

Face Recognition-based on Lecture Attendance System

Daljot Kaur Rahsi¹, Zinal Parekh², Shailaja Mohite³

Department of Information Technology
Mumbai University
Mumbai, India

¹prabhsimran2712@gmail.com, ²zinal0parekh@gmail.com
³shailaja.mohite1@rediffmail.com

Abstract - This system is used for automatic attendance by continuous observation, where our approach is to solve the low effectiveness and improve the efficiency of face recognition. This paper first review the related works in the field of attendance management and face recognition. Then, it introduces our system structure and plan. Finally, experiments are implemented to provide as evidence to support our plan. The result shows that continuous observation improved the performance for the estimation of the attendance. This is developed by using Eigen-Face algorithm and by ASD method. We use various techniques like Shooting plan and Face Detection.

General Terms

Attendance, Effectiveness, Attendance Management, Face Recognition.

Keywords

Automatic Attendance, Shooting plan, Face Detection, ASD method, Eigen-Face algorithm, Continuous Observation.

I. INTRODUCTION

Biometric templates can be of many types like Fingerprints, Eye Iris, Face, Hand Geometry, Signature, Gait and voice. Many systems were developed to manage the context of the students for the classroom lecture by using note PCs for all the students. Because this system uses the note PC of each student, the attendance and the position of the students are obtained. However, it is difficult to know the detailed situation of the lecture. Our system takes images of faces.

Our system uses the face recognition approach for the automatic attendance of students in the classroom environment without students' intervention. Student's context such as presence, seat position, status, and comprehension are mentioned. At the same time face images reflect a lot about this context information. It is possible to estimate automatically whether each student is present or absent and where each student is sitting by using face recognition technology. It is also possible to know whether students are awake or sleeping and whether students are interested or bored in lecture if face images are annotated with the students 'name, the time and the place. We are concerned with the method to use face image processing technology.

By continuously observing of face information, our approach can solve low effectiveness of existing face detection technology, and improve the accuracy of face

recognition. We propose a method that takes the attendance using face recognition based on continuous observation. In this paper, our purpose is to obtain the attendance, positions and images of students' face, which are useful information in the classroom lecture.

II. LITERATURE SURVEY

In recent decade, a number of algorithms for face recognition have been proposed, but most of these works deal with only single image of a face at a time. By continuously observing of face information, our approach can solve the problem of the face detection [1], and improve the accuracy of face recognition.

The different techniques used for face detection are Knowledge based method that finds the relationship between facial features, Feature invariant method aims to find structural features of a face, Template matching method describes the several standard patterns, and Appearance based method which captures the facial appearance.

The different techniques used for face recognition are Holistic approach where the whole face is taken as input for recognition purpose using Principal Component Analysis[8] and Linear Discriminate Analysis[9] and Feature based approach where local features on face like nose and eyes are detected[10].

III. PROPOSED SYSTEM

Face recognition attendance system used to mark the attendance details of students[2]. It can mark the time of lecture and daily presence of the individuals in a premise and generate detailed reports on the same at regular intervals.

Technically following is the usefulness of this project:

1. Enhances security and speed in tracing student attendance and lecture time.
2. Easy to set up and use.
3. Convenient and inexpensive.
4. Helps in managing the time and attendance profiles of students.
5. Eliminates proxy punching.
6. Manages student attendance records.
7. Easy to refer to the lecture time attendance record
8. Easily configured according to your requirement.
9. Reduces the manual student's data entry, register maintenance and monthly requirements.

IV. SYSTEM DESCRIPTION

Our system consists of two kinds of cameras. One is the sensing camera on the ceiling to obtain the seats where the

students are sitting. The other is the capturing camera in front of the seats to capture images of student's face (see Figure 1)

