FACE RECOGNITION IN VIDEOS

Project submitted in partial fulfillment of requirements

For the Degree of

BACHELOR OF ENGINEERING

BY

BHARGAV ASHTEKAR

REHAN SAYED

ANSH MAKHIJA

NEHA BORKAR

Under the guidance of

Internal Guide

Prof. K. T. TALELE

Department of Electronics Engineering

Sardar Patel Institute Of Technology

University of Mumbai

2009-2010

BHARTIYA VIDYA BHAVAN'S

SARDAR PATEL INSTITUTE OF TECHNOLOGY

MUNSHI NAGAR, ANDHERI (W),

MUMBAI - 400 058.

2009-10

CERTIFICATE OF APPROVAL

This is to certify that the following students

BHARGAV ASHTEKAR

REHAN SAYED

ANSH MAKHIJA

NEHA BORKAR

have satisfactorily carried out the project work entitled

"FACE RECOGNITION IN VIDEOS"

towards the fulfillment of Bachelor of Engineering course in Electronics of the Mumbai University

Prof. K. T. Talele
INTERNAL GUIDE

Prof. Dr. S. T. Gandhe
H.O.D ELECTRONICS

Dr. Prachi Gharpure
PRINCIPAL

UNIVERSITY OF MUMBAI

2009-2010

PROJECT APPROVAL CERTIFICATE

This is to certify that the project entitled "**FACE RECOGNITION IN VIDEOS"** has been duly completed by the following students

BHARGAV ASHTEKAR
REHAN SAYED
ANSH MAKHIJA
NEHA BORKAR

Under the guidance of Prof. K. T. Talele in the partial fulfillment of the requirement for the award of degree of Bachelors Of Engineering (Electronics) of the university of Mumbai.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DEPARTMENT OF ELECTRONICS ENGINEERING
SARDAR PATEL INSTITUTE OF TECHNOLOGY
MUNSHI NAGAR, ANDHERI (W), MUMBAI-40005

ACKNOWLEDGEMENTS

We feel privileged to thank **Prof. K. T. Talele** our project guide for providing all the support needed for successful completion of this project.

We would like to express our gratitude towards his constant encouragement, support and guidance throughout the project.

We thank **Mr. Atul Tikare**, our Project Coordinator and **Mr. Sunil Kadam** for their patience and co-operation throughout the project.

Last but not the least; we would also like to express our thanks to all the helping hands which directly or indirectly helped us in our project analysis and design.

Table of Contents

List of Figures

ABSTRACT

Given a video, face recognition is to determine whether or not the specified person is in the video.
The research for face detection and recognition is focusing on

1) Video to Frame Conversion: To sample video at 15 frames per second.
2) Face Detection: To detect the faces in the frames
3) Face Recognition: To identify the person in the video

The video is taken in .avi or .mpg format thereby calculating the total no. of frames present in a video and then sampling it at a predefined rate. We develop a technique to identify an unknown person in an video image by using **SMQT** (Successive Mean Quantization Transform) and **SNOW** classifier and **Template matching**.

SMQT is used to extract features from the local area of an image. The SMQT uses an approach that performs an automatic structural breakdown of information. These properties will be employed on local areas in an image to extract illumination insensitive features.
Intensity is the product of illumination and reflectance components where reflectance represents features of a face. This algorithm extracts reflectance component and trains the SNOW classifier which creates a look up table for classification as face or non face. Usually the recognition methods vary with respect to object detection applications.

The basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which we want to detect. This technique can be easily performed on grey images or edge images. The convolution output will be highest at places where the image structure matches the mask structure, where large image values get multiplied by large mask values.

This method is normally implemented by first picking out a part of the search image to use as a template: We will call the search image **S(x, y)**, where **(x, y)** represent the coordinates of each pixel in the search image. We will call the template $\mathbf{T(x_t, y_t)}$, where $\mathbf{(x_t, y_t)}$ represent the coordinates of each pixel in the template. We then simply move the center (or the origin) of the template $\mathbf{T(x_t, y_t)}$ over each **(x, y)** point in the search image and calculate the sum of products between the coefficients in **S(x, y)** and $\mathbf{T(x_t, y_t)}$ over the whole area spanned by the template. As all possible positions of the template with respect to the search image are considered, the position with the highest score is the best position. This method is sometimes referred to as 'Linear Spatial Filtering' and the template is called a filter mask.

The system has 100% detection rate and 97-98% Recognition rate.

CHAPTER 1

INTRODUCTION

1.1 Face Recognition in Videos

Face detection and recognition is interesting to study because it is an application area where computer vision research is being utilized in both military and commercial products. Much effort has been spent on this problem, yet there is still plenty of work to be done.

Basic research related to this field is currently active. Often, practical applications can grow out of improvements in theoretical understanding and it seems that this problem will continue to demonstrate this growth.

Personally, we are interested in this project because it’s a pattern recognition problem in which humans are very adept, whereas it can be quite challenging to teach a machine to do it. The intermediate and final visual results are interesting to observe in order to understand failures and successes of the various approaches.

Face detection and recognition is challenging because it is a real world problem. The human face is a complex, natural object that tends not to have easily (automatically) identified edges and features. Because of this, it is difficult to develop a mathematical model of the face that can be used as prior knowledge when analyzing a particular image.

Applications of face detection and recognition are widespread. Perhaps the most obvious is that of human computer interaction. One could make computers easier to use if when one simply sat down at a computer terminal, the computer could identify the user by name and automatically load personal preferences. This identification could even be useful in enhancing other technologies such as speech recognition, since if the computer can identify the individual who is speaking, the voice patterns being observed can be more accurately classified against the known individual’s voice.

Human face detection and recognition technology could also have uses in the security domain. Recognition of the face could be one of several mechanisms employed to identify an individual. Face recognition as a security measure has the advantage that it can be done quickly, perhaps even in real time, and does not require extensive equipment to implement. It also does not pose a particular inconvenience to the subject being identified, as is the case in retinal scans. It has the disadvantage, however, that it is not a foolproof method of authentication, since human face appearance is subject to various sporadic changes on a day-to-day basis (shaving, hair style, acne, etc), as well gradual changes over time (aging). Because of this, face recognition is perhaps best used as an augmentation for other identification techniques.

A final domain in which face recognition techniques could be useful is search engine technologies. In combination with face detection systems, one could enable users to search for specific people in images. This could be done by either having the user provide an image of the person to be found, or simply providing the name of the person for well-known individuals. A specific application of this technology is criminal mug shot databases. This environment is perfectly suited for automated face recognition since all poses are standardized and lighting and scale are held constant. Clearly, this type of technology could extend online searches beyond the textual clues that are typically used when indexing information.

Every face or image that we can see on machine is in its 2 dimensional form. There are many different technologies available today to uniquely identify a person’s identity. Many of which like Password/PIN known as Personal Identification Number systems are the most common in practice today. However these systems have their own intrinsic drawbacks. Passwords can be forgotten and worse if they are lost or stolen, person identity can be misused by somebody else. In order to overcome these problems there has been a considerable interest in "biometrics" identification systems, which use pattern recognition techniques to identify people using their unique characteristics. Some of those methods are fingerprints and retina and iris recognition. But these are obtrusive and expensive.

2D face recognition has a natural place in the present and the future environment because it’s unobtrusive and passive in nature. It does not restrict the movements of an individual during recognition.

Our goal at the end of the project is to develop ‘Face Recognition in videos’ which works efficiently and is simple to understand. In addition, we try to implement it for User Authentication Systems too.

CHAPTER 2

LITERATURE REVIEW

2.1 Face Detection using local SMQT features and split up SNOW classifier

The paper 'Face Detection using local SMQT Features and Split up SNOW Classifier' published by *Mikael Nilsson, J¨orgen Nordberg, and Ingvar Claesson* consists of a novel approach in which the local Successive Mean Quantization Transform features are proposed for illumination and sensor insensitive operation in object recognition. Secondly, a split up Sparse Network of Winnows is presented to speed up the original classifier. Finally, the features and classifier are combined for the task of frontal face detection.

2.2 'A Face Recognition System Using Template Matching and Neural Network Classifier'

The paper 'A Face Recognition System Using Template Matching And Neural Network Classifier' published by *Muhammad Firdaus Hashim, Puteh Saad, Mohd Rizon , Mohamed Juhari and Shahrul Nizam Yaakob* consists of novel approach in which the system will identify the input person in a face image by using the technique of Template Matching.In this paper the features are adopted to represent a face under environmental changes and classified based on the chosen representation.The proposed system is to identify the unknown person in an image by matching the image face to the templates of reference faces stored in a database. We process the input image by detecting the irises of both eyes. Some of the process involved before the irises can be detected. The processes are extracting the head region, detection of irises, cost of irises and selecting the pair of irises.

2.3 A Wavelet-based Framework for Face Recognition

The paper 'A Wavelet based framework for Face recognition' published by *Christophe Garcia, Giorgos Zikos, Giorgos Tziritas* proposes detecting and recognizing human faces automatically in video data provide users with powerful tools for performing queries.Each face is described by subset of band filtered images containing wavelet coefficients. These coefficients characterize the face texture and a set of simple statistical measures allows us to form compact and meaningful feature vectors. Then, an efficient and reliable probalistic metric derived from the Bhattacharrya distance is used in order to classify the face feature vectors into person classes.

2.4 Face Recognition Using Ensembles of Networks

The paper 'Face Recognition Using Ensembles of Networks' published by *S. Gutta, J. Huang, B. Takacs,* and *H. Wechsler* describes a novel approach for fully automated face recognition and show its feasibility on a large data base of facial images. It combines the merits of 'discrete and abstractive' features with those of 'holistic' template matching'. Training for face detection takes place over both positive and negative images on a large data base consisting of 748 images corresponding to 374 subjects, among them 11 duplicates, yield on the average 87 % correct match, and (ROC curves where) 99 % correct verification is achieved for a 2 % reject rate.

2.5 Image Analysis for Face Recognition

The paper 'Image Analysis for Face Recognition' published by *Xiaoguang Lu presents* a number of typical algorithms being categorized into appearance based and model-based schemes. For appearance-based methods, three linear subspace analysis schemes are presented, and several non-linear manifold analysis approaches for face recognition are briefly described. The model-based approaches are introduced, including Elastic Bunch Graph matching, Active Appearance Model and 3D Morphable Model methods

2.6 Intelligent Face Recognition Techniques: A Comparative Study

The paper 'Intelligent face recognition techniques: A comparative study' published by *S.T.Gandhe*, *K.T.Talele*, *A.G.Keskar* in IAENG International Journal of Computer Science proposes the feasibility of these algorithms for human face identification is presented through experimental investigation. It provides user authentication via facial features. These proposed systems of face recognition may be applied in identification systems, document control and access control

2.7 Template Matching Approach for Pose Problem in Face Verification

The paper 'Template matching Approach for Pose Problem in Face Verification' published by *Anil Kumar Sao* and *B. Yegnanaarayana*, Speech and Vision Laboratory, Department of Computer Science and Engineering, Indian Institute of Technology, Madras propose a template matching approach to address the pose problem in face verification, which neither synthesizes the face image, nor builds a model of the face image. Template matching is performed using edginess-based representation of face images. The edginess-based representation of face images is computed using one-dimensional (1-D) processing of images. It verifies the identity of a person using score obtained from template matching

2.8 Video-based Face Recognition Using Bayesian Inference Model

The paper 'Video-based Face Recognition Using Bayesian Inference Model' published by *Wei Fan*, *Yunhong Wang*, and *Tieniu Tan*, National Laboratory of Pattern Recognition, China effectively combines the facial configuration and temporal dynamics for the face recognition task. Experimental results on a middle-scale video database demonstrate its effectiveness and flexibility.

CHAPTER 3

FACE DETECTION AND FACE RECOGNITION METHODS

3.1 Methods for Face Detection

A first step of any face processing system is detecting the locations in images where faces are present. However, face detection from a single image is a challenging task because of variability in scale, location, orientation (up-right, rotated), and pose (frontal, profile). Facial expression, occlusion, and lighting conditions also change the overall appearance of faces. There are many closely related problems of face detection. Face localization aims to determine the image position of a single face; this is a simplified detection problem with the assumption that an input image contains only one face. Face tracking methods continuously estimate the location and possibly the orientation of a face in an image sequence in real time. With over 150 reported approaches to face detection, the research in face detection has broader implications for computer vision research on object recognition. Numerous methods have been proposed to detect faces in a single image of intensity or color images. Among the face detection methods, the ones based on learning algorithms have attracted much attention recently and have demonstrated excellent results. Here are existing techniques to detect faces from a single intensity or color image.

