UNIVERSITY OF BURGUNDY

APPLIED MATHS

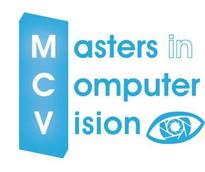
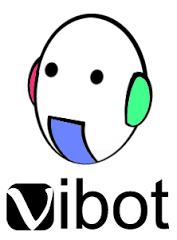
**FACE RECOGNITION ALGORITHM USING PCA**

Submitted by

**Poorna Naga Avinash Narayana**

Supervisor:

Prof. Desired sidibe.



**CONTENTS:**

1. Introduction
2. Methodology
3. Normalization
4. PCA Decomposition of training set
5. Face Recognition
6. GUI
7. Results
8. Conclusions

**FIGURES**

Figure 1: The five facial features used.

Figure 2: Before and after normalization.

Figure 3: GUI : Face recognition for one test image.

Figure 4: GUI : Face recognition for another image with accuracy.

Figure 5: GUI : Face recognition failed for one of the test image.

1. **INRODUCTION:**

Principal component analysis(PCA) is a technique that can be used to simplify the data set. It is a linear transformation that chooses a new coordinate system for the data set such that the greatest variance by any projection of the data set comes to lie on the first axis (called the first principal component), the second greatest variance on the second axis, and so on. PCA can be used for reducing dimensionality in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. The idea is that such low-order components often contain the "most important" aspects of the data.

The main objective of our face recognition algorithm is to extract a set of interesting and discriminative features from the face images with the goal of reducing number of variables.

For this algorithm we used our faces which are high dimension size of 320 x 320 pixels.we used following facial features:1.left eye centre(P1),2.right eye centre (P2),3.tip of nose (P3), 4.left mouth corner (P4), and 5.right mouse corner (P5) as shown in the figure1

P1 P2

P3

P4 P5

pp

Figure 1: The five facial features used.

1. **METHODOLOGY:**

In this facial recognition algorithm initially we took Features of all the images and stored in a excel sheet which is easy to access. For recognition we followed two steps

Firstly we divided the images into two groups a training set and a test set. For training set we used 3 different images for each subject: one frontal view and two side views. Put the other images of each subject in the test set. Secondly with the PCA algorithm we find the eigen values eigen vectors of the faces. The goal of this step is to when we give a face from the test set it will give you the best match from the training set

1. **NORMALIZATION**

After considering Features or Points from all the images we need to do normalization the prior to recognition. The goal of normalization is to account for scale, orientation and location variations. We will use affine transformations here to map the certain features from a face image. Initially we have predetermined values in 64\*64 window as left eye P1(13,20),right eye P2(50,20),Tip of nose P3(34,34),Left mouth corner P4(16,50),Right mouth corner P5(48,50).The affine transformation can be defined by six parameters. We must find the affine parameters for the first image in the data set with respect to predetermined values. The result of this step is to bring all the images into fixed predetermined window of size 64\*64 pixels.

The working of algorithm as follows:

Step 1: Use a vector to store the average locations of each facial feature over all face images i.e is equal to predermined values.

Step 2: Compute the affine parameter that aligns

A =

b=

Step 3: A feature Fi= is mapped to corresponding predetermined location Fip by the equation

Fip = Afi+b

Since we have 10 equations (two for each feature i.e. x and y) we have an over determined system which can be solved by SVD. After calculating the transformation we compute .aligned features We update by setting

Step 4: For every face, Use SVD to compute the affine transformation that aligns with the facial features called the aligned features Fip

Step 5: Update by averaging the aligned feature locations Fip for each face image Fi

Step 6: If the error between Favg and Fp ( according to my program) is less than the threshold, then stop, otherwise, go to step 2.

For my code the algorithm converged at 4 iterations. For each face image, it yields an affine transformation that maps the face to the 64\*64 window the final F is show below

*F =*





Figure 2: Before and after normalization

**4.PCA DECOMPOSITION FOR TRAINING SET**

Let the face image F(x,y) considered as a vector of dimension M \* N. I used a set of 93 images. So that the typical image of size 93\*4096 becomes the vector dimension.

Step 1: Creating a data matrix.