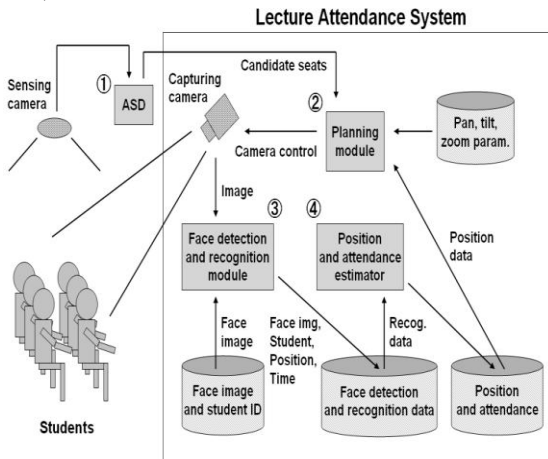


Figure 1: Architecture of the system

The procedure of our system consists the following steps:

A. Active Student Detecting method

We use the method of Active Student Detecting i.e. ASD[3] to estimate the existence of a student sitting on the seat. It is described in detail. In this approach, an observation camera with fisheye lens is installed on the ceiling of the classroom and looks down at the student area vertically. ASD estimates students' existence by using the background subtraction and inter-frame subtraction of the images captured by the sensing camera (see Figure 2). In the background subtraction method, noise factors like bags and coats of the students are also detected, and the students are not detected if the colour of clothes of them are similar the seats. ASD makes use of the inter-frame subtraction to detect the moving of the students.

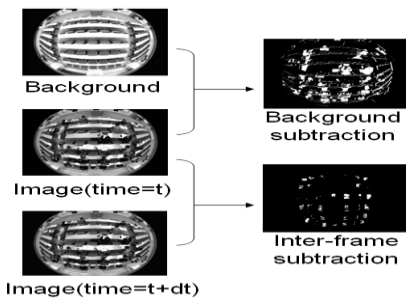


Figure 2: Active Student Detecting method

4.1 Shooting plan

Camera planning module selects one seat from the estimated sitting area in order to determine where to direct the front camera. Actually, in this paper, the module selects a seat by scanning the seats sequentially. This approach is insufficient because it wastes time directing the camera to where the student-and-seat the seats the students correspondence is already decided In other words, if we direct the camera to each seat with the same probability, it is difficult to detect the faces according to the student or the seat, and the system judges the students who are actually present to be absent consequently. In order to solve this problem, it is important to the information of each student's position. The camera is directed to the selected seat using the pan/tilt/zoom that has

been registered in the database. The camera captures the image of the student.

B. Continuous Observation

Our system takes the attendance automatically using face recognition. However, it is difficult to estimate the attendance precisely using each result of face recognition independently because the face detection rate is not sufficiently high. We propose a method for estimating the attendance precisely using all the results of face recognition obtained by continuous observation. Continuous observation improves the performance for the estimation of the attendance we want construct the lecture attendance system based on face recognition, and applied the system to classroom lecture. Then, it introduces our system structure and plan. Finally, experiments are implemented to provide as evidence to support our plan. The result shows that continuous observation improved the performance for the estimation of the attendance. By continuously observing of face information, our approach can solve low effectiveness of existing face detection technology, and improve the accuracy of face recognition. We propose a method that takes the attendance using face recognition based on continuous observation. In this paper, our purpose is to obtain the attendance, positions and images of students' face, which are useful information in the classroom lecture.

C. Estimating seat of each student

In order to solve the problem of ineffectiveness, we integrated students' seat information into the camera planning. In this way, we can solve the problem such as misrecognition of faces and seats by constraints of the correspondence relationship between them. The face detected from the captured image may be another neighbour student's face. Therefore, it is necessary to consider the possibility that the face image is the one of a neighbour student even if the camera is directed to the target seat.

The procedure is repeated during lecture, and estimated the attendance of the students in real time.

V. SYSTEM ALGORITHM

This section describes the software algorithms for the system.

A. We use Eigen-Face algorithm

The reasons for using Eigen-Face algorithm are:

1. This system performs face recognition in real time and also uses this method along with motion cues to segment faces out of images by discarding squares that are classified as non-face images.
2. It is a fast, simple and practical algorithm.

B. The Eigen Classifier

The Eigen recognizer takes two variables. The 1st is the number of components kept for this Principal Component Analysis. There's no rule how many components that should be kept for good reconstruction capabilities. It is based on your input data, so experiment with the number. OpenCV documentation suggests keeping 80 components should almost always be sufficient. The 2nd variable is designed to be a prediction threshold; this variable contains the bug as any value above this is considered as an unknown. For the Fisher and LBPH this is how unknowns are classified however with the Eigen recognizer we must use the return distance to provide our own test for unknowns. In the Eigen recognizer the larger the value returned the closer to a match we have.