1. Knowledge-Based Top-Down Methods

In this approach, face detection methods are developed based on the rules derived from the researchers' knowledge of human faces. For example, a face often appears in an image with two eyes that are symmetric to each other, a nose, and a mouth. The relationships between features can be represented by their relative distances and positions. Facial features in an input image are extracted first, and face candidates are identified based on the coded rules. Kotropoulos and Pitas presented a rule-based localization method. In this method, facial features are located with a projection method .The detected features constitute a facial candidate.

2. Bottom-Up Feature-Based Methods

In contrast to the knowledge-based top-down approach, researchers have been trying to find invariant features of faces for detection. Facial features such as eyebrows, eyes, nose, mouth, and hair-line are commonly extracted using edge detectors.

- **Facial Features**

Sirohey proposed a localization method to segment a face from a cluttered background for face identification .It uses an edge map (Canny detector) and heuristics to remove and group edges so that only the ones on the face contour are preserved. An ellipse is then fit to the boundary between the head region and the background. This algorithm achieves 80 percent accuracy on a database of 48 images with cluttered backgrounds. T.K. Leung, M.C. Burl, and P. Perona developed a probabilistic method to locate a face in a cluttered scene based on local feature detectors and random graph matching. Their motivation is to formulate the face localization problem as a search problem in which the goal is to find the arrangement of certain facial features that is most likely to be a face pattern. Five features (two eyes, two nostrils and nose/lip junction) are used to describe a typical face. For any pair of facial features of the same type (e.g., left eye, right-eye pair), their relative distance is computed, and over an ensemble of images the distances are modeled by a Gaussian distribution. They used a set of 150 images for experiments in which a face is considered correctly detected if any constellation correctly locates three or more features on the faces. This system is able to achieve a correct localization rate of 86 percent.

Detection follows 2 stages: focusing and intensive classification. The CART algorithm is applied to grow a classification tree from the training images and a collection of false positives identified from generic background images.

- **Texture**

Human faces have a distinct texture that can be used to separate them from different objects. Augusteijn and Skufca developed a method that infers the presence of a face through the identification of face-like textures. The textures are computed using second-order statistical features (SGLD) on subimages of 16×16 pixels. Three types of features are considered: skin, hair, and others. They used a cascade correlation neural network for supervised classification of textures and a Kohonen self-organizing feature map to form clusters for different texture classes. To infer the presence of a face from the texture labels, they suggest using votes of the occurrence of hair and skin textures. However, only the result of texture classification is reported, not face localization or detection.

- **Skin Color**

Human skin color has been used and proven to be an effective feature in many applications from face detection to hand tracking. Although different people have different skin color, several studies have shown that the major difference lies largely between their intensity rather than their chrominance. Several color spaces have been utilized to label pixels as skin including RGB, normalized RGB, HSV (or HSI), YCrCb, YES, CIE XYZ, and CIE LUV. Saxe and Foulds proposed an iterative skin identification method that uses histogram intersection in HSV color space. S. McKenna, Y. Raja, and S. Gong presented an adaptive color mixture model to track faces under varying illumination conditions. Instead of relying on a skin color model based on color constancy, they used a stochastic model to estimate an object’s color distribution online and adapt to accommodate changes in the viewing and lighting conditions. Preliminary results show that their system can track faces within a range of illumination conditions. However, this method cannot be applied to detect faces in a single image.

- **Multiple Features**

Recently, numerous methods that combine several facial features have been proposed to locate or detect faces. Most of them utilize global features such as skin color, size, and shape to find face candidates, and then verify these candidates using local, detailed features such as eye brows, nose, and hair.

3. Appearance-Based Methods

In contrast to template matching, the models (or templates) are learnt from a set of training images which should capture the representative variability of facial appearance. These learned models are then used for detection. These methods are designed mainly for face detection.

- **Eigenfaces**

Turk and Pentland applied principal component analysis to face recognition and detection. Principal component analysis on a training set of face images is performed to generate the Eigenpictures (here called Eigenfaces) which span a subspace (called the face space) of the image space. Images of faces are projected onto the subspace and clustered. Similarly, nonface training images are projected onto the same subspace and clustered. Since images of faces do not change radically when projected onto the face space, while the projection of nonface images appear quite different. To detect the presence of a face in a scene, the distance between an image region and the face space is computed for all locations in the image. The distance from face space is used as a measure of faceness, and the result of calculating the distance from face space is a face map. A face can then be detected from the local minima of the face map. Many works on face detection, recognition, and feature extractions have adopted the idea of eigenvector decomposition and clustering.

- **Distribution-Based Methods**

Sung and Poggio developed a distribution-based system for face detection which demonstrated how the distributions of image patterns from one object class can be learned from positive and negative examples (i.e., images) of that class. Their system consists of two components, distribution-based models for face/nonface patterns and a multilayer perceptron classifier. A probabilistic visual learning method based on density estimation in a high dimensional space using an eigenspace decomposition was developed by Moghaddam and Pentland. Principal component analysis (PCA) is used to define the subspace best representing a set of face patterns. In terms of face detection, this technique has only been demonstrated on localization. M.-H. Yang, N. Ahuja, and D. Kriegman proposed a detection method based on a mixture of factor analysis (MFA). Factor analysis (FA) is a statistical method for modeling the covariance structure of high dimensional data using a small number of latent variables. A second method in uses Fishers Linear Discriminant (FLD) to project samples from the high dimensional image space to a lower dimensional feature space. In the second proposed method, they decompose the training face and nonface samples into several subclasses using Kohonens Self Organizing Map (SOM) The maximum likelihood decision rule is used to determine whether a face is detected or not.

- **Neural Networks**

R. Vaillant, C. Monrocq, and Y. Le Cun used convolutional neural networks to detect faces in images. Face and nonface images of 20×20 pixels are first created. One neural network is trained to find approximate locations of faces at some scale. Another network is trained to determine the exact position of faces at some scale. Given an image, areas which may contain faces are selected as face candidates by the first network. These candidates are verified by the second network. R. Feraud, O.J. Bernier, J.-E. Villet, and M. Collobert presented a detection method using auto associative neural network. One auto associative network is used to detect frontal-view faces and another one is used to detect faces turned up to 60 degrees to the left and right of the frontal view. A gating network is also utilized to assign weights to frontal and turned face detectors in an ensemble of auto associative networks. S.H. Lin, S.Y. Kung, and L.J. Lin presented a face detection system using probabilistic decision-based neural network (PDBNN).The architecture of PDBNN is similar to a radial basis function (RBF) network with modified learning rules and probabilistic interpretation. Instead of converting a whole face image into a training vector of intensity values for the neural network, they first extract feature vectors based on intensity and edge information in the facial region that contains eyebrows, eyes, and nose. The extracted two feature vectors are fed into two PDBNNs and the fusion of the outputs determine the classification result.

4. SMQT features and SNOW classifier

SMQT is used to extract features from the local area of an image. The SMQT uses an approach that performs an automatic structural breakdown of information. These properties will be employed on local areas in an image to extract illumination insensitive features. The split up SNoW will utilize the result from the original SNoW classifier and create a cascade of classifiers to perform a more rapid detection. It will be shown that the number of splits and the number of weak classifiers can be arbitrary within the limits of the full classifier.

3.2 Methods for Face Recognition

Eigen-face methods, template matching, graph matching, linear subspace method, neural network method and Fisher-face method are used for face recognition.

- *Eigen-face* approach applies the idea feature extraction which greatly reduces the facial feature dimension and yet maintains reasonable discriminating power.
- The *neural network* approach, though some variants of the algorithm work on feature extraction.
- The *template matching* method operates by performing direct correlation of image segments.
- In the *linear subspace* method for each face three or more images are taken under different lighting directions. By using these 3 images, a 3D basis for the linear subspace is constructed.
- To perform recognition, the distance of a new image to each linear subspace is calculated and the face corresponding to the shortest distance is chosen.
- *Fisher-face* method is a refinement of the Eigen-face algorithm. It further reduces the Eigen space by the fisher’s linear discriminate (FLD). FLD selects the subspace in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. It is reported that the Fisher-face algorithm outperforms the Eigen-face algorithm on the facial database with wide variations in lighting conditions.

3.3 Existing Systems

3.3.1 Face Recognition using Eigen Face Approach

Let a face image I(x,y) be a two dimensional N by N array of (8-bit) intensity values. An image may also be considered as a vector of dimension N^2. An ensemble of images, then, maps to a collection of points in this huge space.

Step1: Prepare the data

Let the training set of face images be T_1, T_2, T_3,….,T_M. This training data set has to be mean adjusted before calculating the covariance matrix or eigenvectors.

Step 2: Calculate the mean and subtract from the original face

The average face is calculated as,

$$\psi = \frac{1}{M}\sum_{i=1}^{M} T_i$$

Each image in the data set differs from the average face by the vector $\Phi = T_i - \Psi$.

Step 3: Calculate the co-variance matrix

The covariance matrix is

$$C = \frac{1}{M}\sum_{i=1}^{M} \Phi_i \Phi_i^T \qquad (1)$$

$$= AA^T$$

where $A = [\Phi_1 \Phi_2 \cdots \Phi_M]$

Step 4: calculate the eigenvectors and eigenvalues of the co-variance matrix

The matrix C is a N^2 by N^2 matrix and would generate N^2 eigenvectors and eigenvalues. For large values of N, a computationally feasible method was suggested to find out the eigenvectors. If the number of images in the training set is less than the no of pixels in an image (i.e. $M < N^{2)}$, then we can solve an M by M matrix instead of solving a N^2 by N^2 matrix.

Consider the covariance matrix as A^TA instead of AA^T. Now the eigenvector v_i can calculated as follows,

$$A^T A v_i = \mu_i v_i \qquad (2)$$

where μ_i is the Eigen value. Here the size of covariance matrix would be M by M.Thus we can have M eigenvectors instead of N^2.

Premultipying equation 2 by A, we have

$$AA^T A v_i = \mu_i A v_i \qquad (3)$$

The right hand side gives us the M eigenfaces of the order N^2 by 1.All such vectors would make the image space of dimensionality M.

Step 5: Selecting the principal components

As the accurate reconstruction of the face is not required, we can now reduce the dimensionality to M' instead of M. This is done by selecting the M' eigenfaces which have the largest associated eigenvalues. These eigenfaces now span a M'-dimensional subspace instead of N^2.

Step 6: Classifying the faces

The process of classification of a new (unknown) face to one of the known faces proceeds in two steps. First a new image T is transformed into its Eigen-face components (projected into 'face space') by a simple operation,

$$w_k = u_k^T (T - \psi) \quad (4)$$

here k = 1, 2,….M'.The weights obtained as above form a weight vector $\Omega^T = [w_1, w_2, w_3 \cdots w_{M'}]$ that describes the contribution of each Eigen-face in representing the input face image. The Euclidean distance of the weight vector of the new image from the face class weight vector can be calculated as follows,

$$\varepsilon_k = \|\Omega - \Omega_k\| \quad (5)$$

where Ω_k is the vector describing the k^{th} face class. The face is classified as belonging to class k when the distance ε_k is below some threshold value θ_ε. Otherwise the face is classified as unknown. Also it can be found whether an image is a face image or not by simply finding the squared distance between the mean adjusted input image and its projection onto the face space.

$$\varepsilon^2 = \|\Phi - \Phi_f\| \quad (6)$$

where Φ_f is the face space and $\Phi = T_i - \Psi$ is the mean adjusted input.

With this we can classify the image as known face image, unknown face image and not a face image.

3.3.2 Principal component analysis cascaded with discrete wavelet transform

We use term Wavelet PCA to refer to computing principle components for a masked or modified set of wavelet coefficient to find Wavelet PCA eigenspectra7+ in spectral domain and then projecting the original image onto the wavelet PCA Eigen spectra basis. In this way, features at a particular scale are indirectly emphasized by the computed projection basis enhancing the reduced dimensionality images without filtering artifacts.

Calculation of DWT of A Given Image:

The Square Two-Dimensional DWT is calculated using a series of one dimensional DWTs which is done as follows:

Step 1: Replace each image row with its 1D DWT
Step 2: Replace each image column with its 1D DWT
Step 3: Repeat steps (1) and (2) on the lowest sub-band to create the next scale
Step 4: Repeat step (3) until the desired number of scales has been created

3.3.3 Template Matching

Template matching is a technique in Digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to mobile robot, or as a way to detect edges in images. There are different approaches to accomplishing template matching. Some are more performing than others, and some find better matches.

The basic method of template matching is to loop through all the pixels in the search image and compare them to the pattern. While this method is simple to implement and understand, it is one of the slowest methods.

This method is normally implemented by firstly creating a subimage (the template), we will call this subimage 'w(x,y)' where x and y represent the coordinates of each pixel. We then simply move the center of this subimage w over each (x,y) point in the a candidate image, which we will call 'o(x,y)' and calculate the sum of products between the coefficients in o and the corresponding neighbourhood pixels in the area spanned by the filter mask. This method is sometimes referred to as 'Linear Spatial Filtering'

This type of spatial filtering is normally only used in dedicated hardware solutions because of the computational complexity of the operation, however we can lessen this complexity by filtering it in the frequency domain of the image, referred to as 'frequency domain filtering,' this is done through the use of the convolution theorem.