The system will insert all the reshaped images into a new matrix (data matrix) as columns. For example in this case N images of m x n dimension the data matrix will have N columns and m\*n rows.

Step 2: Calculating the mean matrix.

The mean matrix will be calculated by adding all the columns of data matrix divided by the total number of columns.

Step 3: Subtract mean from each column.

Then we subtract the mean Face (image) from all the image matrices to get the mean subtracted data matrix.

Step 4: Calculating the Covariance Matrix.

The system calculates the covariance matrix by multiplying the mean subtracted matrix by its transpose to make it a square matrix in the next phase.

Σ = DDT

Step 5 : Calculating Eigen values and Eigen vectors.

The system calculates all the Eigen values and Eigen vectors of all the training set.

Step 6 : calculating the Eigen vectors of sigma by DT(eigen vector of sigma according to program)

Step 7 : Creating Feature Matrix

In this step we will project all the data on the principal components (since we have 93 pcs, we'll have 93 dimensional feature vectors for each of our data(images)).we will get all the feature vector Matrix which are used in the identification of the image. After this step the

system can recognize any face image by comparing it with the main feature matrix.

**5.FACE RECOGNITION**

The face recognition is identified as the person with the smallest value among all the Euclidean values between the test set and images present in the training set. Euclidean distance provides similarity measure of feature between test set images and images present in the training set. Less Euclidean distance shows that maximum feature of input image and image to be recognized is matched. By comparing the label matrix of both the training set and test set we are getting face recognition

We are calculating the correct recognition accuracy by

Accuracy =\*100

**6.GUI (GRAPHICAL USER INTERFACE)**

We have implemented the GUI for the above methods to see the face recognition for different test images and accuracy calculated for all the test images compared with all the train images.

**RESULTS:**

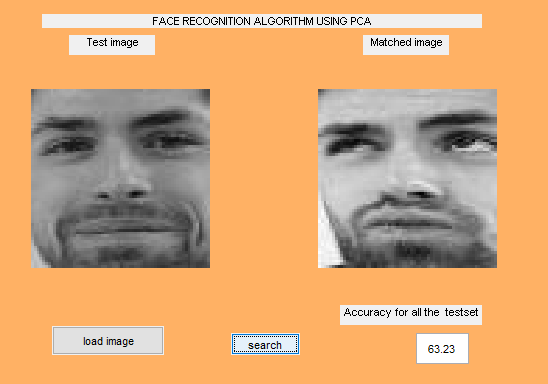
****

Figure 3: Face recognition for one test image

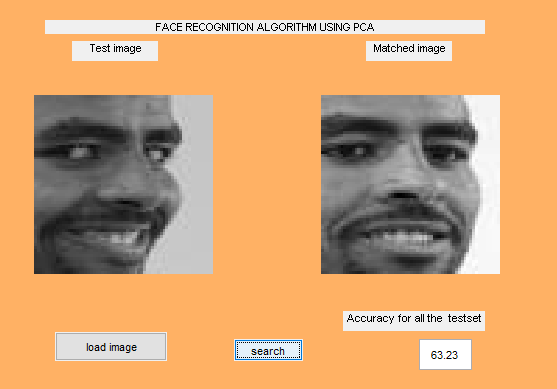
****

Figure 4: Face recognition for another image with accuracy

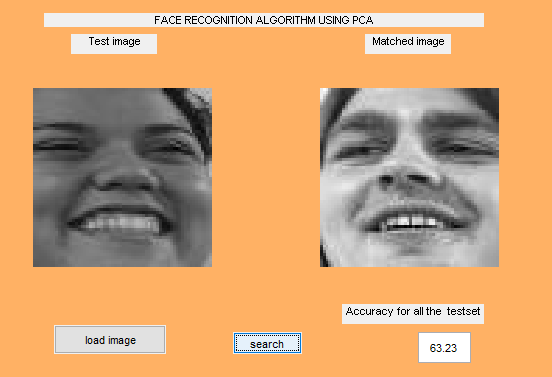
****

Figure 4: Face recognition failed in this case

**CONCLUSIONS**

We have presented a face recognition system based on PCA. We had systematically varied the components and infer the impact of these variations on performance.The PCA plays a immense role in this system dimensionality reduction.