Online Voting System

Sunny Joh[1], Rajnikanth[2]
[1]*Student, BMS Institute of Technology*
[2]*Assistant Professor, BMS Institute of Technology*

***Abstract*- In online voting system people can cast their vote through the internet. In order to prevent voter frauds we use two levels of security. A user id and password are used as the first level of security. The data entered by the user is verified with the contents of the database, if the data is correct then the face of the voter is captured by a web camera and sent to the database. The web page is designed using ASP.NET .The ASP page is then connected to the Microsoft sql sever database. The ASP page is served from an IIS server. In the second level of security the face of the person is verified with the face present in the database and validated using matlab. The comparison of the two faces is done using Eigen face recognition algorithm. The scheme is based on an information theory approach that decomposes face images into a small set of characteristic feature images called 'eigenfaces', which are actually the principal components of the initial training set of face images. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces ('face space') and then classifying the face by comparing its position in the face space with the positions of the known individuals**

I. INTRODUCTION

Developing the application:
Microsoft Active Server Pages (ASP) is a server-side scripting technology. ASP is a technology that Microsoft created to ease the development of interactive Web applications. With ASP you can use client-side scripts as well as server-side scripts. Maybe you want to validate user input or access a database. ASP provides solutions for transaction processing and managing session state. ASP is one of the most successful language used in web development.

II. FACE RECOGNITION

A. The Problem of Face Recognition
Face recognition is a very interesting quandary. Ideally a face detection system should be able to take a new face and return a name identifying that person. Mathematically, what possible approach would be robust and fairly computationally economical? If we have a database of people, every face has special features that define that person. Greg may have a wider forehead, while Jeff has a scar on his right eyebrow from a rugby match as a young tuck. One technique may be to go through every person in the database and characterize it by these small features. Another possible approach would be to take the face image as a whole identity.
Statistically, faces can also be very similar. Walking through a crowd without glasses, blurry vision can often result in misidentifying someone, thus yielding an awkward encounter. The statistical similarity between faces gives way to an identification approach that uses the full face. Using standard image sizes and the same initial conditions, a system can be built that looks at the statistical relationship of individual pixels. One person may have a greater distance between his or her eyes then another, so two regions of pixels will be correlated to one another differently for image sets of these two people.

From a signal processing perspective the face recognition problem essentially boils down to the identification of an individual based on an array of pixel intensities. Using only these input values and whatever information can be gleaned from other images of known individuals the face recognition problem seeks to assign a name to an unknown set of pixel intensities.

B. Obtaining the Eigenface Basis:
The eigenface face recognition system can be divided into two main segments: creation of the eigenface basis and recognition, or detection, of a new face. The system follows the following general flow.

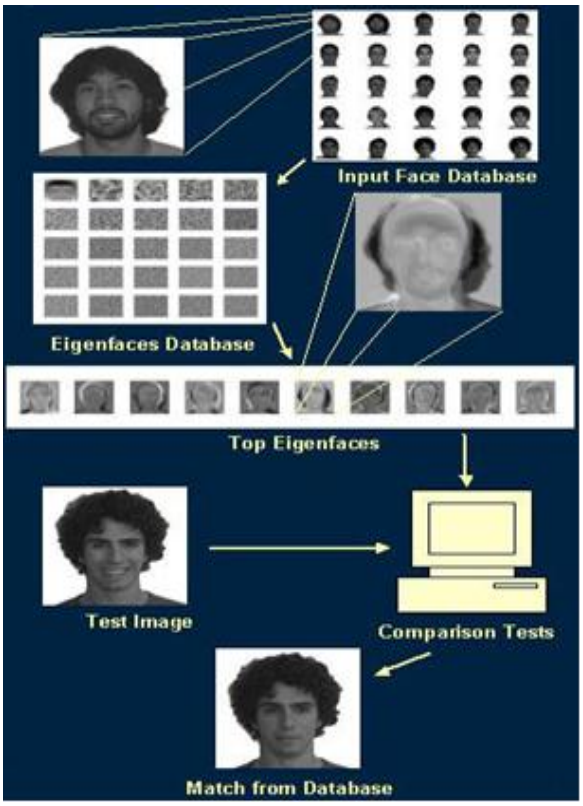


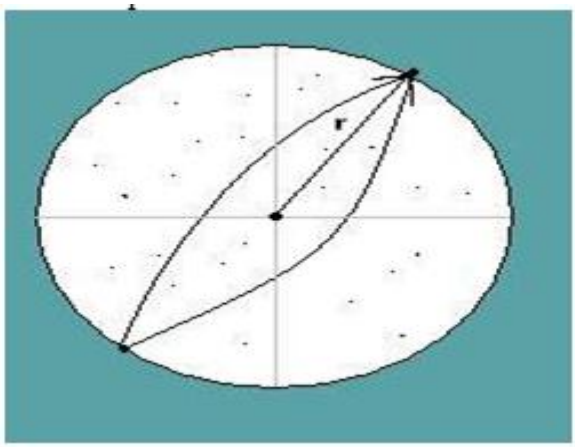
Figure 2: The process of face recognition.