Another way to make the matching faster is to reduce the image into smaller images, and then search the smaller subimages for a match to the template, these smaller images are often referred to as an image pyramid (i.e. a 128 x 128px image can have a pyramid of smaller images which are 64x64 32x32, 16x16 etc).

After finding matches in the smaller images, that information is used in the larger image as a centre location. The larger image is then searched in a small window to find the best location of the pattern.

The features of template matching technique are:

- Concept: simple contour matching
- Easy to understand
- Evaluates the entire image for comparison based on the pre-defined window size
- Clean and a simple method
- Cost efficient than other method

CHAPTER 4

PROPOSED SYSTEM

4.1 Problem Statement

4.1.1 Broad problem statement

Be it for purposes of security or human–computer interaction, there is wide application to face recognition in videos. Recent developments in computer technology and the call for better security applications have turned face-recognition into focus. Face recognition systems have become the subject of increased interest.

Chances of unauthorized access to critical information have become a major concern. So a system must be implemented so that the security system becomes more robust and probability of unauthorized access to the system becomes minimum.

Step 1: To convert video into frames
We can take the video in 2 formats: .mpg or .avi. We have implemented our project by taking the video in .avi format. The image of the person is captured and scaled down to appropriate levels.

Step 2: To perform face detection
To build a face detection system, we have used the SMQT features and the SNOW classifier. Then we have performed normalization on images for increasing the dynamic range thereby improving the contrast.

Step 3: To perform face recognition
We have used the technique of Template Matching. Template matching method is normally implemented by firstly creating a subimage (the template), we will call this subimage 'w(x,y)' where x and y represent the coordinates of each pixel. We then simply move the center of this subimage w over each (x,y) point in the a candidate image, which we will call 'o(x,y)' and calculate the sum of products between the coefficients in o and the corresponding neighbourhood pixels in the area spanned by the filter mask. This method is sometimes referred to as 'Linear Spatial Filtering'.

4.2 Methodology

We implement Face Recognition using Template matching.

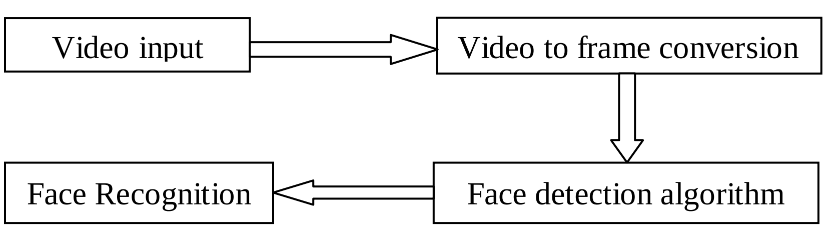


Figure 4.1: Block diagram for Face Recognition in Videos

Initially, the input is taken in video format. This video obtained has to be analyzed for face recognition. The first step is converting the video into frames. This is done by sampling the input video at a predefined rate .The frames obtained are pre-processed. Any image obtained through the capturing devices contains noise due to imperfections in lighting conditions, problems in the capturing device. Also, there is information that is irrelevant to the problem being solved. Pre processing is applied for image enhancement, noise removal and localization. Then the face detection algorithm is applied on the pre- processed image. The face detection algorithm which we are using for face detection is **SMQT features and SNOW classifier.** Then these images act as input images which are stored in a separate folder (detected faces). Now the face recognition algorithm is applied. The algorithm that we are using is template matching because it has a higher success rate as compared to other algorithms. Initially we create a database of images which are considered as template images. Now the algorithm compares the template images with the detected images. The basic method of template matching is to loop through all the pixels in the detected images and compare them to the template images. While this method is simple to implement and understand, it is one of the slowest methods. This procedure is repeated for all the detected images. If a match is found then the image is displayed along with its name. Hence the faces are successfully recognized with acceptable accuracy.

4.2.1 Face detection

4.2.1.1 SMQT Features and SNOW Classifier

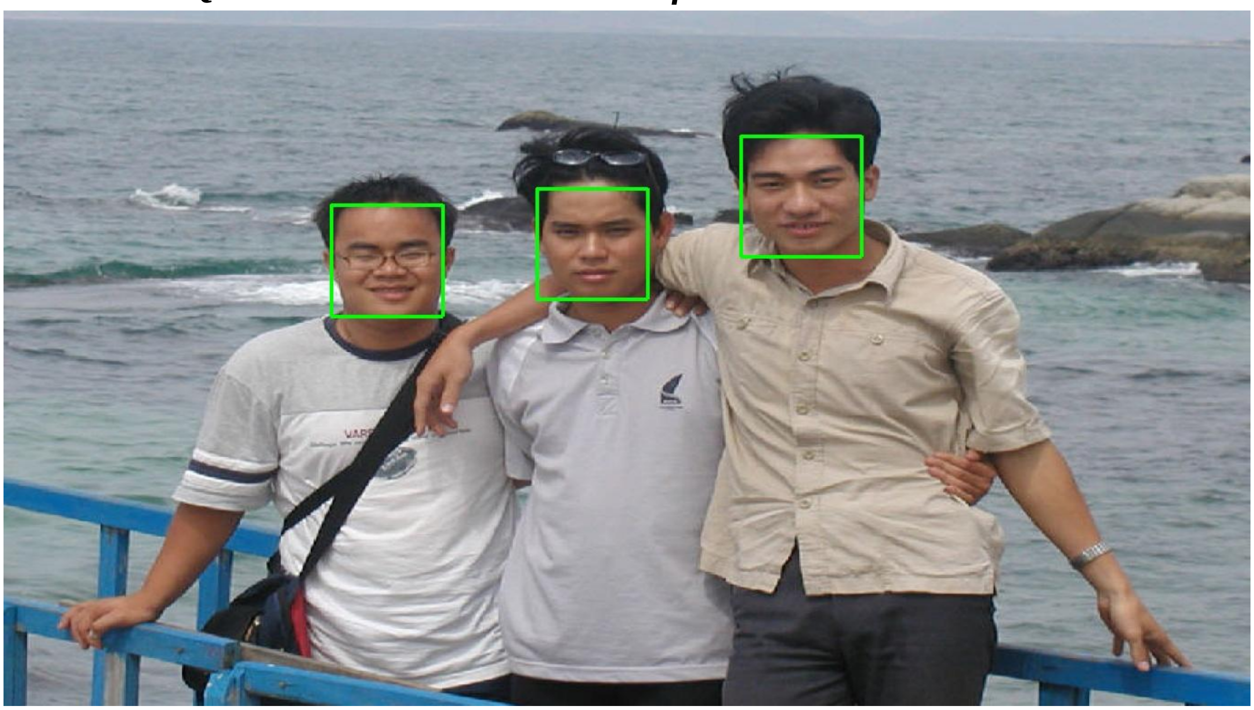


Figure 4.2: Face detection using SMQT and SNOW

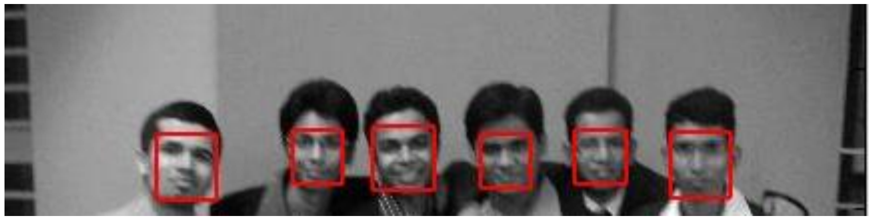


Figure 4.2: Face detection using SMQT and SNoW

Every face detection method is designed in a particular context that is why it is not always easy to compare the results between them. Some detectors have for instance only goal to have a detection rate as near as possible from 100 percent. Even if we naturally want to reach good detection rates, we need to build a real-time oriented detector. So the goal is to detect all the faces (or almost all of them) even if this means we have to accept a higher false positive rate (non-face images labeled as face by the detector). This choice is only in order to respect most of applications which need for example to detect all the people in front of a video camera (Video surveillance for instance).

On the other hand, if for example, a camera is placed in a airport hall, the faces are often low resolution faces, at different scales and the background seems to be quite textured and complicated. In this way, the detector should be robust with respect to illumination, face variation and face size. On the other side, if I keep in mind that I want to detect faces for a further face recognition or comprehension, it would be good to select only faces which can be considered as frontal faces, this will clarifies the choice of the training set used to learn the final classifier. To summarize, even if I could choose other face detection contexts, this one seems to be the most used in the real-world applications.

Illumination and sensor variation are major concerns in visual object detection. It is desirable to transform the raw illumination and sensor varying image so the information only contains the structures of the object. Pattern recognition in the context of appearance based face detection can been approached in several ways

The Successive Mean Quantization Transform (SMQT) can be viewed as a tunable tradeoff between the number of quantization levels in the result and the computational load. SMQT is used to extract features from the local area of an image. The SMQT uses an approach that performs an automatic structural breakdown of information. These properties will be employed on local areas in an image to extract illumination insensitive features. Local areas can be defined in several ways. For example, a straight forward method is to divide the image into blocks of a predefined size. Another way could be to extract values by interpolate points on a circle with a radius from a fixed point. Nevertheless, once the local area is defined it will be a set of pixel values. Let x be one pixel and $D(x)$ be a set of $|D(x)| = D$ pixels from a local area in an image. Consider the SMQT transformation of the local area

$$\text{SMQT}L : D(x) \longrightarrow M(x) \qquad (1)$$

which yields a new set of values. The resulting values are insensitive to gain and bias. These properties are desirable with regard to the formation of the whole intensity image $\mathbf{I}(x)$ which is a product of the reflectance R(x) and the luminance E(x).Additionally, the influence of the camera can be modeled as a gain factor g and a bias term b. Thus, a model of the image can be described by

$$I(x) = \mathrm{g}E(x)R(x) + \mathrm{b} \qquad (2)$$

In order to design a robust classifier for object detection the reflectance should be extracted since it contains the object structure. In general, the separation of the reflectance and the luminance is an ill posed problem. A common approach to solving this problem involves assuming that **E**(x) is spatially smooth. Further, if the luminance can be considered to be constant in the chosen local area then **E**(x) is given by

$$\mathbf{E}(x) = E, \quad \forall x \in D \qquad (3)$$

Given the validity of Eq. 3, the SMQT on the local area will yield illumination and camera-insensitive features.

The local SMQT features can be used as feature extraction for object detection. The features are found to be able to cope with illumination and sensor variation in object detection.Further, the split up SNoW was introduced to speed up the standard

Split Up SNoW Classifier: The split up SNoW will utilize the result from the original SNoW classifier and create a cascade of classifiers to perform a more rapid detection. It will be shown that the number of splits and the number of weak classifiers can be arbitrary within the limits of the full classifier. Further, a stronger classifier will utilize all information gained from all weaker classifiers. The SNoW learning architecture is a sparse network of linear units over a feature space. One of the strong properties of SNoW is the possibility to create lookup-tables for classification. The split up SNoW classifier requires only training of one classifier network.

4.2.1.2 Face Detection Training and Classification

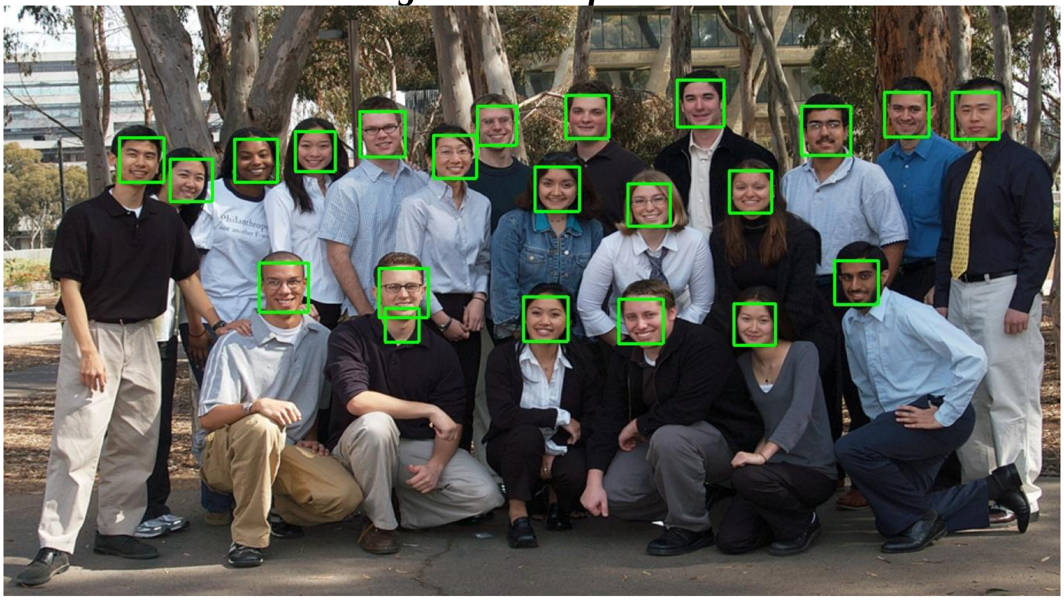


Figure 4.3 – Face Detection Training and Classification

Figure 4.3: Face Detection Training and Classification

In order to scan an image for faces, a patch of 32×32 pixels is applied. This patch is extracted and classified by jumping $\Delta x = 1$ and $\Delta y = 1$ pixels through the whole image. In order to find faces of various sizes, the image is repeatedly downscaled and resized with a scale factor $Sc = 1.2$. To overcome the illumination and sensor problem, the proposed local SMQT features are extracted. Each pixel will get one feature vector by analyzing its vicinity. This feature vector can further be recalculated to an index.