C. The Fisher Classifier

The Fisher recognizer takes two variables as with the Eigen constructor. The 1st is the number of components kept Linear Discriminate Analysis with the Fisher faces criterion. It's useful to keep all components; this means the number of your training inputs. If you leave this at the default (0), set it to a value less than 0, or greater than the number of your training inputs, it will be set to the correct number (your training inputs - 1) automatically. The 2nd Variable is the threshold value for unknowns, if the resultant Eigen distance is above this value the Predict () method will return a -1 value indicating an unknown. This method works and the threshold is set to a default of 3500, change this to constrain how accurate you want results. If you change the value in the constructor the recognizer will need retraining.

D. The Local Binary Pattern Histogram (LBPH) Classifier

The LBPH recognizer unlike the other two takes five variables:

- Radius – The radius used for building the Circular Local Binary Pattern.
- Neighbours – The number of sample points to build a Circular Local Binary Pattern from. A value suggested by OpenCV Documentations is '8' sample points. Keep in mind: the more sample points you include, the higher the computational cost.
- grid_x – The number of cells in the horizontal direction, 8 is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.
- grid_y – The number of cells in the vertical direction, 8 is a common value used in publications. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.
- Threshold – The threshold applied in the prediction. If the distance to the nearest neighbour is larger than the threshold, this method returns -1.

Principle Component Analysis (PCA)

- Stage 1: Subtract the Mean of the data from each variable (our adjusted data)
- Stage 2: Calculate and form a covariance Matrix

- Stage 3: Calculate Eigenvectors and Eigen values from the covariance Matrix
- Stage 4: Chose a Feature Vector (a fancy name for a matrix of vectors)
- Stage 5: Multiply the transposed Feature Vectors by the transposed adjusted data
- Stage 1: Mean Subtraction

This data is fairly simple and makes the calculation of our covariance matrix a little simpler now this is not the subtraction of the overall mean from each of our values as for covariance we need at least two dimensions of data. It is in fact the subtraction of the mean of each row from each element in that row.

(Alternatively the mean of each column from each element in the column however this would adjust the way we calculate the covariance matrix)

- Stage 2: Covariance Matrix

The basic Covariance equation for two dimensional data is:

$$cov(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)}$$

Which is similar to the formula for variance however, the change of x is in respect to the change in y rather than solely the change of x in respect to x. In this equation x represents the pixel value and \bar{x} is the mean of all x values, and n the total number of values.

The covariance matrix that is formed of the image data represents how much the dimensions vary from the mean with respect to each other. The definition of a covariance matrix is:

$$C^{n*n} = (C_{i,j}, C_{i,j} = cov(Dim_i, Dim_j))$$

Now the easiest way to explain this is but an example the easiest of which is a 3x3 matrix.

Now with larger matrices this can become more complicated and the use of computational algorithms essential.

$$C_{mat} = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

$$I1 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad C(I1) = \begin{pmatrix} cov(1,1) & cov(2,5) & cov(3,9) \\ cov(4,1) & cov(5,5) & cov(6,9) \\ cov(7,1) & cov(8,5) & cov(9,9) \end{pmatrix}$$

- Stage 3: Eigenvectors and Eigen values

Eigen values are a product of multiplying matrices however they are as special case. Eigen values are found by multiples of the covariance matrix by a vector in two dimensional space (i.e. a Eigenvector). This makes the covariance matrix the equivalent of a transformation matrix. It is easier to show in a example:

Table 1. Image Pixels

Operational requirement	Horizontal pixels/ face	Pixels/ cm	Pixels/inch
Identification	80 px/face	5 px/cm	12.5 px/in
Recognition	16 px/face	1 px/cm	2.5 px/in
Detection	3 px/face	≈0.2 px/cm	0.5 px/in

$$\text{Covariance Matrix} = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix}$$

$$\text{Eigenvector} = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Multipied:

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} * \begin{bmatrix} 6 \\ 4 \end{bmatrix} = \begin{bmatrix} 24 \\ 16 \end{bmatrix} = 4 \begin{bmatrix} 6 \\ 4 \end{bmatrix}$$

