MATLAB-based Design and Realization of Face Recognition System

Jianming Liu

College of Mathematics and Computer Science, Quanzhou Normal University, Quanzhou 362000, Fujian, P. R. China

Key Laboratory of Intelligent Computing and Information Processing, Fujian Province University, Quanzhou Normal University, Quanzhou 362000, Fujian, P. R. China

liujmcqu1999@sina.com

Abstract

Through MATLAB, the paper makes a comparison between Principal Component Analysis (PCA)face recognition algorithm and Adaboost recognition algorithm and selects the algorithm with higher recognition rates to develop an auto face recognition system. The paper explicates primary techniques the system adopts and its specific realization process. By downloading face database online, the paper conducts an all-round test to the system, the result of which proves that this face recognition system is completely practical and feasible.

Keywords

MATLAB, face recognition, PCA, SVM, AdaBoost.

1. Introduction

With the thriving development of computer technologies and bio-medical project technologies[1], the biometric recognition technology utilizing human body’s characteristics to identify and recognize people’s identity has become a prime choice approach to security check. Biometric recognition technology is a new kind of recognition technology based on information technology and biometric technology, with close integration of such high technologies as computer technology, acoustics, optics, biosensor, bio statistics to recognize people’s identity through their intrinsic physical traits (fingerprints, face recognition, iris recognition, etc.) or behavioral traits (chirography, gait, voice, etc.). Similarly, face recognition technology is defined as a kind of technology which will analyze relevant face photo by using recognition software programmed by computers and then pick up useful recognition information of face characteristics so as to distinguish and verify the identified object. Its widespread application shows great prospect. For example, it can be applied to archive management, financial system, security check system, customs surveillance system, automatic access control system, etc. Bright development prospects, convenient and easy control, increasing acceptance, high deceptiveness prevention and security.These are all the advantages of face recognition technology, which make it increasingly become a prevalent research issue in current fields of pattern recognition and artificial intelligence. Generally, a face recognition system consists of four main technological elements: face testing, face characteristics extracting, face verifying and recognition results submitting.

2. Structure of Recognition System

2.1 Software and Hardware Platform Components of System[2]

(1) Common personal computers or desktop computers with high performance and with Genuine Intel（R） T1600,1.66GHz,1024MB Memory. Usb 2.0 interface should be accompanied .

(2) HD usb 2.0 computer cameras.

(3) windows xp operation system

(4) Matlab 7.0 language translation platform

2.2 Basic System Framework

The main framework and structure of face recognition system is shown in Figure 1. First, build or download face database online. And then, based on face testing algorithm, extract face images by excluding their background and pre-process them, extract their characteristics and train BP neural networks to form a characteristics database to identify face. Input face images to the system, which also needs face testing, face characteristics positioning and extracting, using characteristics database to verify, and then output face recognition results.