Face Database:
Images are collected using a digital camera containing a face, and are hand-labelled with three points; the right eye, the left eye and the center point on outer edge of upper lip (mouth indication).Using these three points the face will be warped to the 32×32 patch using different destination points for variation.

Nonface Database:
Initially the nonface database contains randomly generated patches. A classifier is then trained using this nonface database and the face database. A collection of videos are prepared from clips of movies containing no faces and are used to bootstrap the database by analyzing all frames in the videos. Every false positive detection in any frame will be added to the nonface database. The nonface database is expanded using this bootstrap methodology. In final training, a total of approximately one million nonface patches are used after bootstrapping.

Face detection is a required first step in face recognition systems. It also has several applications in areas such as video coding, video conference, crowd surveillance and human-computer interfaces. As mentioned above, a framework for face detection is proposed using the illumination insensitive features gained from the local SMQT features and the rapid detection achieved by the split up SNoW classifier. Face detection is one of the classic problems in computer vision, with applications in many areas like surveillance, robotics and multimedia processing. Developing a face detection system is still an open problem, but there have been important successes over the past several years.

4.2.2 Face recognition

4.2.2.1 Template matching

The basic method of template matching uses a convolution mask (template), tailored to a specific feature of the search image, which we want to detect. This technique can be easily performed on grey images or edge images. The convolution output will be highest at places where the image structure matches the mask structure, where large image values get multiplied by large mask values. This method is normally implemented by first picking out a part of the search image to use as a template: We will call the search image $S(x, y)$, where (x, y) represent the coordinates of each pixel in the search image. We will call the template $T(x_t, y_t)$, where (x_t, y_t) represent the coordinates of each pixel in the template. We then simply move the center (or the origin) of the template $T(x_t, y_t)$ over each (x, y) point in the search image and calculate the sum of products between the coefficients in $S(x, y)$ and $T(x_t, y_t)$ over the whole area spanned by the template. As all possible positions of the template with respect to the search image are considered, the position with the highest score is the best position. This method is sometimes referred to as 'Linear Spatial Filtering' and the template is called a filter mask.

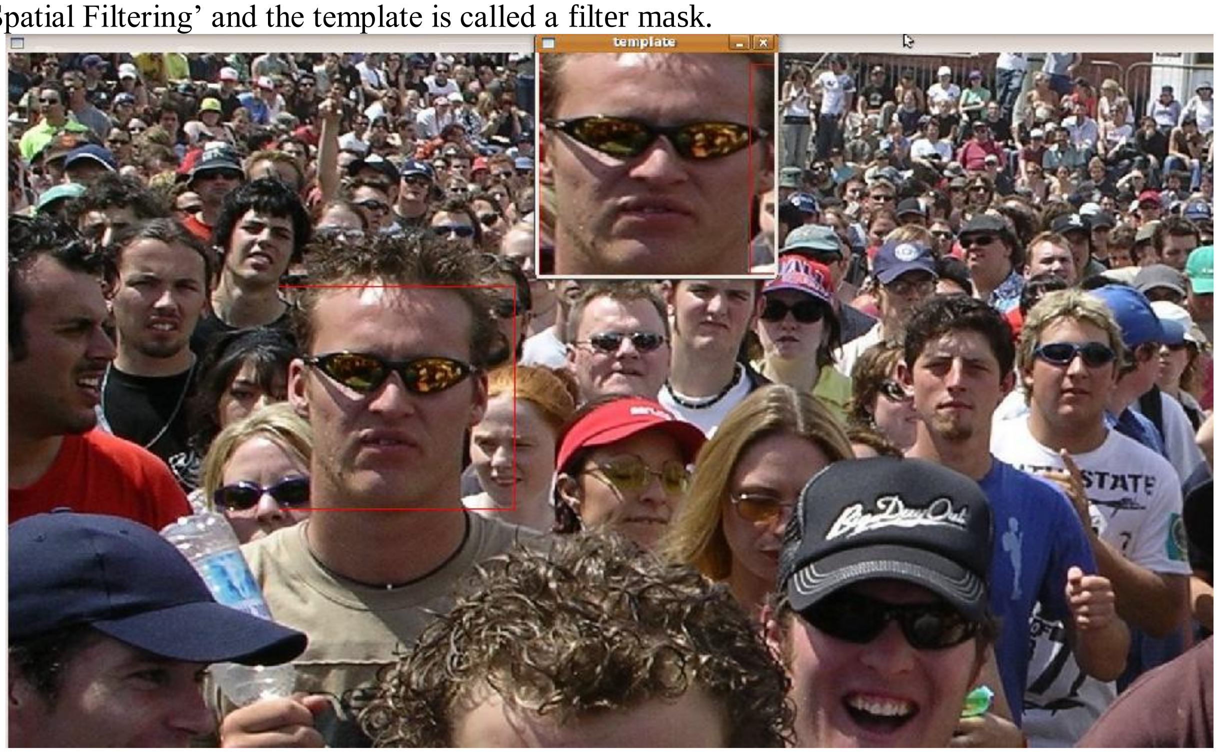


Figure 4.4 – Template Matching example

For example, one way to handle translation problems on images, using template matching is to compare the intensities of the pixels,, using the SAD (Sum of Absolute Differences) measure.

A pixel in the search image with coordinates (x_s, y_s) has intensity Is(x_s, y_s) and a pixel in the template with coordinates (xt, yt) has intensity It(xt , yt). Thus the absolute difference in the pixel intensities is defined as Diff(x_s, y_s, x_t, y_t) = | Is(x_s, y_s) – It(x_t, y_t) |.

In the past, this type of spatial filtering was normally only used in dedicated hardware solutions because of the computational complexity of the operation, however we can lessen this complexity by filtering it in the frequency domain of the image, referred to as 'frequency domain filtering,' this is done through the use of the convolution theorem. Other methods can handle problems such as translation, scale and image rotation.

CHAPTER 5

ALGORITHM

5.1 Face Detection

1. The Successive Mean Quantization Transform (SMQT) can be viewed as a tunable tradeoff between the number of quantization levels in the result and the computational load. SMQT is used to extract features from the local area of an image. The SMQT uses an approach that performs an automatic structural breakdown of information. These properties will be employed on local areas in an image to extract illumination insensitive features.

2. Let x be one pixel and $D(x)$ be a set of $|D(x)| = D$ pixels from a local area in an image. Consider the SMQT transformation of the local area

SMQT L : $D(x) \longrightarrow M(x)$

which yields a new set of values.

3. These properties are desirable with regard to the formation of the whole intensity image $\mathbf{I}(x)$ which is a product of the reflectance $\mathrm{R}(x)$ and the luminance $\mathrm{E}(x)$.Additionally, the influence of the camera can be modeled as a gain factor g and a bias term b. Thus, a model of the image can be described by

$$I(x) = \mathrm{g}E(x)R(x) + \mathrm{b}$$

4. In general, the separation of the reflectance and the luminance is an ill posed problem. A common approach to solving this problem involves assuming that $\mathbf{E}(x)$ is spatially smooth. Further, if the luminance can be considered to be constant in the chosen local area then $\mathbf{E}(x)$ is given by

$$\mathbf{E}(x) = E, \quad \forall x \in D$$

The SMQT on the local area will yield illumination and camera-insensitive features. The local SMQT features can be used as feature extraction for object detection

5.2 Template matching

The algorithm for template matching using NCC is implemented in MATLAB. The following algorithm does the template matching and uses the Cauchy-Sehwartz's inequality to simplify the procedure.

i. Load the original image and template.
ii. Pad the image on all the sides with zeros so that the centre of the template falls on the very first pixel of the main image when kept on the top-left corner as shown in the figure

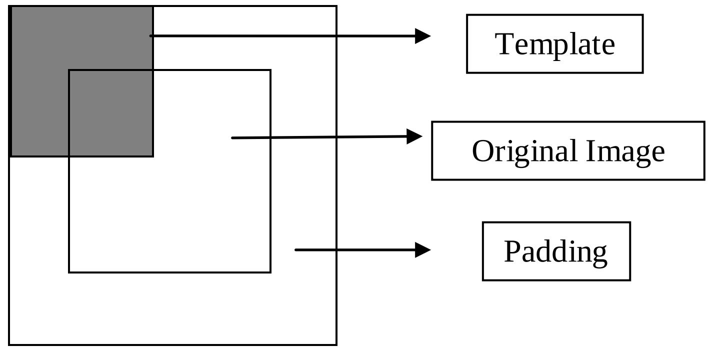


Figure 5.1 Template Matching

a. Calculate the size of the template.
b. Pad rows of zeros on the top and bottom of main image. The number of rows is equal to the size of template in y-direction divided by 2.
c. Pad columns of zeros on the left and right side of the above image with number of columns equal to the size of the template in z-direction divided by 2 and the length of columns being main image size in y-direction plus size of the template in y-direction.

iii. Now, more the mask over the entire image and simultaneously calculate the value of summation of template padded image under the template and store it in an array.
iv. Also calculate the values padded image under the template’s square and sum all the values. Take the square root of the obtained value and store it in an array.
v. Divide the result obtained in step iii. by the result obtained in step iv.
vi. Find the position where the maximum value in the above result falls. The co-ordinates so obtained will give the best match of the template and calculate maximum cross correlation coefficient.
vii. Recover the template from the main image using the above obtained co-ordinates and the size of template.

5.3 FLOWGRAPH

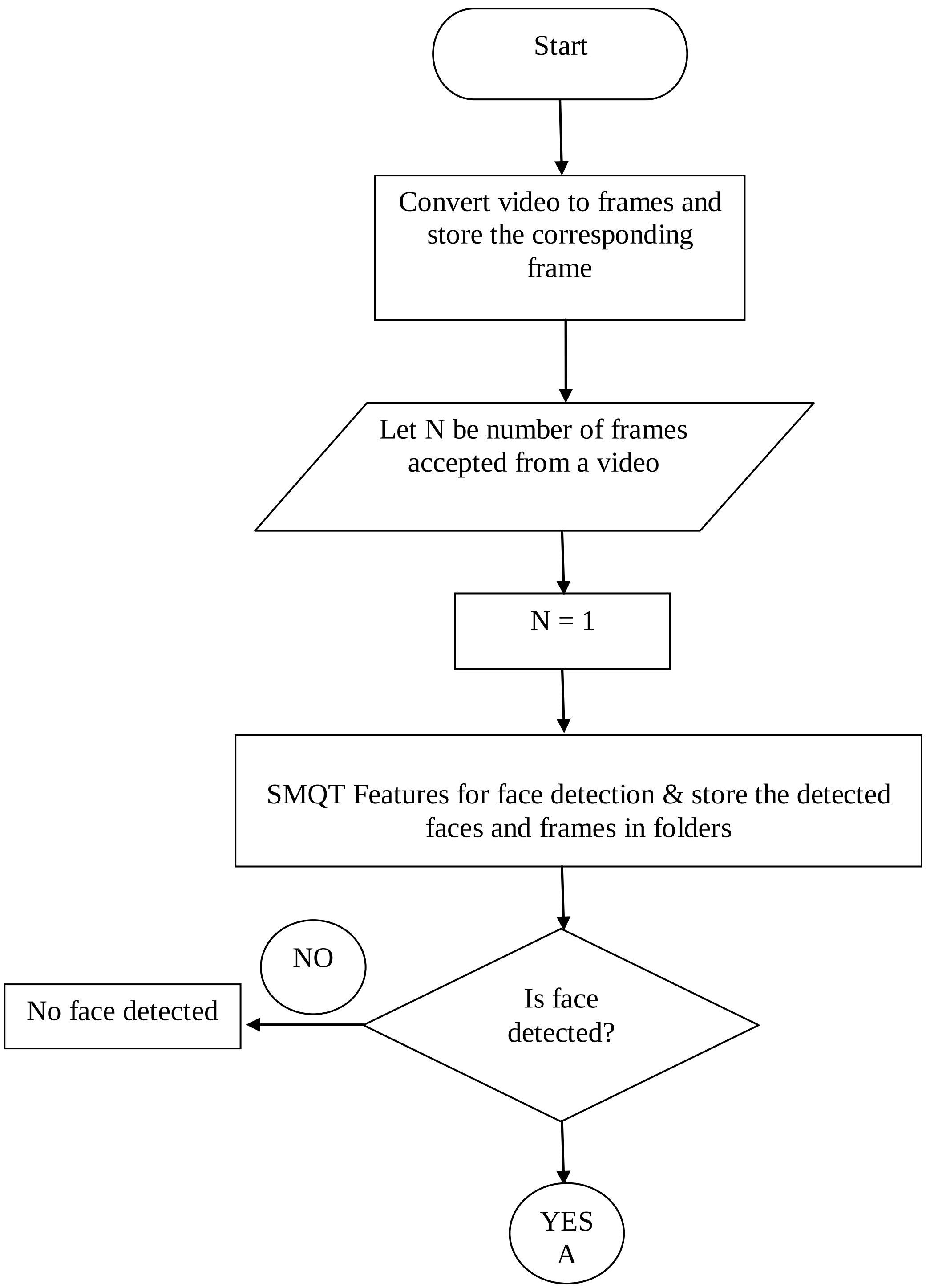


Figure 5.2 Algorithm (part 1)