Eigenvectors can be scaled so ½ or x2 of the vector will still produce the same type of results. A vector is a direction and all you will be doing is changing the scale not the direction.

$$\begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} * \begin{bmatrix} 3 \\ 2 \end{bmatrix} = \begin{bmatrix} 12 \\ 8 \end{bmatrix} = 4 \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Eigenvectors are usually scaled to have a length of 1:

$$\begin{bmatrix} A \\ B \end{bmatrix} \text{ becomes } \begin{bmatrix} A/(\sqrt{A^2 + B^2}) \\ B/(\sqrt{A^2 + B^2}) \end{bmatrix}$$

Thankfully finding these special Eigenvectors is done for you and will not be explained however there are several tutorials available on the web to explain the computation. The Eigen value is closely related to the Eigenvector used and is the value of which the original vector was scaled in the example the Eigen value is 4.

➤ Stage 4: Feature Vectors

Now a usually the results of Eigen values and Eigenvectors are not as clean as in the example above. In most cases the results provided are scaled to a length of 1. So here are some example values calculated using Matlab:

$$\text{Covariance Matrix} = \begin{bmatrix} 0.212 & 0.345 \\ 0.621 & 0.111 \end{bmatrix}$$

$$\text{Eigenvector} = \begin{bmatrix} 0.6271 \\ 0.3041 \end{bmatrix}$$

$$\text{Eigenvalues} = \begin{bmatrix} 0.6392 & -0.557 \\ 0.7691 & 0.3814 \end{bmatrix}$$

Once Eigenvectors are found from the covariance matrix, the next step is to order them by Eigen value, highest to lowest. This gives you the components in order of significance. Here the data can be compressed and the weaker vectors are removed producing a lossy compression method, the data lost is deemed to be insignificant.

$$\text{Resultant Eigenvalues} = \begin{bmatrix} 0.6392 \\ 0.7691 \end{bmatrix}$$

➤ Stage 5: Transposition

The final stage in PCA is to take the transpose of the feature vector matrix and multiply it on the left of the transposed adjusted data set (the adjusted data set is from Stage 1 where the mean was subtracted from the data).

E. Image Pixel:

F. Eigen-Face algorithm and the formulas used:

1. The first step is to obtain a set S with M face images. In our example M = 25 as shown at the beginning of the tutorial. Each image is transformed into a vector of size N and placed into the set.

$$S = \{ \Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M \}$$

2. After you have obtained your set, you will obtain the mean image Ψ

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

3. Then you will find the difference Φ between the input image and the mean image

$$\Phi_i = \Gamma_i - \Psi$$

4. Next we seek a set of M orthonormal vectors, u_n, which best describes the distribution of the data. The kth vector, u_k, is chosen such that

$$\lambda_{k} = \frac{1}{M} \sum_{n=1}^M (u_{k}^T \Phi_n)^2$$

5. is a maximum, subject to

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases}$$

Note: u_k and λ_k are the eigenvectors and Eigen-Values of the covariance matrix C.

