E-ATTENDANCE SYSTEM BASED ON FACE RECOGNITION

Bipasha Biswas

Dept. of Computer Applications, Haldia Institute of Technology, Haldia, Purba Medinipur,

bipasm@gmail.com.

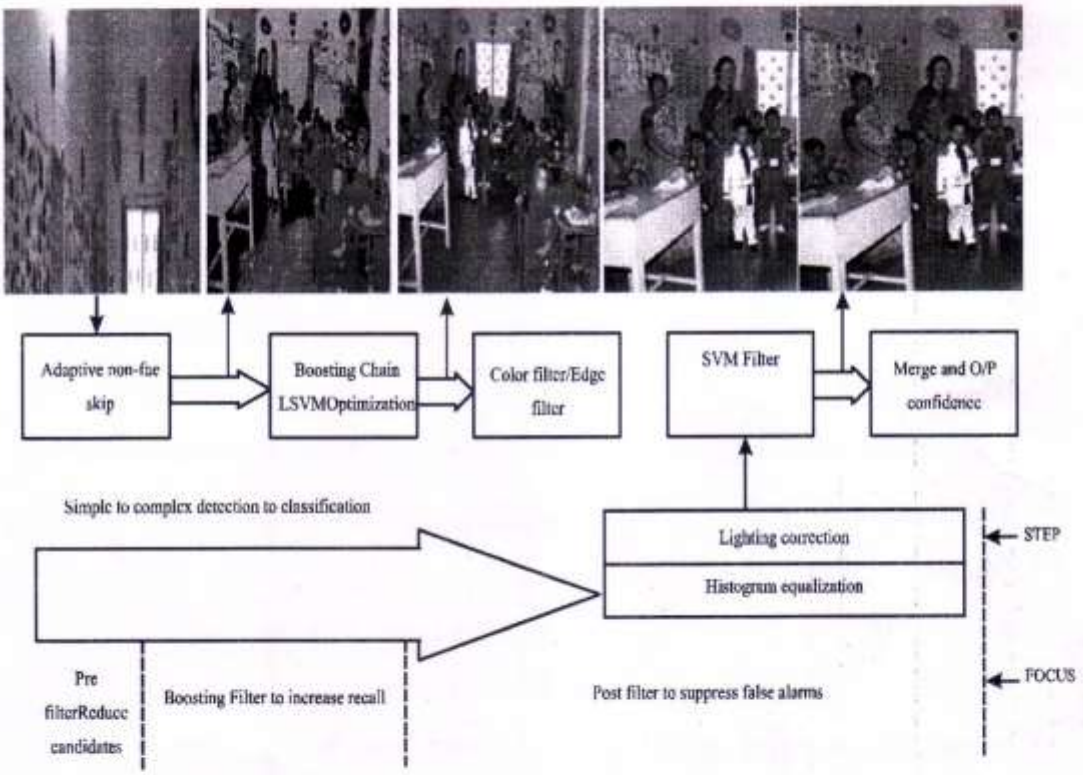
Abstract:-The objective of this paper is to develop the E-Attendance system and recognize the faces using face recognition technique. MATLAB based programs are implemented to identify the faces using Indian databases and the Face recognition data. Face recognition is one of the most relevant applications of image processing. It is a true challenge to build an automated system which is equal to human ability to recognize faces. This paper is emphasized on face recognition system to record the attendance of valid students in any classroom. The system takes photographic images or a video stream as input. The output consists of an array of rectangles which corresponds to the locations and scales of faces detected. If it detects no face an empty array is returned.

Keywords: *Face recognition, Image processing, MATLAB, DCT, Attendance*

Introduction

Many researchers tried to understand how human recognizes faces to design and implement automatic face recognition technique.

