can detect class i from other classes in group j . To further improve the results of classification, algorithm above could be repeated N times. For a given image X , the following equation creates the results of classification:

$$C = \sum_{j,n} f_{j,n}(p) \times \frac{1}{1 + \frac{1}{N} err_{j,n}} \quad (6)$$

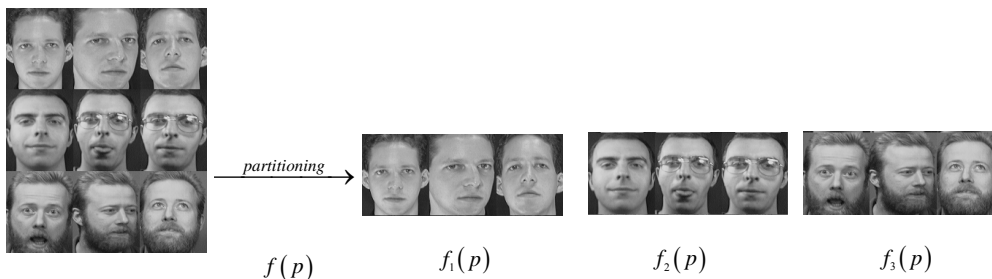


Figure 2. Partitioning sample AT&T face database.

where $f_{j,n}$ is result of n th iteration on the j th group, n is the iteration number from total N repetitions, and $err_{j,k}$ is sum of total errors for all images in the training group.

To determine a new image's class, all values acquired from (6) are compared. The class which yields the greatest C is nominated as the new candidate class. It should be noted that a threshold could be defined, as if the results of all classifiers for an image yield lower than a certain value, the image is certainly misclassified.

5. Simulation and Results

The algorithms were implemented in Python and then were tested on the AT&T face image database [12]. The AT&T database consists of 40 groups, each containing ten 112×92 gray scale images of a single subject. Each subject's images differ in lighting, facial expression and details (smiling/frowning, glasses/no glasses, etc.).

Two set of images were created from the AT&T database; For the Five-to-Five dataset, five random images of each group were selected for training while the others were used for testing. For the Leave-One-Out set, 9 images were used for training and the remaining image was kept for validation.

First Genetic Programming was tested without leveraging. To evolve the population, an Evolutionary Strategy (ES) of $1 + \lambda$ with $\lambda = 4$ was chosen. Mutation rate was set to 15 percent.

The selected function set was $\{+, -, \times, <, >, MIN, MAX, AND, OR, NOT, CNST\}$ where Boolean operators first compare their operands with 0 and $CNST$ returns a random constant floating point number in range of $[-10, 10]$. Inputs were chosen from all available PCA results. To limit algorithm time and prevent bloat, each chromosome's depth was limited to 25 and a maximum of 20000 iterations for each evolution was maintained.

To test the leveraged algorithm, algorithm was executed with the same parameters. Also the number of iterations was set to $N = 8$, while the set was divided to $k = 8$ different groups.

Table 1 shows a few of discovered relationship functions for a set of pictures. It could be seen that the generated formulas are often simple while only depending on a

Table 1. Examples of Acquired Relationship Functions for Detecting Image Group 1. $PCA[n]$ Is the Nth Value on PCA Vector.

No	Function
1	$(PCA[8] - MAX(PCA[19], PCA[7])) > PCA[12]$
2	$AND((PCA[2] < PCA[13]), MIN(PCA[1], 1))$
3	$(PCA[0] \times NOT(PCA[5]))$
4	$(PCA[1] \times (PCA[21] > (PCA[2] - PCA[18])))$

few components and as a result have a relatively low computational overhead.

Results are brought in **Table 2**, where they are compared to EigenFace [3] and SVM [13] clustering methods. It is observed that Genetic Programming without leveraging has the worst results. On the other hand, Leveraged Genetic programming beats other methods in Five-to-Five. In leave-one-out the results are repeated for Genetic Programming, although this time Leveraged Genetic Programming fell %2.5 (one image in total of 40 images) short of SVM.