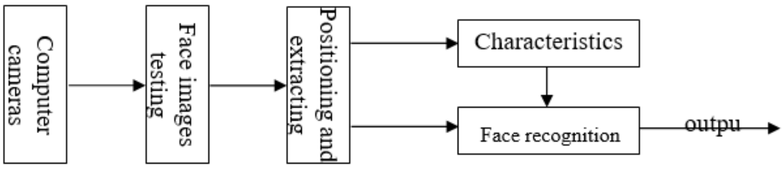


Figure1 framework and structure of face recognition

3. Algorithm Design and Realization

3.1 PCA Algorithm Design and Realization

3.1.1. Matlab Realization of Fast PCA

The paper uses Matlab translation platform to realize face recognition system, in which fast PCA code is: function [pcaA V] = fastPCA(A, k)

```
% Fast PCA
% Input: A ---sample matrix, each row as a sample
% k --- reduce dimension to Dimension k
% Output: pcaA --- Matrix consisting of Dimension k feature vector after dimension reduction, each row as a sample, Column k as sample feature dimension number after dimension reduction
% V --- primary component vector
[r c] = size(A);
%  mean value of Samples
meanVec = mean(A);
% compute conversion of covariance matrix :cov Mat T
Z = (A-repmat(meanVec, r, 1));
covMatT = Z * Z';
% compute previous k eigenvalues and eigenvectors
[V D] = eigs(covMatT, k);
%  extract eigenvectors of covariance matrix
V = Z' * V;
% normalize eigenvectors to unit eigenvectors
for i=1:k
   V(:,i)=V(:,i)/norm(V(:,i));
end
% Through linear transformation(projection), reduce dimension to Dimension k
pcaA = Z * V;
% save transformation matrix V and transformation origin mean Vec
save('Mat/PCA.mat', 'V', 'meanVec');
```

3.1.2. Primary Components Decomposition Steps by Using PCA

In PCA, the main thing it to ask for the largest transformation direction of variances, whose specific decomposition steps are as follows:

(1)construct relational matrix: $Cx = E[x * x^T]$, $Cx \in P^{n \times n}$

In practical application, since it's not easy to decompose expected value of original data, we can use the following formula to approximately construct relational matrix:

$$Cx = [x_1 x_1^T + x_2 x_2^T + \cdots + x_N x_N^T]/N$$

($x_1, x_2, \cdots x_N$ are vectors equivalent to all pixel dots of every original gray images, while N is the number of original images)

(2)compute each eigenvalue of Cx

(3) order eigenvalues in numerical size sequence $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_N \geq 0$

(4)compute orthogonal eigenvectors equivalent to previous m eignvalues to form w

(5)project original data on the eigenvector w and get the principal eigen data of original images

For example, Figure 2 are principal component faces of PCA from 200 pictures with 20 dimensions of eigens and 20 faces.

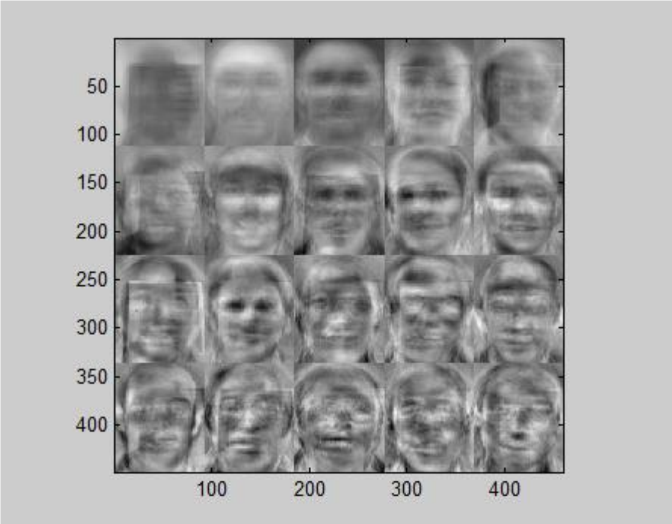


Figure2 Principal Component Faces of Samples

3.2 Data Normalization

Honestly speaking, data normalization plays a significant role in mode recognition problem, which can even decide whether the whole mode recognition system will be succeed or not. The paper uses maximum-minimum normalization with explanations as follows:

The method conducts a kind of linear transformation towards original data. We denote $\min_A$ and $\max_A$ as Minimum and Maximum Value of Attribute A. Through max-min normalization, one value v of Attribute A will be projected as $v1$ and $v1 \in [new\_\min_A, new\_\max_A]$.