Face recognition system structure: In the first step the system uses an adaptive prefilter to eliminate candidate rectangle in the input that it can confidently determine do not contain faces. In the second step the system feeds the remaining rectangles to an improved implementation of an algorithm called "Boosting chain". A general classification framework, called boosting chain, is proposed for learning boosting cascade. In this framework, a "chain" structure is introduced to integrate historical knowledge into successive boosting learning. Moreover, a linear optimization scheme is proposed to address the problems of redundancy in boosting learning and threshold adjusting in cascade coupling. By this means, the resulting classifier consists of fewer weak classifiers yet achieves lower error rates than boosting cascade in both training and test. Experimental comparisons of boosting chain and boosting cascade are provided through a face detection problem. The promising results clearly demonstrate the effectiveness made by boosting chain.



Pic1: paper detailing boosting chain

Feature Extraction: In the face recognition, the feature representation of a face is a critical aspect. If representation step does not perform well, even the best classifiers cannot produce appropriate results. Good representations are those that on one hand minimize intra-person dissimilarities, on the other hand maximize differences between persons. Additionally, a significant representation should be fast and compact .There are several views related to the classifications of the feature extraction methods. One very important area of application is image processing, in which algorithms are used to detect and isolate various desired portions or shapes (features) of a digitized image or video stream. The feature extraction method is involved in several steps-dimension reduction, feature extraction and selection. **Dimension reduction** is the process of reducing the number of random variables under consideration and can be divided into feature selection and feature extraction methods. **Feature selection** approaches try to find a subset of the original variables (also called features or attributes). There are three strategies such as, filter(e.g. information gain), wrapper(e.g. search guided by accuracy) approaches, and embedded(features are selected to add or be removed while building the model based on the prediction errors). In some cases, data analysis such as regression or classification can be done in the reduced space more accurately than in the original space. **Feature extraction** transforms the data in the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal

component analysis (PCA), but many non-linear dimensionality reduction techniques also exist. For multidimensional data, tensor representation can be used in dimensionality reduction through multilinear subspace learning. There are many feature extraction algorithms:

Method	Notes
Principal component analysis	Eigenvector based linear
Kernel PCA	Eigenvector based non linear
Weighted PCA	PCA using Weighted coefficient
Linear Discriminate Analysis	Eigenvector based, supervised linear map
SDA(Semi-supervised Discriminate Analysis)	Semi-supervised adaptation of LDA
Kernel LDA	LDA-based, uses kernel method
ICA	Linear map
Neural Network based method	Diverse linear network using PCA
Multi Dimensional Scaling	Non-linear map,noise
Self-organizing map	Non-linear based on a grid of neurons in the feature space
ASM	Statistical method, searching boundaries
AAM	Evolution of ASM
Gavor wavelet transforms	Biological motivated, linear filter
DTC	Linear function, usually used in 2D-DCT

Some of the algorithms of Feature Selection methods are like:

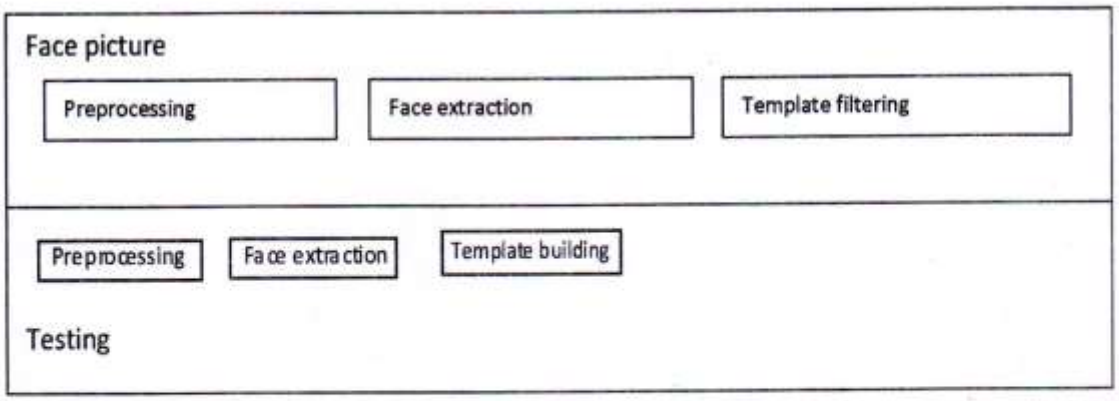
Method	Definition	Comments
Exhaustive search	Evaluate all possible subsets of features	Optimal but complex
Branch and Bound	Use branch and bound algorithm	Can be optimal
Best individual features	Evaluate and select features	Not very effective