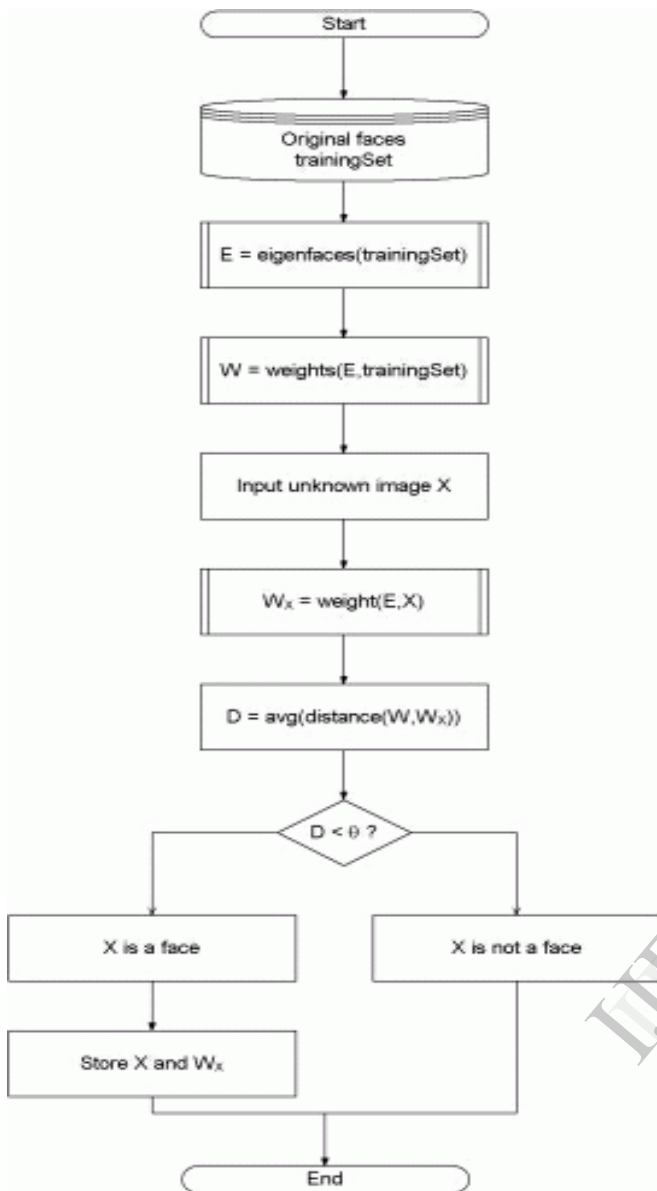


Figure 3: Software algorithm- Flow diagram of Eigen-Face algorithm.

VI. EXPERIMENT

A. Result of Estimating the seat of each student

19 students existed in the centre area, and we ran the process of camera control and detection for 20 minutes. We labelled the images of the detected faces with the name of the students manually. The system detected faces 186 times, and 15 students were detected. Table 1 shows the accuracy of seat estimation.

B. Result of Estimating the attendance based on continuous observation

We compared the results one cycle only and continuous observation. 12 students existed in the centre area, and 2 of them did not have their faces registered. In this experiment of 79 minutes, 8 scanning cycles were completed during this period. Table 2 shows face detection rate, and Table 3 shows the result of estimating the attendance. In the case of 1 cycle only, we judge the recognized students to be present. In the case of continuous observation, the system estimates

the attendance by the method of section 3.5 using the recognition data obtained during 79 minutes. This table shows that continuous observation improved the face detection rate and improved F-score of estimation of the attendance, which is the harmonic mean of precision and recall.

Table 2. Result of estimating the seat of each student

Iteration	Accuracy
Iteration 1	60.0% (9/15)
Iteration 2	73.3% (11/15)
Iteration 3	80.0% (12/15)

Table 3. Face detection rate

Time	face detection rate
1 cycle only	37.5% (3.8/10)
79 min	80.0% (8/10)

Table 4. Result of estimating the attendance

Time	Precision	Recall	F-score
1 cycle only	89.2%	33.8%	48.3%
79 min	70.0%	70.0%	70.0%