Specific projection is as following:

$$v1 = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A$$

Data normalization of face recognition system in the paper is complemented by Matlab, whose principal codes are:

function [SVFM, lowVec, upVec] = scaling(VecFeaMat, bTest, lRealBVec, uRealBVec)

% Input: VecFeaMat --- m*n dimension data matrix needing scaling, each row as a sample eigenvector and each column as dimensions

```
% bTest--=1 means performing scaling on testing samples, meanwhile, lRealBVec and uRealBVec should be provided.
%=0: default value, by performing scaling on training samples
% lRealBVec ---Dimension n vector, get actual lower limit information of every dimension by performing scaling on training samples
% uRealBVec --- Dimension n vector, get actual upper limit information of every dimension by performing scaling on training samples
% output: SVFM --- scaling version of VecFeaMat
% upVec --- upper limit of every dimension feature (significant only when performing scaling on training samples, bTest = 0)
% lowVec --- lower limit of every dimension feature(significant only when performing scaling on training samples, bTest = 0)
if nargin < 2
    bTest = 0;
end
% scale target range[-1, 1]
lTargB = -1;
uTargB = 1;
[m n] = size(VecFeaMat);
SVFM = zeros(m, n)
if bTest
    if nargin < 4
        error('To do scaling on testset, param lRealB and uRealB are needed.');
    end
    if nargout > 1
        error('When do scaling on testset, only one output is supported.');
    end
    for iCol = 1:n
        if uRealBVec(iCol) == lRealBVec(iCol)
            SVFM(:, iCol) = uRealBVec(iCol);
            SVFM(:, iCol) = 0;
        else
            SVFM(:, iCol) = lTargB  +  ( VecFeaMat(:, iCol) - lRealBVec(iCol) ) / ( uRealBVec(iCol)-lRealBVec(iCol) ) * (uTargB-lTargB); % testing samples scaling
        end
    end
else
    upVec = zeros(1, n);
    lowVec = zeros(1, n);
    for iCol = 1:n
        lowVec(iCol) = min( VecFeaMat(:, iCol) );
        upVec(iCol) = max( VecFeaMat(:, iCol) );
        if upVec(iCol) == lowVec(iCol)
```

```
            SVFM(:, iCol) = upVec(iCol);
            SVFM(:, iCol) = 0;
        else
            SVFM(:, iCol) = lTargB + ( VecFeaMat(:, iCol) - lowVec(iCol) ) / ( upVec(iCol)-lowVec(iCol) ) * (uTargB-lTargB); % training samples scaling
        end
    end
End
```

3.3 Realization of SVM Vector Machine in Matlab

Matlab 7.0 can support SVM vector machine, whose SVM toolkit is set by installing SVM's training and classification functions through these two functions svmtrain and svmclassify.

Before opening SVM, two problems need to be settled:

(1)A kind of kernel function needs to be selected.

(2)Parameter of the kernel function and optimum value of coefficient C of error cost need to be determined.

there are four common kernel functions:

Linear kernel function: $K(x,y)=x\cdot y$, Polynomial kernel function: $K(x,y)=(x\cdot y+1)^d$

Radial basis function: $K(x,y)=\exp(-\gamma\|x-y\|^2)$ Sigmoid kernel function: $K(x,y)=\tanh(b(x\cdot y)-c)$ The paper selects radial basis function (RBF), for the reasons:

(1) As a kernel function corresponding to non-linear projection, RBF can deal with non-linear separable situations.

(2) Linear kernel function is a special situation of RBF, which means by selecting appropriate parameter (γ,C), RBF can always get the results same to linear kernel function with coefficient of error cost, but not vice versa.

(3) While selecting some parameters, Sigmoid kernel function is similar to RBF, but the former needs to determine two related parameters b and c.

(4) Polynomial function needs to compute interior product, which may lead to overflow computing problem.

Since RBF in the toolkit cannot flexibly select parameter, so we need to compile RBF. The paper compiles it by using Matlab, whose main codes are:

```
function K = kfun_rbf(U, V, gamma)
[m1 n1] = size(U);
[m2 n2] = size(V);
K = zeros(m1, m2);
for ii = 1:m1
    for jj = 1:m2
        K(ii, jj) = exp( -gamma * norm(U(ii, :)-V(jj, :))^2 );
    end
End
```

3.4 Adaboost Realization

Adaboost algorithm [3] can improve self-adaptively a group of weak classifiers to strong classifiers. Since the output of weak classifiers is +1and -1, it can be called Discrete Adaboost. Adaboost algorithm introduces a weight w_i for every training and makes selection of next iterative weak classifiers pay more attention to wrong samples.