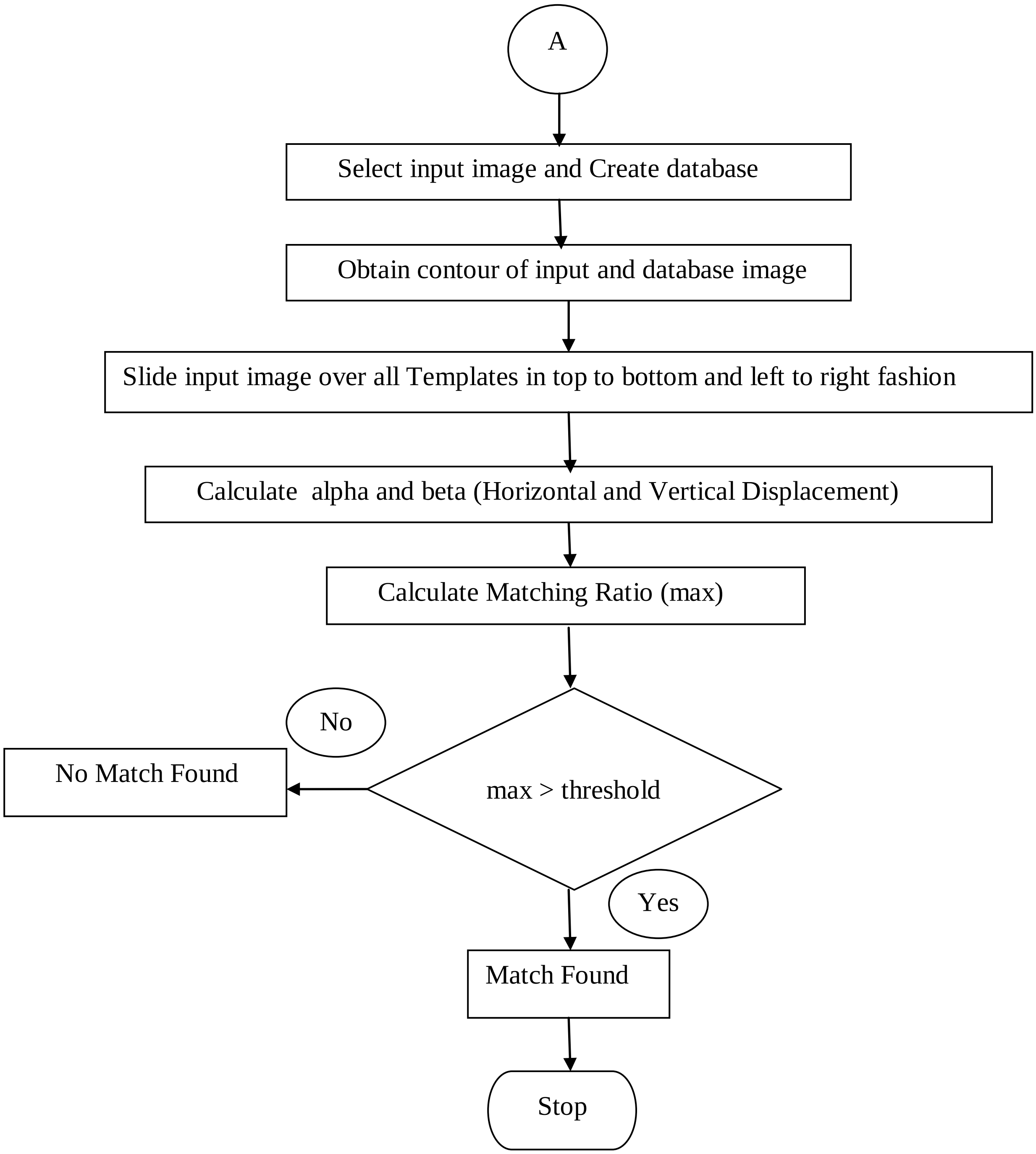


Figure 5.2 Algorithm (part 2)

CHAPTER 6

EXPERIMENTATION & RESULT ANALYSIS

6.1 Performance Parameters

6.1.1 Matching Ratio

Matching Ratio can be defined as the measure of similarity between two images where matching ratio 1 denotes an exactly similar image.

6.1.2 False Acceptance Rate (FAR)

In biometrics, the instance of a security system incorrectly verifying or identifying an unauthorized person is known as false acceptance. Also referred as a *type II error*, a false acceptance typically is considered the most serious of biometric security errors as it gives unauthorized users access to systems that expressly are trying to keep them out.

The *false acceptance rate,* or FAR, is the measure of the likelihood that the biometric security system will incorrectly accept an access attempt by an unauthorized user.

$$FAR = \frac{\text{Number of false acceptances}}{\text{Number of identification attempts}}$$

6.1.3 False Rejection Rate (FRR)

In biometrics, the instance of a security system failing to verify or identify an unauthorized person is known as false rejection. Also referred as a *type I error*, a false rejection does not necessarily indicate a flaw in the biometric system for example, in face recognition base system a blurred image can cause a false rejection of the authorized user.

The *false rejection rate*, or FRR, is the measure of the likelihood that the biometric security system will incorrectly reject an access attempt by an unauthorized user.

$$FRR = \frac{\text{Number of false rejection}}{\text{Number of identification attempts}}$$

6.1.4 Recognition Rate (RR)

The recognition rate, or RR, is the measure that the biometric security system will correctly accept an access attempt by an authorized user.

$$RR = \frac{\text{Number of correct recognitions}}{\text{Number of identification attempts}}$$

6.2 Experimentation

6.2.1 Database

Our database contains images in JPEG format with resolution 150 X 112. The threshold value for the recognition is set to 0.5 for the experimentation purpose.

6.2.2 Experiment 1

Face Detection results

Figure 6.1 - Input test image (4 face)

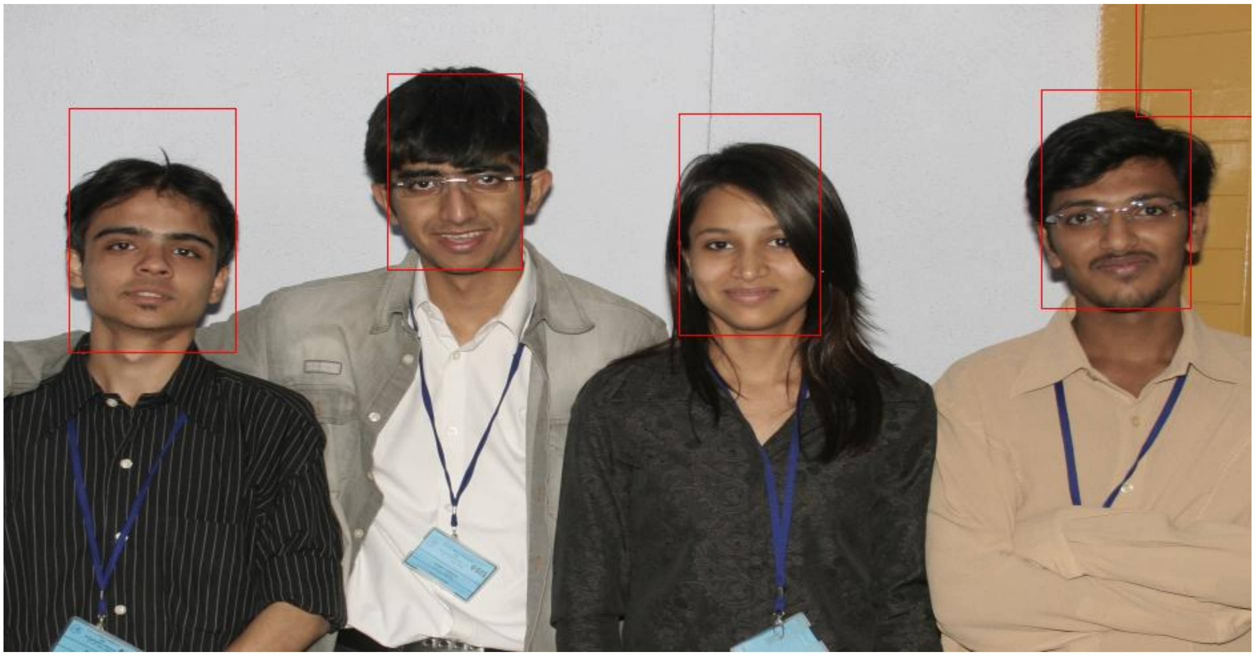


Figure 6.2 - Image with detected faces

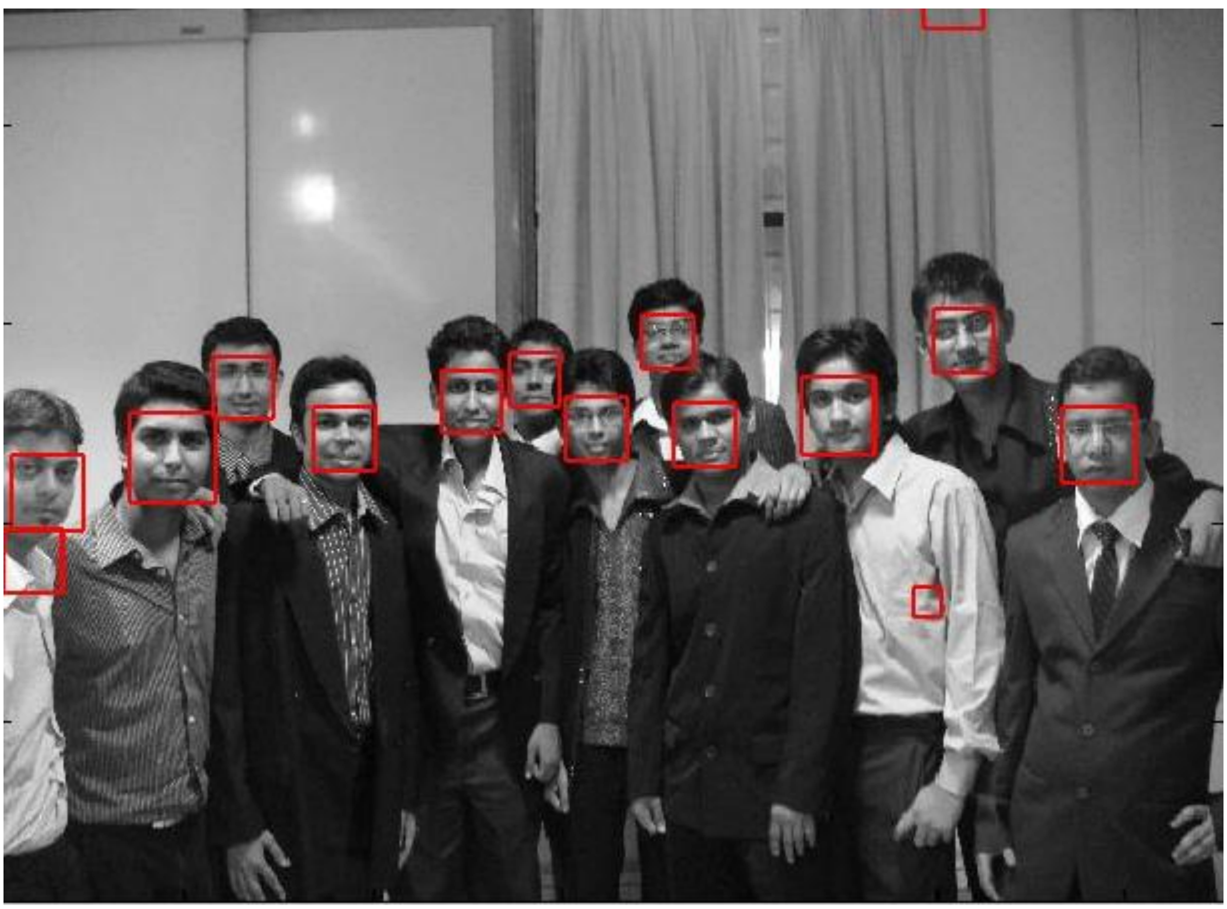


Figure 6.3 - Group Test Image

In face detection, the test images include images with nonface and no. of faces increasing in ascending order. The test images consist of brightness variation, various backgrounds and texture. The results of detection on these test images are given below in tabular form.