While it could be inferred that the Genetic Programming is usable as a feature detector, it is believed that PCA is limited in reducing data dimension [14]. Further research is required to investigate Genetic Programming's results with 2D PCA and Multilinear PCA.

To further investigate Genetic Programming's performance, number of partitioned class groups was changed and the results were brought in **Table 3**. It was observed that the further partitioning of the images increases recognition error, while decreasing k might mandates increase in time spent for Genetic Programming's evolution.

6. Conclusions

Genetic programming is a general purpose search algorithm that can be utilized in classification problems. In this paper, Genetic programming was exploited to classify face images. The results showed that Genetic Programming alone is not suitable, as required time and computational overhead surpasses that of other methods, and also its recognition ratio is usually lower.

Table 2. Comparison of Different Algorithms Recognition Rate.

Method	Five-to-Five	Leave One Out
Eigenface	87.0%	85.0%
SVM	91.0%	95.0%
GP	63.5%	67.5%
Leveraged GP	91.5%	92.5%

Table 3. Effect of Number of Partitions in Leveraged Genetic Programming on Recognition Rate.

Number of Partitions	Recognition Rate
2	88.5%
4	91.5%
5	91.5%
8	91.0%
10	89.0%

To improve results, a leveraging algorithm was applied to Genetic Programming. The leveraged Genetic Programming showed a good recognition rate, comparable to or in some cases even better than that of other methods.

REFERENCES

- [1] S. Liu, Y. Tian and D. Li, "New research Advances of Facial Expression Recognition," *International Conference on Machine Learning and Cybernetics*, Baoding, Vol. 2, July 2009, pp. 1150-1155.
- [2] I. T. Jolliffe, "Principal Component Analysis," Springer-Verlag New York, Inc., 2002.
- [3] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86.
- [4] A. Pentland, B. Moghaddam and T. Starner, "View-Based and Modular Eigenspaces for Face Recognition," *Proceedings CVPR'94, 1994 IEEE Computer Society Conference on*, Seattle, July 1994, pp. 84-91.
- [5] A. Eleyan and H. Demirel, "PCA and LDA Based Face Recognition Using Feedforward Neural Network Classifier," *Lecture Notes in Computer Science*, Vol. 4105, 2006, pp. 199-206.
[doi:10.1007/11848035_28](https://doi.org/10.1007/11848035_28)
- [6] J. R. Koza, "Genetic Programming: On the Programming of Computer by Means of Natural Selection," MIT Press: Cambridge, 1992.
- [7] S. Xuesong and Y. Zhou, "Gray Intensity Images Processing for PD Pattern Recognition Based on Genetic Programming," *International Joint Conference on Artificial Intelligence JCAI'09*, Haikou, 2009, pp. 711-714.
- [8] A. Teredesai and V. Govindaraju, "Issues in Evolving GP Based Classifiers for a Pattern Recognition Task," *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, 20-23 June 2004, pp. 509-515.
- [9] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu and G. Lanza, "Genetic Programming IV: Routine Human-Competitive Machine Intelligence," Kluwer Academic Publishers, Norwell, 2003.
- [10] N. Krause and Y. Singer, "Leveraging the Margin More Carefully," *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, 2004, p. 63.

- [11] J. K. Sing, S. Thakur, D. K. Basu and M. Nasipuri, "Direct Kernel PCA with RBF Neural Networks for Face Recognition," *IEEE TENCON Region 10 Conference*, Hyderabad, 2008, pp. 1-6.
- [12] AT&T, "The Database of Faces," 2011.
<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedata/e.html>.
- [13] Y. Q. Pan and Y. Liu, "Face Recognition Using Kernel PCA and Hybrid Flexible Neural Tree," *Proceedings of the 2007 International Conference on Wavelet Analysis and Pattern Recognition*, Beijing, 2-4 November 2007, pp. 1361-1366.
- [14] J. Wang, Y. Chen and M. Adjouadi, "A Comparative Study of Multilinear Principal Component Analysis for Face Recognition," *37th IEEE Applied Image Pattern Recognition Workshop*, 2008, pp. 1-6.
[doi:10.1109/AIPR.2008.4906476](https://doi.org/10.1109/AIPR.2008.4906476)