VII. OUTPUT

1. Face detected from webcam

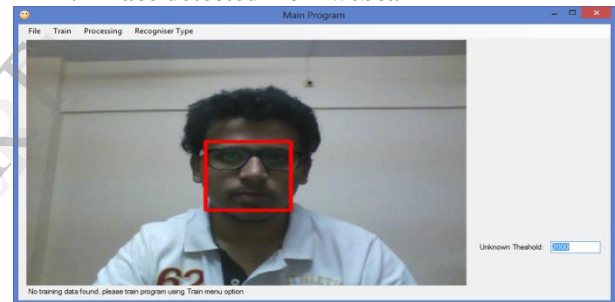


Figure 4: Output 1

2. Saved information about face

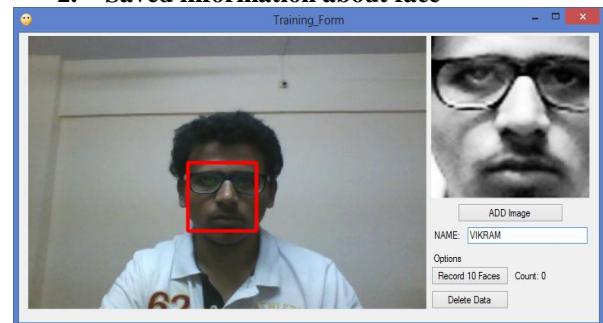


Figure 5: Output 2

3. Face Recognized

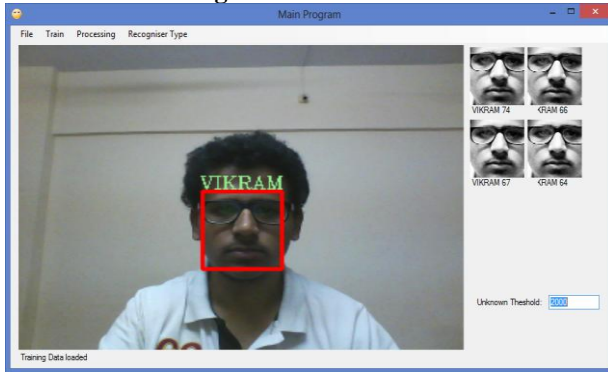


Figure 6: Output 3

4. We have used 3 algorithms we can switch between them

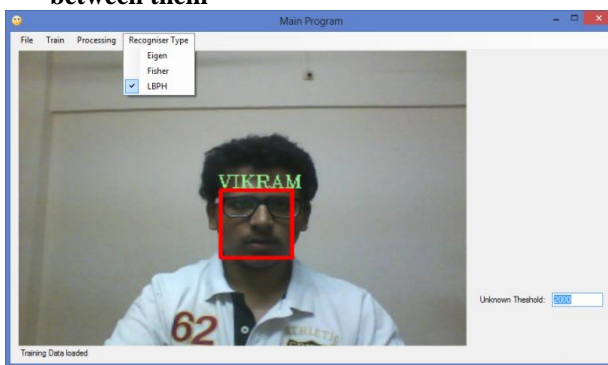


Figure 7: Output 4

VIII. CONCLUSION

This paper introduces the efficient and accurate method of attendance in the classroom environment that can replace the old manual methods. This method is secure enough, reliable and available for use. No need for specialized hardware for installing the system in the classroom. It can be constructed using a camera and computer.

IX. FUTURE SCOPE

In further work, we intend to improve face detection effectiveness by using the interaction among our system, the students and the teacher. On the other hand, our system can be improved by integrating video-streaming service and lecture archiving system, to provide more profound applications in the field of distance education, course management system (CMS) and support for faculty development (FD). We can further improve this system so as we can run this system with more than two students on a bench and allowing them to change their positions.

REFERENCES

- [1] http://faculty.ucmerced.edu/mhyang/papers/icpr04_tutorial.pdf
- [2] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, 2003, vol. 35, no. 4, pp. 399-458.
- [3] S. Nishiguchi, K. Higashi, Y. Kameda and M. Minoh, "A Sensor-fusion Method of Detecting A Speaking Student," *IEEE International Conference on Multimedia and Expo (ICME2003)*, 2003, vol. 2, pp. 677- 680.
- [4] *Handbook of Face Recognition* by- Stan Z. Li, Anil K. Jain
- [5] *Thermal Infrared Face Recognition – A Biometric Identification Technique for Robust Security system* by- Mrinal Kanti

- Bhowmik, Kankan Saha, Sharmistha Majumder, Goutam Majumder, Ashim Saha, Aniruddha Nath Sarma, Debotosh Bhattacharjee, Dipak Kumar Basu and Mita Nasipuri
- [6] *System Design, Implementation and Evaluation* by- Harry Wechsler
 - [7] *Face Recognition* by- Dilloo Lubna
 - [8] M. Turk, A. Pentland, "Eigenfaces for Recognition", *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, 1991, pp. 71-86
 - [9] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, "Face Recognition Using LDA-Based Algorithms", *IEEE Trans. On Neural Networks*, Vol. 14, No. 1, January 2003, pp. 195-200
 - [10] Ramesha, K and Raja "Feature extraction based facerecognition, gender and age classification" ,(IJCSE) *International Journal on Computer Science and Engineering*, Vol. 02, No.01S, 2010, pp 14-23.