Main steps are as follows:

(1)Initialize the weighted value of every sample $w_i = \frac{1}{N}, i = 1, 2, \ldots N$

(2)Use weak classifier learning algorithm weak classifier to ever y sample $f_t(x) \in \{-1, 1\}$, and calculate error rate $\varepsilon_t = E_\omega[I(y \neq f_t(x))]$ and weighted coefficient $a_t = \log[(1-\varepsilon_t)/\varepsilon_t]$

(3)Renew weighted value $\omega_i = \omega_i \exp[a_t \cdot I(f_t(x_i) \neq y_i)]$, and renormalize it to make $\sum_i w_i = 1$

(4)Output classifier $F(x) = \operatorname{sgn}[\sum_{t=1}^{T} a_t f_t(x)]$

To embed feature selections into discrete Adaboost algorithm, the paper designs simple threshold weak classifier to every dimension feature x_i , and the form of the classifier is $f(x_j) = a_j \cdot I(x_j \leq \theta_j) + b_j \cdot I(x_j > \theta_j)$.

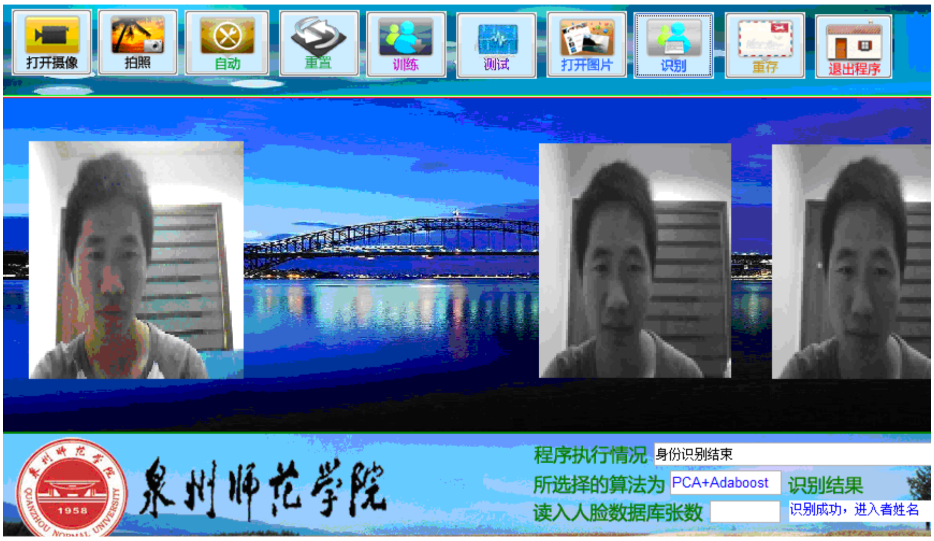
Given threshold of feature θ_j, get values of a_j and b_j by minimizing weighted classification error rate of samples. And the values are 1or-1, that is $a_j = \operatorname{sgn}\{E_\omega[y \cdot I(x_j \leq \theta_j)]\} = \operatorname{sgn}\{\sum_{i=1}^{N} \omega_i y_i \cdot I(x_{i,j} \leq \theta_j)\}$ or $b_j = \operatorname{sgn}\{E_\omega[y \cdot I(x_j > \theta_j)]\} = \operatorname{sgn}\{\sum_{i=1}^{N} \omega_i y_i \cdot I(x_{i,j} > \theta_j)\}$.Now, the error rate is

$$\varepsilon_t = E_\omega\{I[y \neq f(x_i)]\} = \sum_{i=1}^{N} \omega_i \cdot I[y_i \neq f(x_{i,j})]$$

The best segregation threshold value θ_j of the feature j can be got by exhaustion method. Since there are N samples, the searching complexity degree of the best θ_j of each feature is $O(N)$. Every time Adaboost chooses the optimum weak classifier, it is equivalent to choose optimum feature. Suppose there are M features, select feature with smallest error rate and computational complexity degrees of each iteration are $O(MN)$. When the algorithm performs T times of iteration, computational complexity degrees of the whole training procedure are $O(TMN)$.

