Mood Detection Assistant

Mood based Music Player

Prashant Kumar Mahanta
Computer Science and Engineering, UG 2
Indian Institute of Information Technology
Sri City, A.P., India
prashantkumar.m16@iiits.in

Abstract*—In this paper, we use the principal component analysis (PCA) algorithm for the detection of facial expression. First the eigen spaces are created with the help of eigen vectors and eigen values. With the help of this space eigen faces are created and with the help of PCA algorithm the most matching eigen face is selected. The databases of 4 persons are generated each person having 25 photographs with different expressions like happy, angry, sad, neutral etc. The classifier used are based on Euclidian distance. Train and test databases are there but that should be in similar conditions such as size, background, etc. The result is, the music player plays music based on your mood, the songs are decided by user before.

Keywords

Principal component analysis, Eigen vector, Eigen values.

I. INTRODUCTION

The human face is an extremely complex and dynamic structure with characteristics that can quickly and significantly change with time. It is the primary focus of attention in social relationships and plays a major role in the transmission of identity and emotions. Therefore, face recognition is applied in many important areas such as security, identification of criminals and so on. Here we have developed a music player which plays music based on the user's mood. The aim of this paper is to study and develop an efficient MATLAB program for face recognition using principal component analysis and to perform test for program optimization and accuracy. This approach is preferred due to its simplicity, speed and learning capacity [1].

II. FACE RECOGNITION PROCESS

One of the simplest and most effective PCA approaches used in face recognition systems is called eigen face approach. This approach transforms faces into a small set of essential characteristics, eigenfaces, which are the main components of the initial set of learning images (training set). Recognition is done by projecting a new image in the eigenface subspace, after which the person is classified by comparing its position in eigenface space with the position of known individuals [2].
The problem is limited to files that can be used to recognize the face. Namely, the images must be vertical frontal views of human faces. The whole recognition process involves two steps:

A. Initialization process

B. Recognisation process

The Initialization process involves the following operations:
i. Acquire the initial set of face images called as training set.
ii. Calculate the Eigenfaces from the training set, keeping only the highest eigenvalues. These M images define the face space.
iii. Calculate distribution in this M-dimensional space for each known person by projecting his or her face images onto this face-space.

The Recognisation process involves the following operations:

i. Projecting centered image vectors into face space. All the centered images are projected into face-space by multiplying in Eigenface basis.
ii. Projecting the test image in the same face-space.
iii. Measure the Euclidian distance between projected training images and projected test image in the face-space.
iv. The test image is supposed to have minimum distance with its corresponding image in the training database.

III. EIGENFACE ALGORITHM

Let a face image T (x, y) be a two-dimensional M by N array. In this thesis, we used a set of images 600 × 800 pixels. An image may also be considered as a vector of dimension M × N, so that a typical image of size 600 × 800 becomes a vector of dimension 48,000 or equivalently a point in a 48,000-dimensional space.

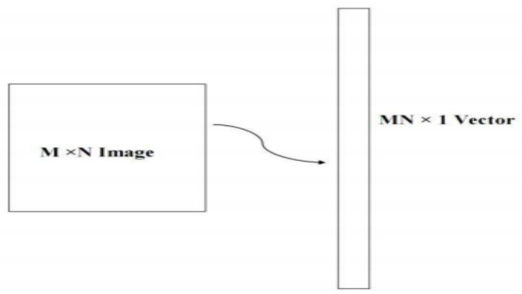


Fig-1: Conversion of M × N image into MN × 1 vector.

Step 1: Prepare the training faces

Obtain face images I_1, I_2, I_3, . . . I_q (training faces). The face images must be centered and of the same size.

Step 2: Prepare the data set

Each face image I_i in the database is transformed into a vector and placed into a raining set S.

$$S = \{T_1, T_2, T_3, T_4, \ldots T_q\}$$

In my example q = 100. Each image is transformed into a vector of size MN × 1 and placed into the set. For simplicity, the face images are assumed to be of size N × N resulting in a point in N^2 dimensional space. An ensemble of images, then, maps to a collection of points in this huge space.

Step 3: compute the average face vector

The average face vector (μ) has to be calculated by using the following formula:

$$\mu = 1/q \sum_{\iota=1}^{q} T_\iota \qquad (1)$$

Step 4: Subtract the average face vector. The average face vector μ is subtracted from the original faces T_i and the result stored in variable ϕ_i ,

$$\phi_i = T_i - \mu \qquad (2)$$

Step 5: Calculate the covariance matrix. We obtain the covariance matrix C in the following manner,

$$C = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T$$

$$= AA^T \qquad (N^2 \times N^2\ matrix \quad \text{Where,}$$

$$A = [\Phi_1, \Phi_2, \Phi_3, \Phi_4, \ldots \ldots \ldots \Phi_M$$

$$(N^2 \times M\ matrix)$$

Step 6: Calculate the eigenvectors and eigenvalues of the covariance matrix

The covariance matrix C in step 5 has a dimensionality of $N^2 \times N^2$, so one would have eigenface and eigenvalues. For a 600 × 800 image that means that on must compute a 48,000 × 48,000 matrix and calculate 48,000 eigenfaces. Computationally, this is not very efficient as most of those eigenfaces are not useful for our task. In general, PCA is used to describe a large dimensional space with a relative small set of vectors [3].