Sequential forward Selection(SFS)	Evaluate growing Feature sets	Retained features Can not discared
Sequential Backward selection	Evaluate shrinking feature set	Deleted features cannot be evaluated
Sequential forward and backward floating search	Like automatic pick and dynamic update	Close to optimal
Selection	First to SFS than SBS	Must choose the value of vector

Different approaches of face recognition:

There are so many approaches of face recognition.

1. Geometric/Template based approach
2. Piecemeal approaches
3. Appearances-based/model based approaches
4. Template/statistical network approaches
   - 4.1. Template matching
   - 4.2. Statistical approach
   - 4.3. Neural networks



Template Design

- Manually selected a number of quality faces: as centered, straight, lighting, diverse.
- Measured face dimension and used Matlab to uniformly scale and align them.
- Efforts resulted in sample faces added to produce final template.





Discrete Cosine Transformation Algorithm:

A discrete cosine transform (DCT), expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies in image compression. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out (as described below) that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real

numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. There are eight standard DCT variants, of which four are common.

The most common variant of discrete cosine transform is the type-II DCT, which is often called simply "the DCT", its inverse, the type-III DCT, is correspondingly often called simply "the inverse DCT" or "the IDCT". Two related transforms are the discrete sine transform (DST), which is equivalent to a DFT of real and *odd* functions, and the modified discrete cosine transform (MDCT), which is based on a DCT of *overlapping* data.

The DCT is used in JPEG image compression, MJPEG, MPEG, DV, Daala, and Theora video compression. There, the two-dimensional DCT-II of NXN blocks are computed and the results are quantized and entropy coded. In this case, N is typically 8 and the DCT-II formula is applied to each row and column of the block. The result is an 8 × 8 transform coefficient array in which the (0,0) element (top-left) is the DC (zero-frequency) component and entries with increasing vertical and horizontal index values represent higher vertical and horizontal spatial frequencies. Let, N real numbers x_0, x_1,.....,x_{n-1} are transformed. Formally, the Discrete Cosine Transform (DCT) is a linear, invertible function $f:R^N->R^N$ (where R denotes the set of real numbers), or equivalently an invertible NXN square matrix. There are several variants of DCT with slightly modified definition. The N real numbers $x_0, \ldots, x_{N-1}$ are transformed into the N real numbers $X_0, \ldots, X_{N-1}$ according to one of the formulas:

DCT-1

$$x^k = \frac{1}{2}(x_0 + (-1)^k x_{n-1}) + \sum_{n=1}^{N-2} x_n \left[\cos \frac{\pi}{N-1} nk\right]$$

k=0,.....N-1

Some authors multiply x_0 and x_{N-1} terms by $\sqrt{2}$ and correspondingly multiply the x_0 and x_{N-1} terms by $\frac{1}{\sqrt{2}}$. This makes the DCT-1 matrix orthogonal. The DCT-I is exactly equivalent (up to an overall scale factor of 2), to a DFT of 2N-2 real numbers with even symmetry. For example, a DCT-I of N=5 real numbers *abcde* is exactly equivalent to a DFT of eight real numbers *abcdedcb* (even symmetry), divided by two. (In contrast, DCT types II-IV involve a half-sample shift in the equivalent DFT.)

Note, however, that the DCT-I is not defined for N less than 2. (All other DCT types are defined for any positive N). Thus, the DCT-I corresponds to the boundary conditions: x_n is even around n=0 and even around n=N-1; similarly for X_k.