4. Face Recognition System Design

The paper designs face recognition access control system based on GUI of MATLAB as Figure 3 and performs related simulation experiments [4] [5] [6] .



程序执行情况 身份识别结束

所选择的算法为 PCA+Adaboost

识别结果 识别成功，进入者姓名

读入人脸数据库张数

Figure 3: The operation interface of Face Recognition Access Control System

(a) Function Button Section: contains all function buttons the system needs.

(1) Open camera button and computer camera.

(2) Shoot button, take pictures and input them to face database for recognition later.

(3) Auto button, automatically take pictures and send them to recognition section to automatically realize the whole process of face recognition.

(4) Reset button, re-take pictures.

(5) Training button, save face images and train them into face feature database for comparison in recognizing.

(6) Testing button, test recognition rate of selected algorithm.

(7) Open image button, open a picture randomly in face database and perform simulation experiment.

(8) Recognize button, start recognizing images in recognition section.

(9) Re-save button, resave face images.

(10) Exit button, exit face recognition system.

(b) Real-time Practice Display Section: display real-time time.

(c) Image Display Section:

(1) Real-time face image display section, get face image to be recognized and send it into system to proceed.

(2) Face-to-be-recognized Display Section, display image after proceeding and begin to recognize.

(3) People's identification display section, display recognition results.

(d) Recognition State Section:

(1) Procedure Performance Situation, display recognize proceedings real-time.

(2) Selected Algorithm, display selected recognition algorithm.

(3) Recognition results, display written recognition results.

(4) read the face number in database, display the number of saved faces while saving pictures.

5. Conclusion

The paper conducts a research on current hot study topic—biological character recognition technology, which combines PCA, Adaboost and SVM and builds a face recognition access control system based on MATLAB. The system makes a simulation experiment on faces of twelve members, which proves that except some covered faces and grimaces, other situations can be identified effectively and in real time. Later, we do a test on situations in different illumination, sunlight, background and so on, which shows that except images under poor conditions, such as background close to skin color or too strong sunlight, the recognition rate can reach more than 90%. The system is easy to install and convenient to use, which is flexible and can be equipped with different hardware environment and used in different occasions based on customers' needs. However, since there are only two recognition algorithm adapted, the recognition rate is limited.

Acknowledgements

This research was supported by Fujian Provincial Key Laboratory of Data Intensive Computing and Key Laboratory of Intelligent Computing and Information Processing, Fujian Province University, and by Nature Science Research Pragramme of the Education Department of Fujian Province(JAT160410).

References

[1] Wang Yinghui.Face Recognition Theory Methodology and Technology[M]. Beijing,Science Press, 2010.

[2] Yuan Shu, Huang Xuechao,et al.Computing Configuration and Application of Matlab Parallel Machines under Windows[J]. Computers and Modernization,2010,51(5):189-194.

[3] Li Shengwen,Bao Susu.Face Recognition Technology Based on PCA+ AdaBoost Algorithm[J]. Computer Engineering and Application. 2010,46(4):170-173.

[4] Dong Huoming,Gao Jun,et al.Face Recognition and Identification Confirmation Combined with Multi-classifiers[J]． Systematic Simulation Journal, 2004, 16 (8):1849-1853.

[5] Li Junhua,Peng Li.Face Expression Recognition Based on Principal Component Analysis of Characteristic Module[J]. Computer Engineering and Design, 2008,29(12):151-153.

[6] Wang Min, Duan Liming, et al. Face Recognition Simulation System Study Based on MATLAB 7.0 [J]. Scientific Technology and Engineering, 2009,9(20):6224-6226.