No of faces in the image	No of detected faces	No of false detections	Correct detection (%)	False detection(%)
1	1	0	100	0
2	2	1	100	33.33
3	3	2	100	40
4	4	1	100	20
5	5	4	100	68.75
6	6	6	100	50
7	7	4	100	36.36
8	8	6	100	42.85
9	9	8	100	50.1
10	10	2	100	5.36

Figure 6.4: Face Detection for rotation about x axis

Figure 6.5: Face Detection for rotation about z axis

6.3 Face Detection Difficulties

It is really easy to do the face detection with our human visual system. But automatic face detection systems developed so far has not produced good results. In fact, the object "face" is hard to define because of its large variability, depending on the identity of the person, the lightning conditions, the psychological context of the person etc. The main challenge for detecting faces is to find a classifier which can discriminate faces from all other possible images. The first problem is to find a model which can englobe all the possible states of faces.
Lets define the main variable points of the faces:

- **The face global attributes:** A face is globally an object which can be estimated by a kind of ellipse but there are thin faces, rounder faces... The skin color can also be really different from one person to one another.
- **The pose of the face:** The images of a face vary due to the relative camera-face pose (frontal, 45 degree, profile, upside down).
- **The facial expression:** The appearance of faces are directly affected by a person's facial expression.
- **Presence of added objects:** Facial features such as beards, mustaches, and glasses may or may not be present and there is a great deal of variability among these components including shape, color, and size. The glasses can change one of the main characteristics of the faces: the darkness of the eyes. Natural facial features such as mustache, beards or hair can occult one part of the face.
- **Image Condition:** When the image is formed, factors such as lighting (spectra, source distribution and intensity) and camera characteristics (sensor response, lenses) affect the appearance of a face.

The background composition is one of the main factors for explaining the difficulties of face detection. Even if it is quite easy to build systems which can detect faces on uniform backgrounds, most of the applications need to detect faces in any background condition, meaning that the background can be textured and with a great variability. So the two-class classification task is to assign an image to the face class or the Non-face class. Given a set of examples, We can extract some properties of faces for representing the face class but it is impossible to find properties which can represent all the non face class.

Figure 6.6 - False Detection

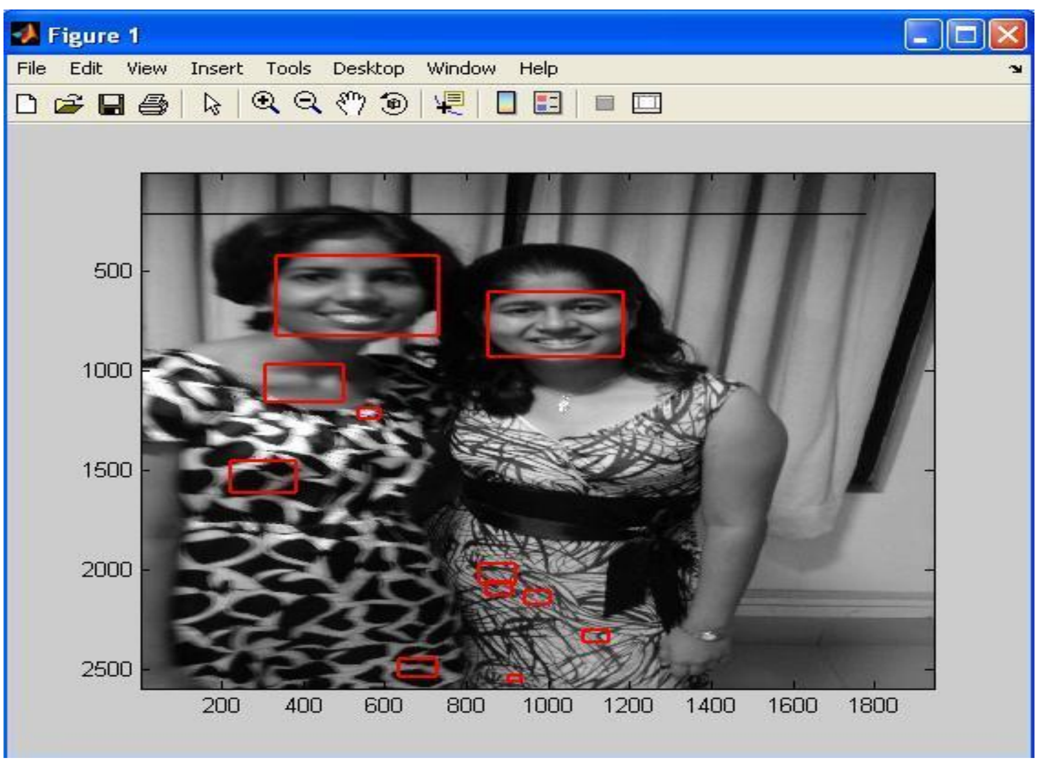


Figure 6.6 - False Detection

6.3.1 *Experiment 3*

Face recognition results:

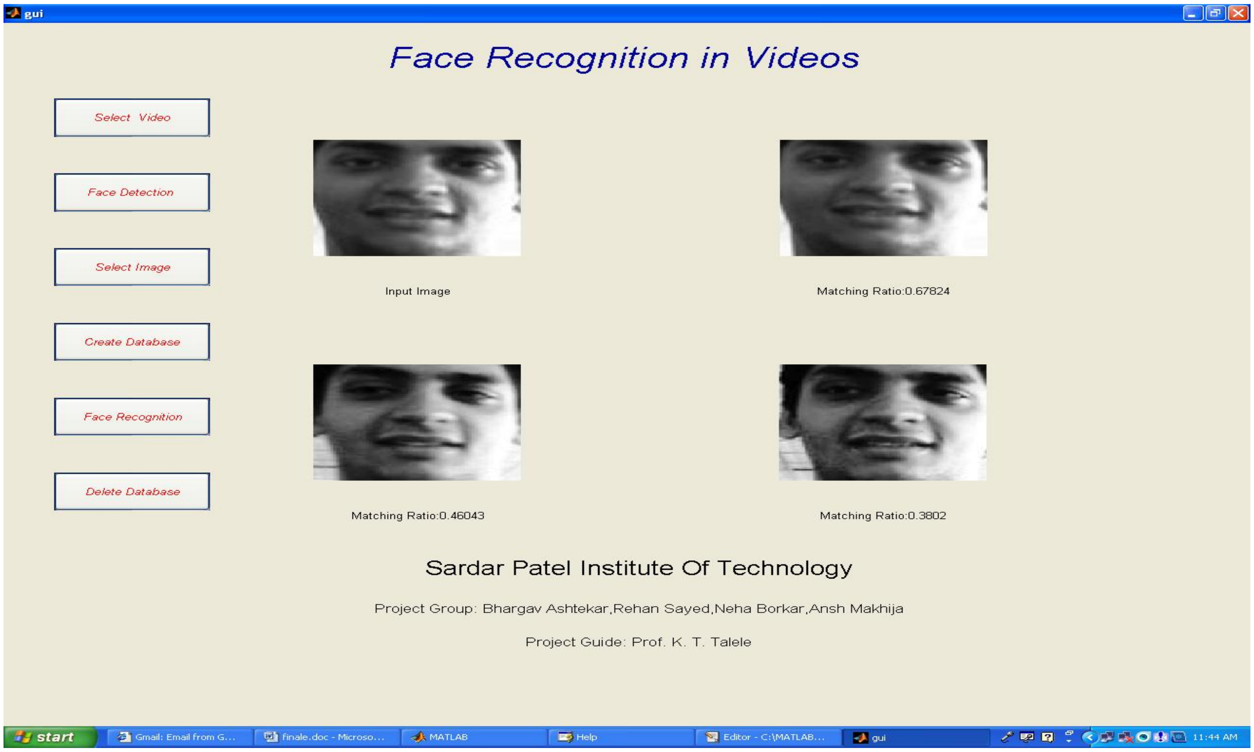
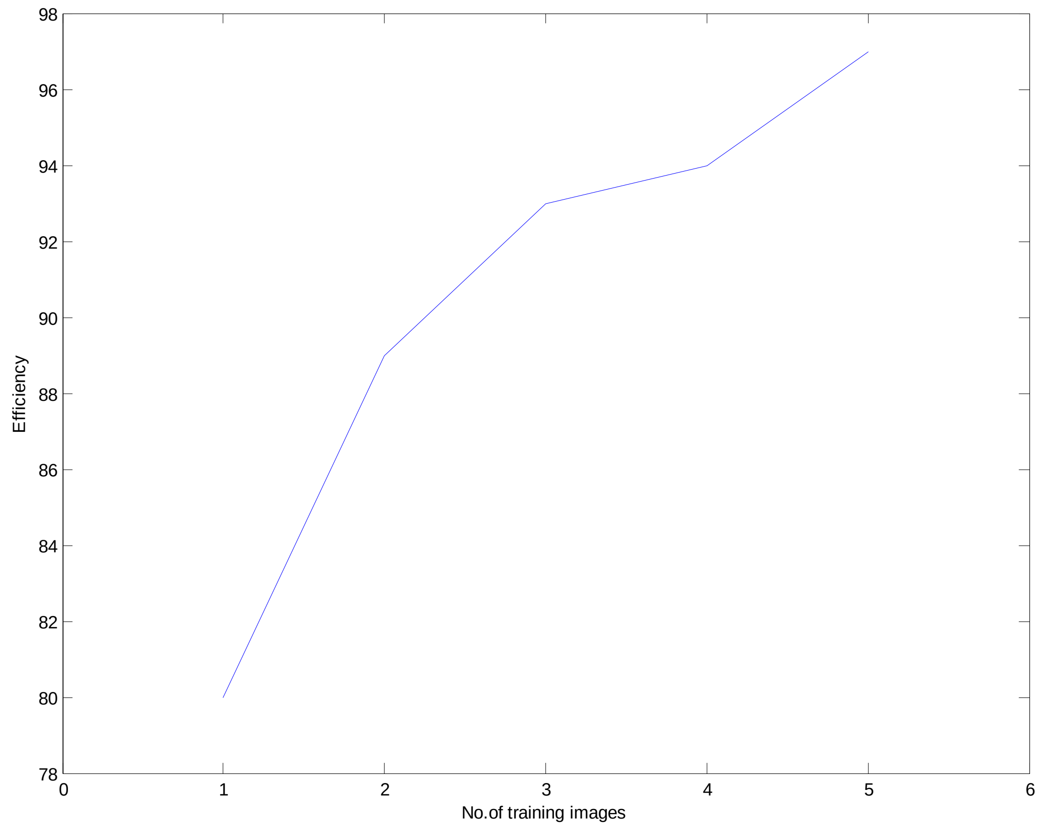


Figure 6.7 – Face Recognition Results I

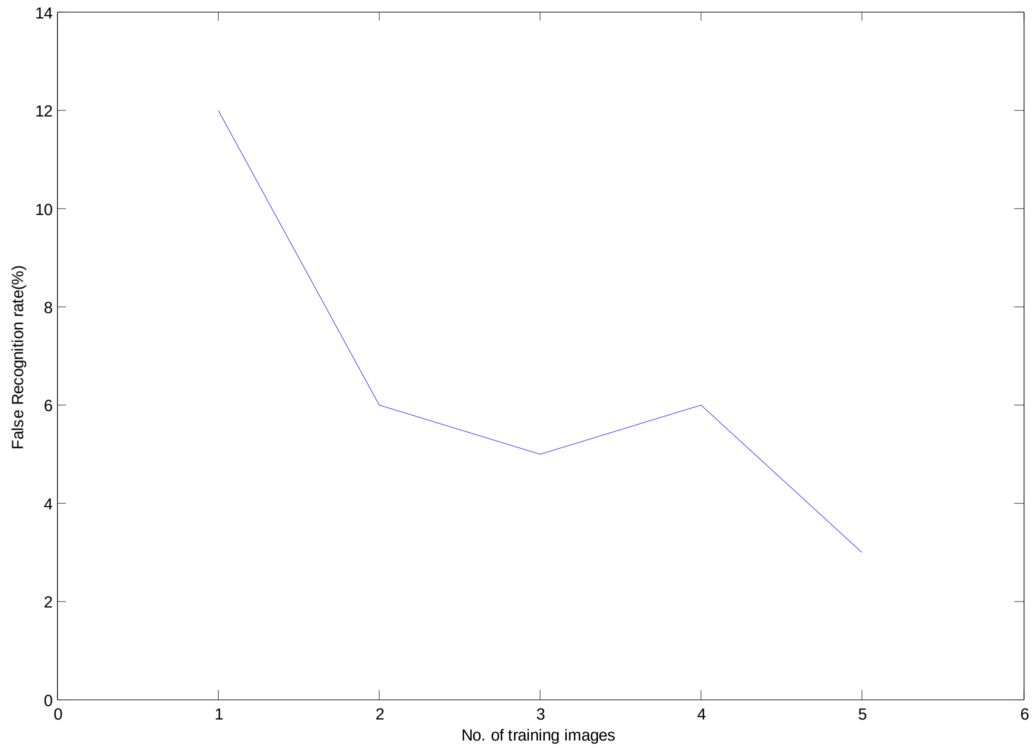
As we perform the above experiments, we observe that face recognition rate has considerably improved at 256 bits per pixel level. The false acceptance and false recognition rate has also declined favourably. We have increased the size of the database in steps by adding one image each, then two images each and finally all the three images of 50 individuals to the database. Although the increase in the database size improves the efficiency of the experiment to a 100% face recognition rate, we need to compromise with the additional storage area and the increase in the processing speed for comparison of testing image with the training images.