DCT-II

$$x^k = \sum_{n=0}^{N-1} x_n \left[\cos \frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

k=0,...N-1

The DCT-II is probably the most commonly used form, and is often simply referred to as "the DCT". This transform is exactly equivalent (up to an overall scale factor of

2) to a DFT of 4N real inputs of even symmetry where the even-indexed elements are zero. That is, it is half of the DFT of the 4N inputs y_n, where, $y_{2n}=x_n$ for, 0d"nd"N, $y_{2n}=0$ and y_{4N-n} for 0d"nd"2N. Some authors further multiply the X_0 term by 1/"2 and multiply the resulting matrix by an overall scale factor of $\sqrt{2/N}$. This makes the DCT-II matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted input. This is the normalization used by Matlab, for example. In many applications, such as JPEG, the scaling is arbitrary because scale factors can be combined with a subsequent computational step (e.g. the quantization step in JPEG), and a scaling that can be chosen that allows the DCT to be computed with fewer multiplications. The DCT-II implies the boundary conditions: x_n is even around n=-1/2 and even around n=N-1/2; X_k is even around k=0 and odd around k=N.

DCT-III

$$x^k = \frac{1}{2}x0 + \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{n}n\left(k+\frac{1}{2}\right)\right]$$

k=0,...N-1

Because it is the inverse of DCT-II (up to a scale factor, see below), this form is sometimes simply referred to as "the inverse DCT" ("IDCT"). Some authors divide the x_0 term by "2 instead of by 2 (resulting in an overall x_0/"2 term) and multiply the resulting matrix by an overall scale factor of $\frac{\sqrt{2}}{N}$ so that the DCT-II and DCT-III are transposes of one another. This makes the DCT-III matrix orthogonal, but breaks the direct correspondence with a real-even DFT of half-shifted output. The DCT-III implies the boundary conditions: x_n is even around n=0 and odd around n=N; X_k is even around k=-1/2 and odd around k=N-1/2.

DCT-IV $\sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)\left(k+\frac{1}{2}\right)\right]$

k=0,..N-1

The DCT-IV matrix becomes orthogonal (and thus, being clearly symmetric, its own inverse) if one further multiplies by an overall scale factor of $\sqrt{\frac{2}{N}}$. A variant of the DCT-IV, where data from different transforms are *overlapped*, is called the modified discrete cosine transform (MDCT) (Malvar, 1992). The DCT-IV implies the boundary conditions: x_n is even around n=-1/2 and odd around n=N-1/2; similarly for X_k.

DCT V-VIII

DCTs of types I-IV treat both boundaries consistently regarding the point of symmetry: they are even/odd around either a data point for both boundaries or halfway between two data points for both boundaries. By contrast, DCTs of types V-VIII imply boundaries that are even/odd around a data point for one boundary and halfway between two data points for the other boundary. In other words, DCT types I-IV are equivalent to real-even DFTs of even order (regardless of whether N is even or odd), since the corresponding DFT is of length 2(N"1) (for DCT-I) or 4N (for DCT-II/III) or 8N (for DCT-IV). The four additional types of dis-

crete cosine transform (Martucci, 1994) correspond essentially to real-even DFTs of logically odd order, which have factors of $N\pm½$ in the denominators of the cosine arguments.

However, these variants seem to be rarely used in practice. One reason, perhaps, is that FFT algorithms for odd-length DFTs are generally more complicated than FFT algorithms for even-length DFTs (e.g. the simplest radix-2 algorithms are only for even lengths), and this increased intricacy carries over to the DCTs as described below.

(The trivial real-even array, a length-one DFT (odd length) of a single number a, corresponds to a DCT-V of length N=1.)