C. Deriving the Eigenface Basis

The eigenface technique is a powerful yet simple solution to the face recognition dilemma. In fact, it is really the most intuitive way to classify a face. As we have shown, old techniques focused on particular features of the face. The eigenface technique uses much more information by classifying faces based on general facial patterns. These patterns include, but are not limited to, the specific features of the face. By using more information, eigenface analysis is naturally more effective than feature-based face recognition. Eigenfaces are fundamentally nothing more than basis vectors for real faces. This can be related directly to one of the most fundamental concepts in electrical engineering: Fourier analysis. Fourier analysis reveals that a sum of weighted sinusoids at differing frequencies can recompose a signal perfectly! In the same way, a sum of weighted eigenfaces can seamlessly reconstruct a specific person's face. Determining what these eigenfaces are is the crux of this technique. Before finding the eigenfaces, we first need to collect a set of face images. These face images become our database of known faces. We will later determine whether or not an unknown face matches any of these known faces. All face images must be the same size (in pixels), and for our purposes, they must be gray scale, with values ranging from 0 to 255. Each face image is converted into a vector Γn of length N (N=image width*image height). The most useful face sets have multiple images per person. This sharply increases accuracy, due to the increased information available on each known individual. We will call our collection of faces "face space." This space is of dimension N.

III. THRESHOLDS FOR EIGENFACE RECOGNITION

For a real system, where the pictures are of standard format like a driver's license photo, the first two cases are useful. In general, the case where one tries to identify a random picture, such a slice of pizza, with a set of faces images is pretty unrealistic. Nonetheless, one can still define these threshold values to characterize the images. Looking back at the weight matrix of values using M eigenfaces, let's define the face space as an M-dimensional sphere encompassing all weight vectors in the entire database. A fairly approximate radius of this face space will then be half the diameter of this sphere, or mathematically, half the distance between the furthest points in the sphere.


Fif.2.M-dimensional sphere

To judge whether a new image falls within this radius, let's calculate the reconstruction error between the image and its reconstruction using M eigenfaces. If the image projects fairly well onto the face space (image follows a face distribution), then the error will be small. However a non-face image will almost always lie outside the radius of the face space.

IV. INTERFACING THE ASP.NET PAGE WITH MATLAB

The matlab program usually has the extension .m. In order to display the output of the matlab program in asp.net the file is converted into .dll by using a deploy tool. The deploy tool converts the matlab file into a dll file. Once converted to dll the file can be deployed in .net environment. The steps to interface: i) Building the .net component, ii) Using the component in an asp.net application. First we create a .NET component that encapsulates a MATLAB function, facerec. The component compares the faces and shows the result. The component is then used in the application called voting.cs

Building the .NET Component: The source code for an application that uses the .NET component is written. Although .NET Builder generates C# code for the facerec component and the sample application is in C#, applications that use the component do not need to be coded in C#. You can access the component from any CLS-compliant .NET language. Build the application using Visual Studio .NET.

V. TESTING

Information Processing has undergone major improvements in the past two decades in both hardware and software. Hardware has decreased in size and price, while providing more and faster processing power. Software has become easier to use, while providing increased capabilities. There is an abundance of products available to assist both end-users and software developers in their work. Software testing, however, has not progressed significantly. It is still largely a manual process conducted as an art rather than a methodology. It is almost an accepted practice to release software that contains defects. Software that is not thoroughly tested is released for production. This is true for both off-the-shelf software products and custom applications. Software vendor and in-house systems developers release an initial system and then deliver fixes to the code. They continue delivering fixes until they create a new system and stop supporting the old one. The user is then forced to convert to the new system, which again will require fixes. Testing is generally associated with the execution of programs. The emphasis is on the outcome of the testing, rather than what is tested and how it's tested. Testing is not a one-step activity; execute the test. It requires planning and design. The tests should be reviewed prior to execution to verify their accuracy and completeness. They must be documented and saved for reuse.

VI. CONCLUSION

At present our government is spending more than 125 crores for conducting a Lok Sabha election. This money is spent on issues such as security, electoral ballots etc. The average percentage of voting is a less than 60% .Moreover voting fraud can be easily done in the present system. Also the percentage of literates coming to vote is very less. But with our system the money spent on election can be reduced to less than 10 crores. Also there is no chance of voter frauds and the money spent on security can be drastically decreased. Persons who have an internet connection at home with a web camera can vote without taking the strain to come to voting booths.

REFERENCES

[1] Matthew A. Turk and Alex P. Pentland. "Eigenfaces for recognisation". Journal of cognitive nerosciences, Volume 3, Number 1, Nov 27, 2002.

[2] Dimitri PISSARENKO. "Eigenface-based facial recognition". Dec1, 2202.

[3] Matthew A. Turk and Alex P. Pentland. "Face recognition using eigenfaces". Proc. CVPR, pp 586-591. IEEE, June 1991.

[4] ASP.NET Bible by Mridula Parihar , Essam Ahmed , Jim Chandler , Bill Hatfield , Rick Lassan , Peter MacIntyre , Dave Wanta
ASP.NET: The Complete Reference by Matthew MacDonald