Compute the eigenvectors u_i of AA^T. The matrix AA^T is very large (i.e. 48,000 × 48,000) which is not practical to compute.

Step 6.1: Consider the matrix

$$L = A^TA \qquad (M\times M \quad \text{matrix})$$

Step 6.2: Compute eigenvector v_i of $L=AA^T$ $\quad AA^Tv_i=\mu_i v_i$

$AA^T A\, v_i = \mu_i\, Av_i$ [where $AA^T = C$ and $Av_i = u_i$]

$$C\, u_i = \mu_i\, Av_i$$

Where v_i is an eigenvector of $L=A^TA$ and u_i are eigenvectors[4].

Step 7: Keep only K eigenvectors (corresponding to the K largest eigenvalues). Eigenfaces with low eigenvalues can be omitted, as they explain only a small part of Characteristic features of the faces.

IV. TESTING SAMPLES CLASSIFICATIONS

a) Read the test image and separate face from it.

b) Calculate the feature vector of the test face. The test image is transformed into its eigenface components. First, we compare line of our input image with our mean image and multiply their difference with each eigenvectors [2]. Each value would represent a weight and would be saved on a vector Ω_T. $\quad \Omega_{test} = u_i^T(T_{test} - \mu)$ Where, u_i is the i^{th} Eigenfaces and i=1,2,3…..K.

c) Compute the average distance (Euclidean distance) between test feature vector and all the training feature vectors. Mathematically, recognition is finding the minimum Euclidean distance α_k , between a testing point and a training point given in the following equation.

$\alpha_k = (|\Omega_{test} - \Omega_i|)^{1/2}$ Where, i = 1, 2, 3. . . . . . K. The Euclidean distance between two weight vectors thus provides a measurement of similarity between the corresponding images.

d) The face class with minimum Euclidian distance shows similarity to test image [5].

V. SCHEMATIC DIAGRAM & FLOWCHART

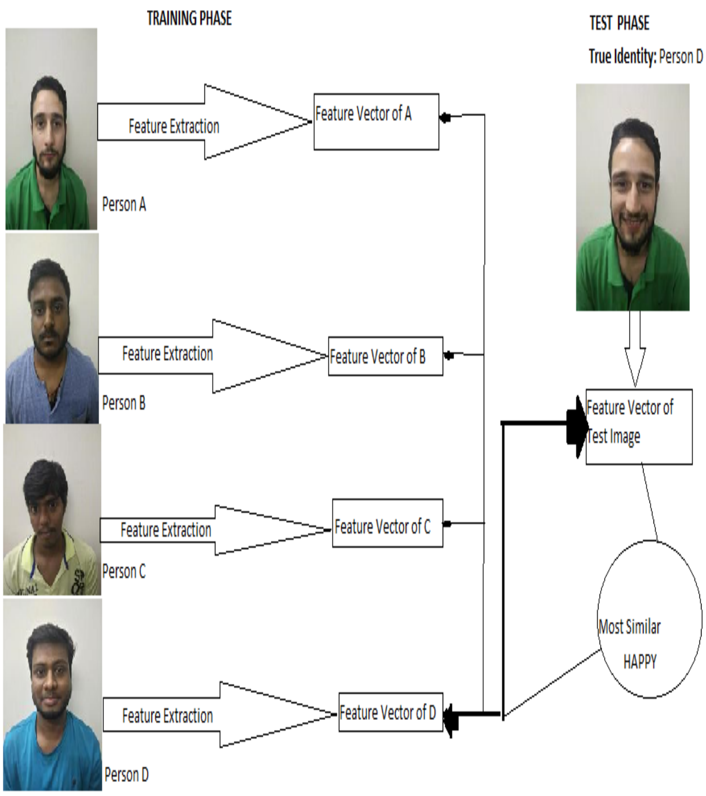
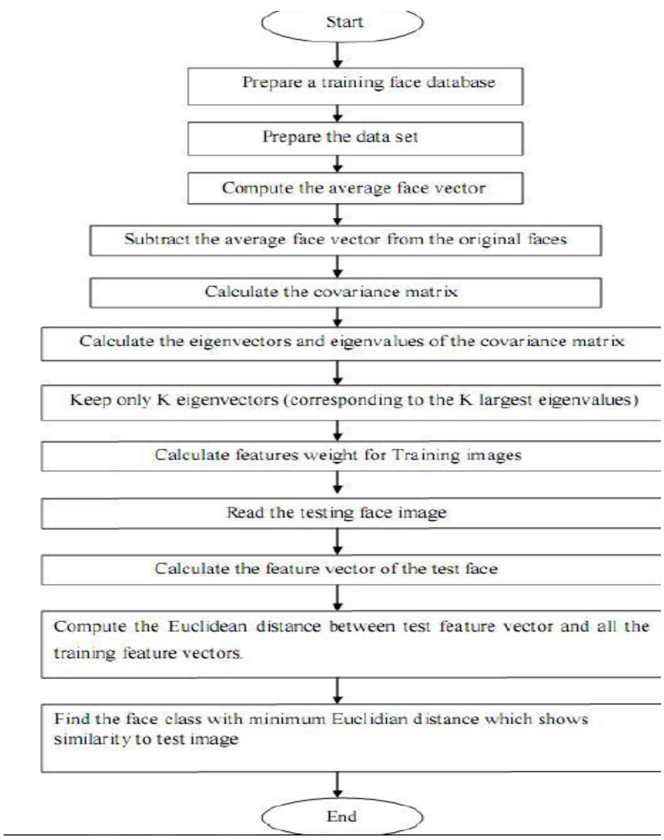