The results for the negative images at 256 pixel levels however are better as compared to positive images at the same pixel level. So we conclude that images must be pre-processed to bring them at negative 256 bits per pixel level and then be used for performing human face recognition.

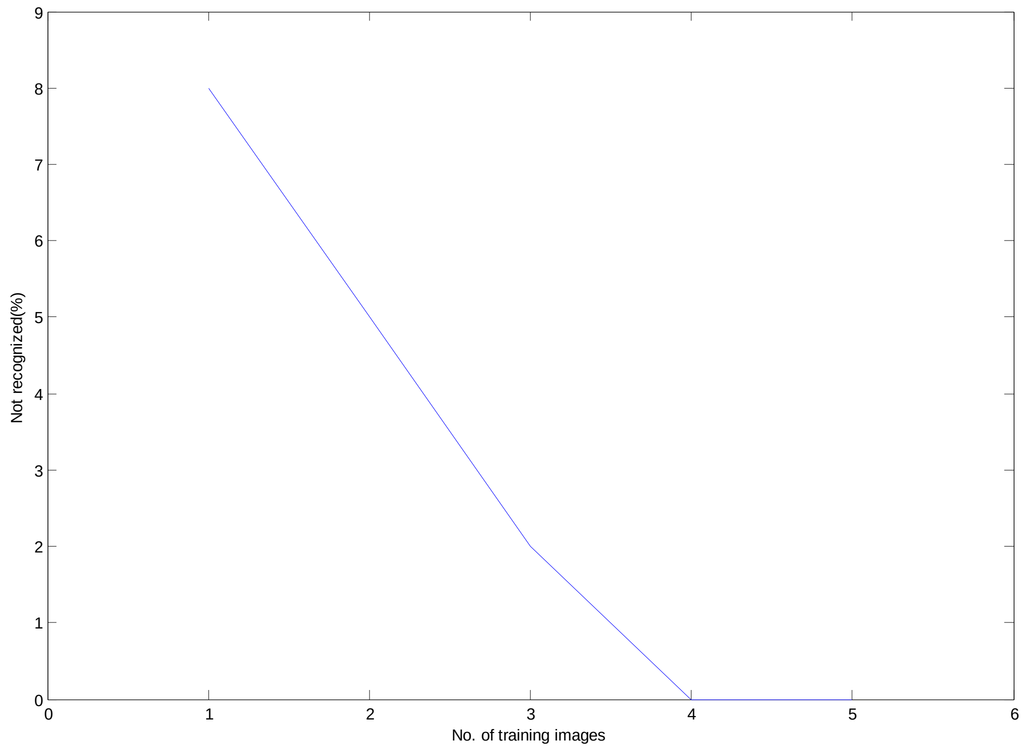
6.4 Results for normalized images



No. of training images	Recognition rate (%)
1	80
2	89
3	93
4	94
5	97



No. of training images	False Recognition rate(%)
1	12
2	6
3	5
4	6
5	3



No. of training images	Not Recognized in %
1	8
2	5
3	2
4	0
5	0

Figure 6.8: Graphs of different parameters of face recognition

From the graphs displayed above, it is observed that as the no. of training images increases the recognition efficiency also increases. We can also observe that, as compared to original images the recognition efficiency increases for every case in case of normalized images. Correspondingly, false recognition rate and not recognized rate decreases in both the cases.

CHAPTER 7

USER MANUAL

7.1 Face Recognition Manual

7.1.1 To start the process of Face Recognition on images

- Install MATLAB on your PC and then run MATLAB program
- Open the gui.m in the folder 'facereco jpeg'
- Run the code
- This will show the 'Face Recognition' GUI on your screen