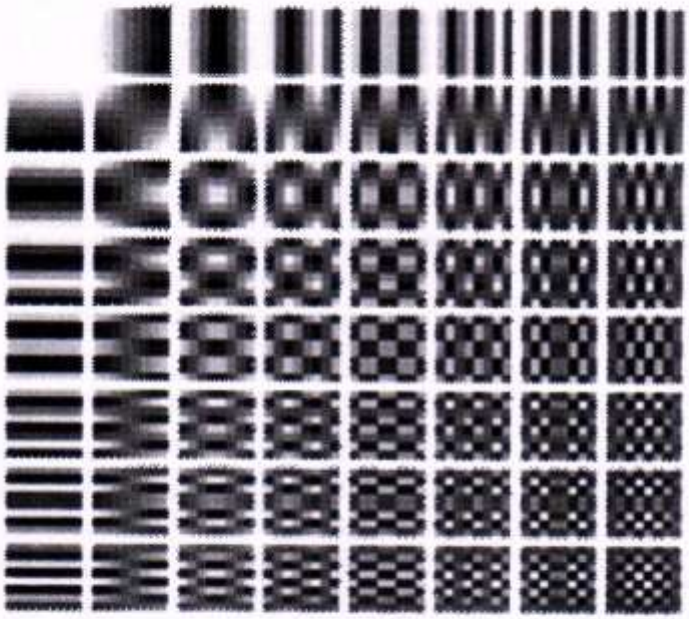
Inverse Transforms Using the normalization conventions above, the inverse of DCT-I is DCT-I multiplied by $2/(N-1)$. The inverse of DCT-IV is DCT-IV multiplied by $2/N$. The inverse of DCT-II is DCT-III multiplied by $2/N$ and vice versa.

Like for the DFT, the normalization factor in front of these transform definitions is merely a convention and differs between treatments. For example, some authors multiply the transforms by $\sqrt{\frac{2}{N}}$ so that the inverse does not require any additional multiplicative factor. Combined with appropriate factors of "2 (see above), this can be used to make the transform matrix orthogonal.

Multidimensional DCTs Multidimensional variants of the various DCT types follow straightforwardly from the one-dimensional definitions: they are simply a separable product (equivalently, a composition) of DCTs along each dimension. For example, a two-dimensional DCT-II of an image or a matrix is simply the one-dimensional DCT-II, from above, performed along the rows and then along the columns (or vice versa). That is, the 2D DCT-II is given by the formula (omitting normalization and other scale factors, as above):

$$x_{k1,k2} = \sum_{n1=0}^{N_1-1}\left(\sum_{n2=0}^{N_2-1} x_{n1,n2}\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]\cos\frac{\pi}{N_1}\left(n1+\frac{1}{2}\right)k_1\right]$$

$$= \sum_{n1=0}^{N_1-1}\sum_{n2=0}^{N_2-1} x_{n1,n2}\cos\left[\frac{\pi}{N_1}\left(n1+\frac{1}{2}\right)k_1\right]\cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]$$



2D DCT frequencies from JPEG DCT

MATLAB based face recognition security system:

MATLAB is an interactive working environment in which the user can carry quite complex computational tasks with few commands. It was originally developed by Cleve Muller in 1970. The initial programming was in FORTRAN and then after various evolutions the latest version is written in C. The results of calculations are viewed in numerically as well as in 2D and 3D graph easily and quickly. The quite complex computations can be done in just a few commands because the details of programming are stored in a script file which is called m-file s and they can be invoked by their names. One m-file can invoke another m-file and by the way series of m-files running behind a scene can allow executing required tasks easily.

Digital image processing is an area characterised by the need for extensive experimental work to establish the viability of proposed solutions to a given problem. An image may be defined as a two dimensional function f(x,y) where x and y are spatial coordinates and the amplitude od f at any pair of coordinates (x,y) is called the intensity or gray level of image at the point. When x,y and the amplitude values of f are all finite , discrete quantities, we call the image a digital image. Digital image processing means the processing of digital image by computer.

A digital image can be represented naturally as MATLAB matrix:

$$f = \begin{matrix} f(1,1) & f(1,2) & f(1,N) \\ f(2,1) & f(2,2) & f(2,N) \\ f(M,1) & f(M,2) & f(M,N) \end{matrix}$$

Where f(1,1)=f(0,0). Clearly, the two representations are identical, except for the shift in origin. The notation f(p,q) denoted the element located in row p and column q. Matrices in MATLAB are stored in variables with names such as A, a, RGB, real_array and so on. Variables must begin with a letter and contain letters, numerals and so on.