Fig 2: Schematic diagram of a face recognizer



VI. EXPERIMENTAL RESULTS

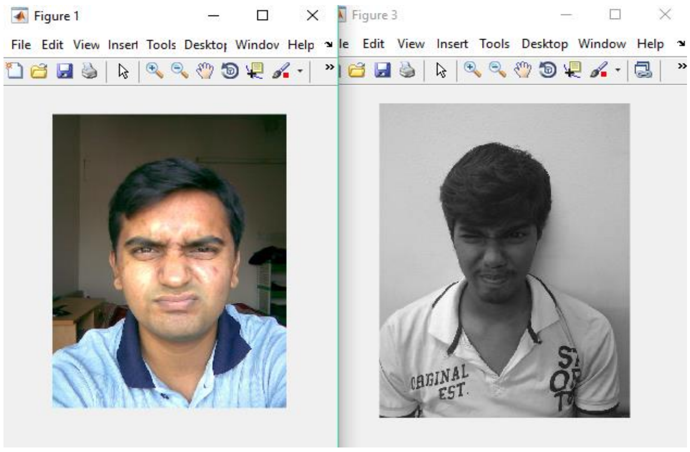


Fig 3: The Recognition of Individual Images having Same Pose

Principal component analysis approach to the face recognition problem was studied and a face recognition system based on the eigenfaces approach was proposed. The algorithm developed in a generalized one which works well with any type of images. The tests conducted on PNG images and JPG images of various subjects in different poses showed that this method gave very good classification of faces though it has limitations over the variations in size of image. The eigenface approach thus provides a practical solution that is well fitted to the problem of face recognition. It is fast, relatively simple and has been shown to work well in constrained environment.

After all the process is carried out then we get the mood of the input image. After that the music player plays songs based on the test image mood which is identified by the system. User can stop the music play by clicking on the stop button in the window which will open after music starts.

Accuracy:

Facial Expression	Recognition Rate(in %)
HAPPY	90.40
SAD	90.60
DISGUIST	88.60
ANGRY	89.60
NEUTRAL	**81.8**

The overall accuracy of the system is 88.16 %.

VII. LIMITATIONS

There are few limitations of this system they are stated below:

- Different faces will influence the recognition rate
- Size of the test image and trained image should be same, different size will decrease the recognition rate.
- Real-time mood detection cannot be processed.

ACKNOWLEDGMENT

I would like to thank Prof. Dr. Anish Chand Turlapaty and T.A. Sai Charan Addanki who gave me such a great opportunity to work on this project. I am also thankful to my colleagues, Wasim Ishaq Khan, Vishal Prasad, Rahul Kumar who supported me on each and every moment to out pass all the hurdles in the implementation and execution of this project.

REFERENCES

[1] Shemi P M, Ali M A, A Principal Component Analysis Method for Recognition of Human Faces: Eigenfaces Approach, International Journal of Electronics Communication and Computer Technology (IJECCT),Volume 2 Issue 3 (May 2012).

[2] M. Turk, A. Pentland: Face Recognition using Eigenfaces, Conference on Computer Vision and Pattern Recognition, 3 – 6 June 1991, Maui, HI , USA, pp. 586 – 591.

[3] https://en.wikipedia.org/wiki/Eigenface

[4] https://en.wikipedia.org/wiki/Principal_component_analysis

[5] Parvinder S. Sandhu, Iqbaldeep Kaur, Amit Verma, Samriti Jindal, Inderpreet Kaur, Shilpi Kumari, Face Recognition Using Eigen face Coefficients and Principal Component Analysis, International Journal on Electrical and Electronics Engineering 3:8 2009.