7.1.2 To select the video to be analyzed

Figure 7.1: To select the video

7.1.3 To perform Face Detection

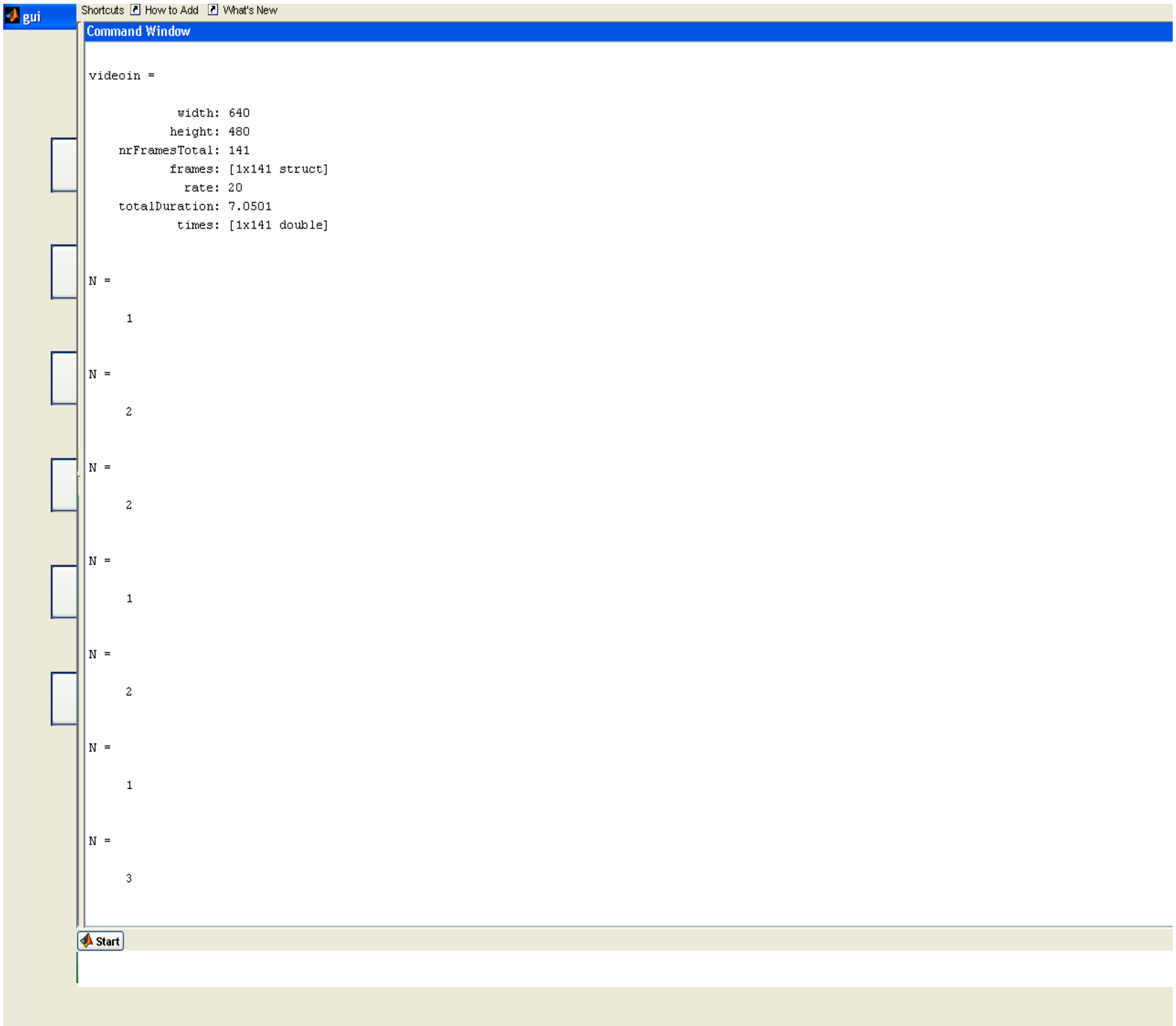


Figure 7.2: After performing face detection

7.1.4 To Load the Image as a New record

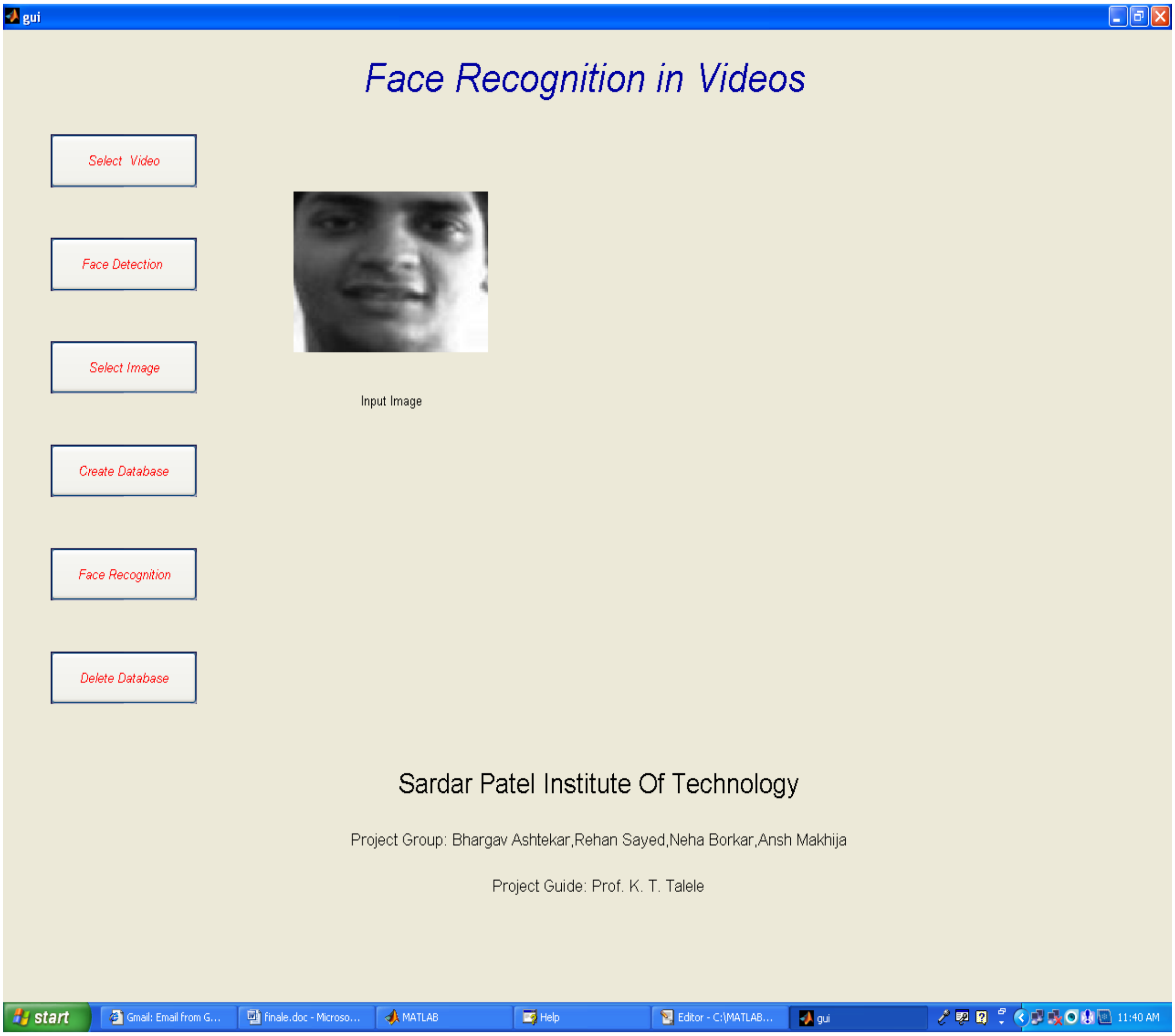


Figure 7.3: To select the image from detected faces

7.1.5 Display Result

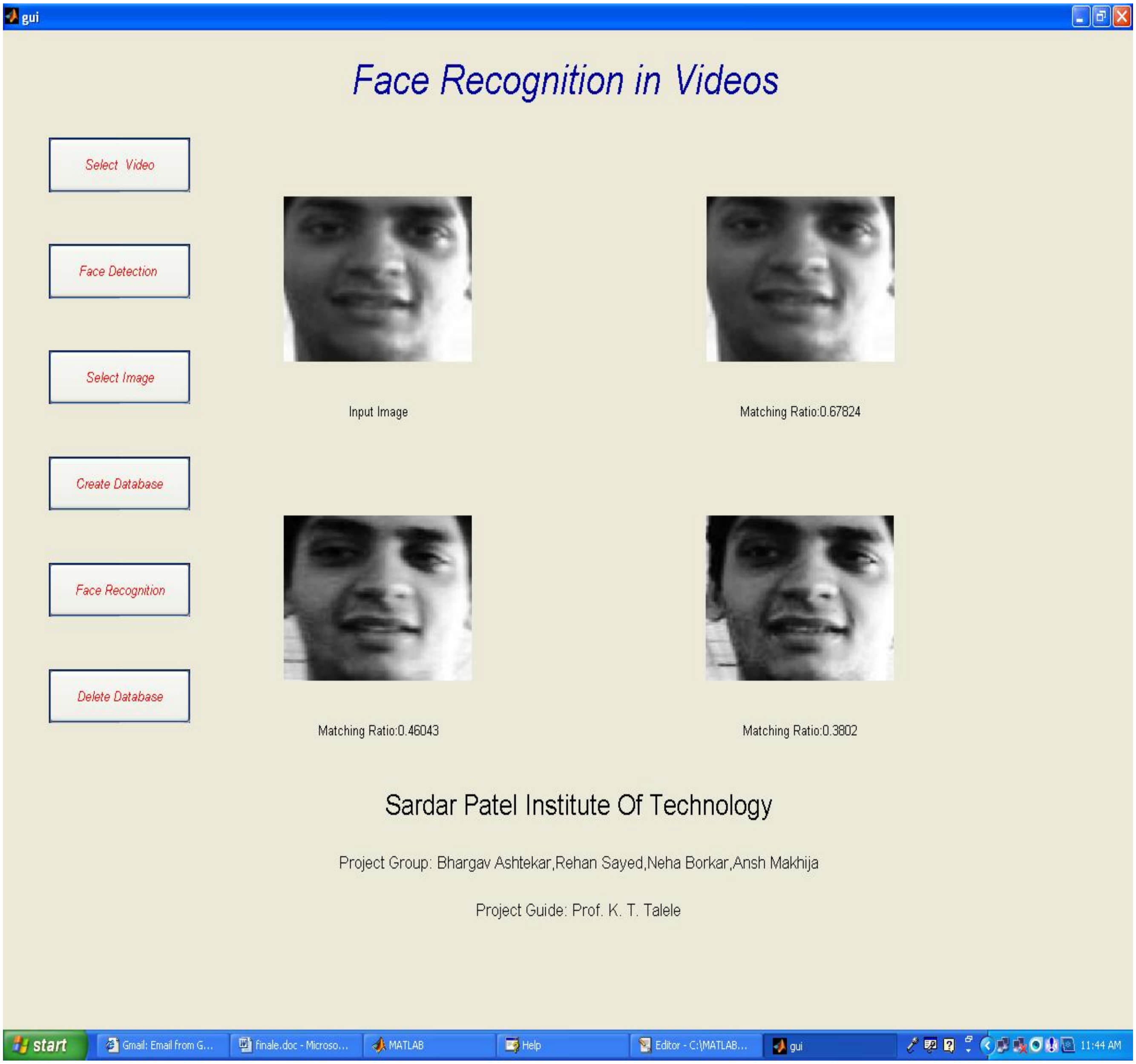


Figure 7.4: After performing face recognition

CHAPTER 8

APPLICATIONS

1. Human computer interaction:

One could make computers easier to use if when one simply sat down at a computer terminal, the computer could identify the user by name and automatically load personal preferences. This identification could even be useful in enhancing other technologies such as speech recognition, since if the computer can identify the individual who is speaking, the voice patterns being observed can be more accurately classified against the known individual's voice.

2. Biometrics:

It uses pattern recognition techniques to identify people using their unique characteristics. Some of these methods are fingerprints and retina and iris recognition.But these are obtrusive and expensive.

3. Security:

There are many different identification technologies available today to uniquely identify a person's identity, many of which have been in commercial use for years. Password/PIN known as Personal Identification Number systems are the most common in practice today. However these systems have their own intrinsic drawbacks. Passwords can be forgotten and worse if they are lost or stolen, person identity can be misused by somebody else. Chances of unauthorised access to critical information have become a major concern. So a system must be implemented so that the security system becomes more robust and probability of unauthorized access to the system becomes minimum.

- Facial Recognition is unobtrusive and discrete. The infrastructure for its implementation is already widespread and inexpensive.
- At present the demand for Facial Recognition technologies are fueled by the needs of Homeland Security.
- Applications for Facial Recognition are varied and vast:
    - ✓ Document Control e.g. **digital chip in passports, drivers' licenses**
    - ✓ Transactional authentication e**.g. credit cards, ATMs, point-of-sale**
    - ✓ Computer security e.g. **user access verification**
    - ✓ Physical access control e.g. **smart doors**
    - ✓ Logical access control
    - ✓ Voter registration e.g. **election accuracy**
    - ✓ Time and Attendance e.g. **entry and exit verification**
    - ✓ Computer games e.g. **a virtual "you" plays against virtual opponents**
    - ✓ Border Controls/Airports

With improvements in reliability, future applications will come increasingly from the commercial sector.

CHAPTER 9

CONCLUSION

The aim of the proposed project was to develop an efficient human face recognition system in videos that would provide a high probability of appropriate face recognition and a minimum probability for false acceptance and false rejection. The human face recognition system developed has been successful in achieving its set objectives. The efficiency of the project is subject to the size of database.

By the realization of the proposed system we have learnt many aspects of human face recognition system.

The project can be revised further in terms of additional functionalities and features that can be appended to the developed system in the future.

CHAPTER 10

LIMITATIONS

Although the idea of the whole process sounds convincing, there are certain limitations in implementing the process which are stated as follows:

- Images with different profiles cannot be recognised by the system
- If the number of colour levels in an image is low, the system does not create a perfect contour of the image
- The system works only for images with the .jpg extension
- The system will work only for black and white images
- There is a compromise between the detection rate and no. of false detections

Another notable point is that as the size of the database increases, the required processing power along with memory requirement also increases.

CHAPTER 11

FUTURE SCOPE

Image-based face recognition in videos is still a very challenging topic after decades of exploration.

Face recognition systems used today work very well under constrained conditions, although all systems work much better with frontal images and constant lighting. All current face recognition algorithms fail under the vastly varying conditions.

Sensitivity to variations in pose is still a challenging problem. The face of an individual rotates in depth, including up/down rotation and left/right rotation. In essence, the difference between the same people under the varied poses is larger than the difference between the distinct persons under the same pose. So it is difficult for the computer to do the face identification when the poses of the probe and gallery images are different. Adding the images at different poses increases the database size and thus it takes a lot of computation time for recognition.

Also face recognition using coloured images is a topic to be explored. The biggest challenge is locating an individual face from a video and pre-processing the image for face recognition. The difficulties that need to be tackled for implementing face recognition in video are video-base segmentation, face tracking, video-based feature extraction, low quality compressed images depending on the application.

The scope for future development of our project is enormous. The field of face recognition incorporating coloured images is still new field with huge scope of exploration.

2D face recognition gives best result when image is frontal and effect of illumination is very less. Tilted image with illumination effect are not suited best in 2 dimensional recognition. Depth information is one of the most important parameter which mostly concern about eye, nose etc detection and recognition. We can not realize depth information in 2D view as it concern only about X and Y axis. Information about Z axis i.e. depth information is simply ignored. 3D face recognition is a challenging area of exploration. In 2D face recognition, recognition rate is less as compared to 3D face recognition. 3 dimensional views give us depth information with best recognition rate. Also we can normalize tilted image for its best result.

The project primarily aims to build face recognition-based authentication systems that are more intuitive and less intrusive and has been increasing greatly along with the growing demands for security and personal safety.

Your face can now replace a key or password. Thus the scope of video surveillance is enhanced. Due to variations in facial features due to ageing and wrinkles and the orientation of a person , this project has its own limitation. Sometimes faces which should be detected go undetected because of geometric orientation.

CHAPTER 12

REFERENCES

The literature review conducted for this project primarily includes review of published papers dealing with Face Detection and Recognition. The objective was to gain an insight into the theory of the concepts that would be used in further work of this project.

[1] Mikael Nilsson, J¨orgen Nordberg, and Ingvar Claesson, 'Face Detection using local SMQT features and split up SNOW classifier' 1st International Workshop on Artificial Life and Robotics

[2] Muhammad Firdaus Hashim, Puteh Saad, Mohd Rizon , Mohamed Juhari and Shahrul Nizam Yaakob 'A Face Recognition System Using Template Matching And Neural Network Classifier' Blekinge Institute of Technology School of Engineering

[3] Christophe Garcia,Giorgos Zikos, Giorgos Tziritas 'A Wavelet based framework for Face recognition' ICS – Foundation for Research and Technology-Hellas – FORTH

[4] 'Face Recognition Using Ensembles of Networks' published by *S. Gutta+, J. Huang+, B. Takacs*, and H. Wechsler*, International Conference on Pattern Recognition (ICPR), 1996, Vienna, Austria

[5] 'Image Analysis for Face Recognition' published by *Xiaoguang Lu*, Dept. of Computer Science & Engineering Michigan State University

[6] E. Saber and A.M. Tekalp, 'Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions,' Pattern Recognition Letters, 19(8),pages 669-680, 1998.

[7] M.F. Augusteijn and T.L. Skujca, 'Identification of Human Faces through Texture-Based Feature Recognition and Neural Network Technology,' Proc. IEEE Conf. Neural Networks, pp. 392-398, 1993.

[8] M. Turk and A. Pentland, 'Eigenfaces for Recognition,' J. Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[9] S.H. Lin, S.Y. Kung, and L.J. Lin, 'Face Recognition by Probabilistic Decision-Based Neural Network,' IEEE Trans. Neural Networks, vol. 8, no. 1, pp. 114-132, 1997.

[10] S.T.Gandhe, K.T.Talele, A.G.Keskar, 'Face Recognition Using Contour Matching' IAENG International Journal of Computer Science.

[16] Paul Viola and Micheal Jones, 'Robust real-time object detection,' Second International Workshop on Statistical Learning and Computational Theories of Vision Modeling, Learning, Computing and Sampling, July 2001.

[17] Anil Kumar Sao and B. Yegnanaarayana 'Template matching Approach for Pose Problem in Face Verification' Speech and Vision Laboratory, Department of Computer Science and Engineering, Indian Institute of Technology Madras

[18] Wei Fan, Yunhong Wang, and Tieniu Tan, 'Video-based Face Recognition Using Bayesian Inference Model' Video-based Face Recognition Using Bayesian Inference Model' published by National Laboratory of Pattern Recognition, China

[11] CMU Dataset: http://vasc.ri.cmu.edu/idb/html/face/frontal images/
http://vasc.ri.cmu.edu/idb/images/face/frontal images/images.tar

[12] Bao Face Database: www.facedetection.com
http://www.facedetection.com/facedetection/datasets.htm

[13] ORL Database:
http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html
http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att faces.zip

[14] Y. Moses, Y. Adini and S. Ullman, 'Face Recognition: the problem of the compensating for changing in illumination direction,' In Europeean conference on Computer Vision, pages 286-296, 1994.

[15] M.H. Yang, D. Roth, and N. Ahuja, 'A SNoW-Based Face Detector,' Advances in Neural Information Processing Systems 12, S.A. Solla, T. K. Leen, and K.-R. Muller, eds., pp. 855-861, MIT Press, 2000.

APPENDIX
ANNEXES

Implementation

The Software and hardware used to develop the project is:

Software Platform

- Compatible Operating Systems
  - ✓ Windows XP Professional
  - ✓ Windows 2000 Professional
  - ✓ Windows NT/ME
  - ✓ Windows 98
- Versions of MATLAB
  - ✓ MATLAB version 7.0 or 8.0 installed .

Hardware Platform

- Configuration
  - ✓ 2.2 GHz Core 2 duo Intel processor
  - ✓ 1 GB of RAM