Some MATLAB functions to handle digital images are like,

- Images are displayed on the MATLAB desktop using function *imshow* which has the basic syntax:

  imshow(f,G) where f is an image and G is the number of intensity levels used to display it. If G is omitted it defaults to 256 levels using syntax *imshow(f,[low,high])*.
- Images are read into MATLAB using function *imread('filename')*.

  Different images with proper extension name is read from different saved format.

Format name	Description	Extension
Tiff	Tagged image	.tif
JPEG	Joint photographic Expert group	jpg
GIF	Graphic Interchange Format	gif
BMP	Window Bitmap	bmp
PNG	Portable N/W graphics	png

In the JPEG image compression algorithm, the input image is divided into 8-by-8 or 16-by-16 blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, coded, and transmitted. The JPEG receiver (or JPEG file reader) decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image. For typical images, many of the DCT coefficients have values close to zero; these coefficients can be discarded without seriously affecting the quality of the reconstructed image. Such algorithm results particularly robust also for face identification. Moreover the 2D DCT operator can be applied to overlapping data.

The code has been tested with AT&T database achieving an excellent recognition rate of 99.20% (40 classes, 5 training images and 5 test images for each class, hence there are 200 training images and 200 test images in total randomly selected and no overlap exists between the training and test images).

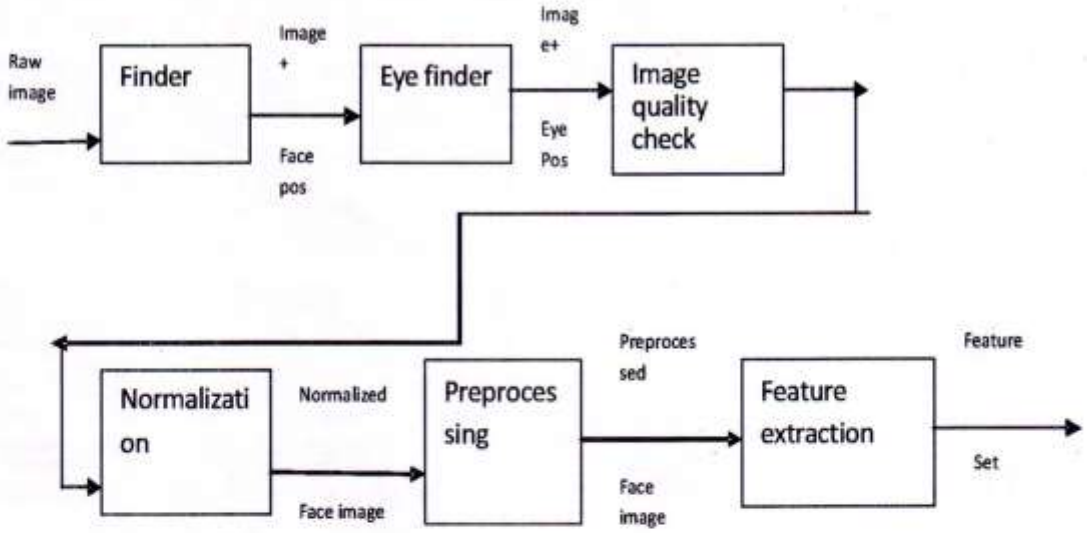
Face Detection Steps and Progressive Masking:

After detecting peaks at output of the matched filter, the following steps are taken:

Peaks within threshold range-> faces.

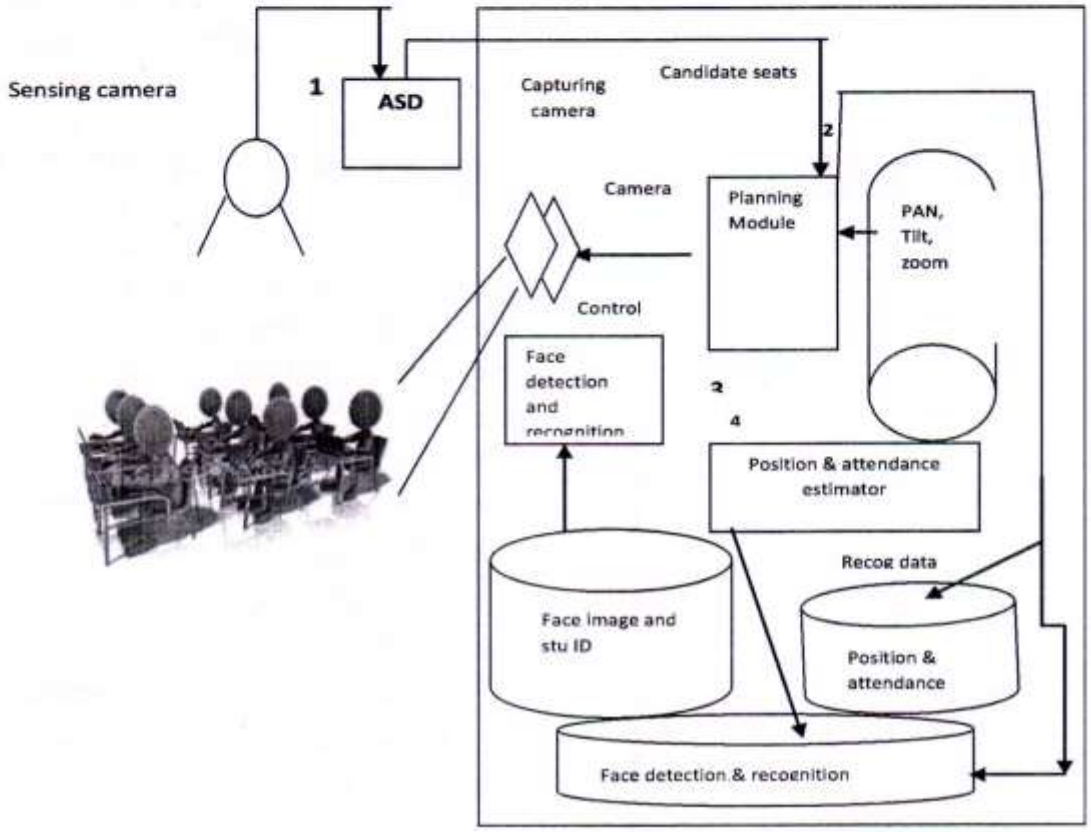
Face pixels are convolved with oval remove neighborhood of detected face pixel.

After all processing face pixels are consolidated into blobs by dilation. Finally, centroids of blobs deemed to be face centers.



Face recognition attendance system:

The e-attendance system based on face recognition can be likely



Algorithm:

1. Count total students sum using sum(A) where A is a stored array of students.
2. Student features eg. Retina identified by sensing camera.
3. Features saved by capturing camera.
4. Using planning module students are seated and seats are arranged with required pan tilt and zoom.
5. Captured images are stored in a database. Features are recognized and detected applying face detection algorithm using MATLAB. MATLB uses matrix manipulation, plotting of function and data implementation algorithms, creation of user interfaces and interfacing of programs written in other languages like C, C++,JAVA.

```
Public class FaceRecMain{
Public static void main(String[],args){
{EventQueue.invokeLater(new Runnable(){
@override
Public void run()
{
FaceRecView
View=newFaceRecView("FACERECOGNITION");
FaceRec model=new FaceRec();
New SimpleController(view,model);
}
};
}
Public static void debug(String msg){
System.out.println(msg);
}
}
```

JAVA codes of face detection is shown

6. Face images are matched with student ID and database.
7. Using attendance estimator the student attendances are calculated.

Conclusion:

Some problems can be faced to apply this system efficiently. Nothing is 100% perfect in real world. There are lots of hackers always try to hack the security system. Again the problems in the light contrast of face in different environment and the hit matrix always may not identify the faces perfectly.

- Not only the skin color but the attempt on face in every day is not perfect to previous image.
- In most cases effective use of color space-face color relationships and morphological processing allowed effective pre-processing.
- For images trained on, able to detect faces with reasonable accuracy and miss and false alarm rate.

General result:

