Chapter 1

1. INTRODUCTION

Within today's environment of increased importance of security, identification and authentication methods have developed into a key technology in various areas; entrance control in buildings; access control for computers in general or for automatic teller machines in particular; day-to-day affairs like withdrawing money from a bank account or dealing with the post office; or in the prominent field of criminal investigation. Such requirement for reliable personal identification in computerized control has resulted in an increased interest in biometrics.

In today's world security is main issue of every individual and even nation. Increase in security demand has added a new cost and factor for every project, but with high cost of sector equipments and services, reach of every man to opt for it, is still difficult.

Technology has always been appreciated when it's for mankind. Following the same path many biometric equipments already exist in the market like Fingerprint, Hand geometry, WS, Retina, Face Recognition, Voice Signature etc.

Biometric identification is the technique of automatically identifying or verifying an Individual by a physical characteristic or personal trait. Biometric characteristics and traits arc divided into behavioral or physical categories. Behavioral biometrics encompasses such behaviors as signature and typing rhythms. Physical biometric systems use the eye, finger, hand, voice and face for identification.

A facial recognition system is a computer application for automatically identifying or verifying a portion from a digital image or a video frame from a video source.

A face recognition system would allow user to be identified by simply walking past a surveillance camera Human beings often recognize one another by unique facial characteristics. One of the newest biometric technologies, automatic facial recognition, is based on this phenomenon. Facial recognition is the most successful form of human surveillance. It is being used to improve human efficiency when recognizing faces, is one of the fastest growing field in the biometric industry. Interest in facial recognition is being fueled by the availability and low cost of video hardware, the ever increasing numbers of video cameras being placed in the workspace, and the non invasive aspect of facial recognition systems and systems already exist in the markets which claim to provide accurate, effective and fast human face recognition, but following limitations exists:

- Have some error when conditions like intensity of light, angle of face changes
- Most of Them lack accuracy with different gestures.

1.1 3d Technique

A newly emerging trend claimed to achieve accuracy. This technique uses 3D sensors to capture information of a face and this information is then used to identify features.

1.2 Skin Texture Analysis Technique

Another emerging trend uses the visual details of the skin, as captured in standard digital or scanned images. This technique, called skin texture analysis, tolls the unique lines, patterns, and spots apparent in a person's skin into a mathematical space. Tests have shown that with the addition of skin texture analysis, performance in recognizing faces can inch ease 20 to 25 percent.

There is a tremendous need of Face recognition systems in today's scenario in which there are various security problems which are hard to be handled manually like:

- Surveillance
- Site Access.
- Criminal investigations
- House Hold security
- Prevent fraud votes
- Retail sector to eliminate cards
- At ATM for user authentication
- Finding, person on networking sites.

1.3 Advantages

Face region has number of advantages. Firstly, it is non intrusive. Whereas many techniques require the subject's cooperation and awareness in order to perform identification or verifications such as looking into eye scanner or placing their hand on fingerprint reader . FRVS could be wed without subject's notice. A system based on face recognition could be more safe, cheap and easy to use.

1.4 Problem Definition

A general statement of problem can be formulated as follows: given still or video images of Scene, identify one or more persons in the scene using stored database of faces and display their information.

The environment surrounding a face recognition application can cover a wide spectrum from a well controlled environment to an uncontrolled one.

Recognition of faces from an uncontrolled environment is a very complex, lightning condition may vary tremendously, facial expressions also vary from time to time appear at different orientations and a face can be partially occluded. Further, depending on the application, handling facial features over time (aging) may also be required.

Although existing methods performs well under certain conditions, the problems with the illumination changes, out of plane rotations and occlusions are still remain unsolved. The proposed algorithm, deals with two of these three important problems, namely occlusion and illumination changes.

1.5 Related Work

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past several years. At least two reasons account for this trend; the first is the wide range of commercial and law enforcement applications, and the second is the availability of feasible technologies after 30 years of research. Even though current machine recognition systems have reached a certain level of maturity, their success is limited by the conditions imposed by many real applications. For example, recognition of face images acquired in an outdoor environment with changes in illumination and/or pose remains a largely unsolved problem. In other words, current systems are still far away from the capability of the human perception system.

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past few years. The strong need for user-friendly systems that can secure our assets and protect our privacy without losing our identity in a sea of numbers is obvious. At present, one needs a PIN to get cash from an ATM, a password for a computer, a dozen others to access the internet, and so on. Although very reliable methods of biometric personal identification exist, e.g., fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant's cooperation or knowledge. Some of the advantages/disadvantages of different biometrics are described in Table lists some of the applications of face recognition. Commercial and law enforcement applications of FRT range from static, controlled-format photographs to uncontrolled video images, posing a wide range of technical challenges and requiring an equally wide range of techniques from image processing, analysis, understanding and pattern recognition.

A general statement of the problem of machine recognition of faces can be formulated as follows: Given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces. Available collateral information such as race, age, gender, facial expression, or speech may be used in narrowing the search (enhancing recognition). The solution to the problem involves segmentation of faces (face detection) from cluttered scenes, feature extraction from the face regions, recognition, or verification .

In identification problems, the input to the system is an unknown face, and the system reports back the determined identity from a database of known individuals, whereas in verification problems, the system needs to confirm or reject the claimed identity of the input face.

Face perception is an important part of the capability of human perception system and is a routine task for humans, while building a similar computer system is still an on-going research area. The earliest work on face recognition can be traced back at least to the 1950's in psychology and to the 1960's in the

engineering literature. (Some of the earliest studies include work on facial expression of emotions by. But research on automatic machine recognition of faces really started in the 1970's after the seminal work of Kanade and Kelly . Over the past thirty years extensive research has been conducted by psychophysicists, neuroscientists, and engineers on various aspects of face recognition by humans and machines.

Over the past 15 years, research has focused on how to make face recognition systems fully automatic by tackling problems such as localization of a face in a given image or video clip and extraction of features such as eyes, mouth, etc. Meanwhile, significant advances have been made in the design of classifiers for successful face recognition. Among appearance-based holistic approaches, Eigenfaces and Fisherfaces have proved to be effective in experiments with large databases. Feature-based graph matching approaches have also been quite successful. Compared to holistic approaches, feature-based methods are less sensitive to variations in illumination and viewpoint and to inaccuracy in face localization. However, the feature extraction techniques needed for this type of approach are still not reliable or accurate enough. For example, most eye localization techniques assume some geometric and textural models and do not work if the eye is closed.

During the past five to eight years, much research has been concentrated on video-based face recognition. The still image problem has several inherent advantages and disadvantages. For applications such as drivers' licenses, due to the controlled nature of the image acquisition process, the segmentation problem is rather easy. However, if only a static picture of an airport scene is available, automatic location and segmentation of a face could pose serious challenges to any segmentation algorithm. On the other hand, if a video sequence is available, segmentation of a moving person can be more easily accomplished using motion as a cue. But the small size and low image quality of faces captured from video can significantly increase the difficulty in recognition.

We had given below a concise summary, followed by conclusions:

Machine recognition of faces has emerged as an active research area spanning disciplines such as image processing, pattern recognition, computer vision, and neural networks. There are numerous applications of FRT to commercial systems such as face verification based ATM and access control, as well as law enforcement applications to video surveillance, etc. Due to its user-friendly nature, face recognition will remain a powerful tool in spite of the existence of very reliable methods of biometric personal identification such as fingerprint analysis and iris scans.

- Extensive research in psychophysics and the neurosciences on human recognition of faces is documented in the literature. I do not feel that machine recognition of faces should strictly follow what is known about human recognition of faces, but it is beneficial for engineers who design face recognition systems to be aware of the relevant findings. On the other hand, machine systems provide tools for conducting studies in psychology and neuroscience.

- Numerous methods have been proposed for face recognition based on image intensities. Many of these methods have been successfully applied to the task of face recognition, but they have advantages and disadvantages. The choice of a method should be based on the specific requirements of a given task. For example, the EBGM-based method has very good performance, but it requires an image size, e.g., which severely restricts its possible application to video-based surveillance where the image size of the face area is very small. On the other hand, the subspace LDA method works well for both large and small images.

- Recognition of faces from a video sequence (especially, a surveillance video) is still one of the most challenging problems in face recognition because video is of low quality and the images are small. Often, the subjects of interest are not cooperative, e.g., not looking into the camera. One particular difficulty in these applications is how to obtain good-quality gallery images. Nevertheless, video-based face recognition systems using multiple cues have demonstrated good results in relatively controlled environments.

1.6 Overview Of Major Face Recognition Technologies

1.6.1 Eigenface Technique

It utilizes two dimensional global grayscale images representing distinctive characteristics of human face describing what is common to groups of individuals and where they differ most. Just as any color can be created by mixing using primary colors(RGB),vast majority of faces can be reconstructed by combining features of approx 100-150 eigenfaces. The performance depends on number of eigenvectors. If it is too small, important information about identity may be lost. If too high, weight corresponding to small eigen values may be noisy.

1.6.2 Limitations

i. One of the assumptions in eigenface is that variability in the underlying images corresponds to differences between individual faces. This assumption is, unfortunately, not always valid.

ii. Other factors that may "stretch" image variability in directions that tend to blur identity in PCA space include changes in expression, camera angle, and head pose. Since the eigenvectors are determined only by data variability, you're limited in what you can do to control how eigenface behaves. However, you can take steps to limit,

or to otherwise manage, environmental conditions that might confuse it. For example, placing the camera at face level will reduce variability in camera angle.

iii. Lighting conditions, such as side lighting from windows, are harder for a mobile robot to control. But you might consider adding intelligence on top of face recognition to compensate for that. For example, if your robot knows roughly where it's located, and which direction it's facing, it can compare the current face image only to ones it's seen previously in a similar situation.

1.6.3 Neural Network Technology

Extracts features from the entire face as visual contrast element, quantifying, normalizing and compressing them into template code. After that, it uses a specialized algorithm to determine that parts one by one to find the matching image or features in the database.

1.6.4 Dynamic Link Architecture Or Graph Matching

It is an extension to classify artificial neural networks. Memorized objects are represented by sparse graphs, whose vertices are labeled with a multidimensional description in terms of local power spectrum and whose edges are labeled with geometric distance vectors that can be described by vector representation the position and size of facial features such as eye, eyebrows, nose, mouth.

1.6.5 Automatic Face Processing

It is the most rudimentary and easy to understand. It simply uses distance ratios between key facial features.

1.6.6 Fisherfaces/Subspace LDA

Fisher's Linear Discriminant and produces well separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions. The Eigenface technique, another method based on linearly projecting the image space to a low dimensional subspace, has similar computational requirements. Yet, extensive experimental results demonstrate that the proposed "Fisherface" method has error rates that are lower than those of the Eigenface technique for tests on the Harvard and Yale Face Databases.

1.6.7 Innovativeness & Usefulness

Existing algorithms can be enhanced alongwith development of new algorithms.
FRVS will assist security personnel's in keeping track of all the people entering and leaving a secured zone. The core functionality of FRVS will allow real time tracking,recognition and retrieval of stored relevant information of human faces in a video stream. Today this is based mainly on magnetic cards. A system based on face recognition could be more safer and cheaper.

Main areas in which it could be used are:

- Surveillance.
- Site Access.
- Criminal Investigation.
- House-Hold security. Etc.

The other used techniques are: Fingerprints, Hand geometry, IRIS, Retina, Voice, Signature.

Advantages of FRVS over other techniques:
Face recognition has number of advantages. Firstly, it is non intrusive. Whereas many techniques require the subject's cooperation and awareness in order to perform, an identification or verification, such as looking into eye scanner or placing their hand on fingerprint reader, FRVS could be used without subject's notice.

1.6.8 Limitations

Due to dynamic nature of face a face recognition system encounters various problems during the recognition process. A robust face recognition system should have:

i. Scale invariance: The same face can be presented to system at different perspective and orientations. Eg. Frontal, profile views. Besides head orientation may change due to translation and rotation.
ii. Illumination invariance: Face images of same person can be taken under different illumination condition, as position and strength of light source can be modified.
iii. Noise invariance: The robust face recognition system should be insensitive to noise generated by camera or frame grabbers.
iv. Emotional Expression: Face images of same person can differ on expressions when smiling or laughing. Also, beards, moustaches can be present,

Chapter 2

2. FACE DETECTION

- **2.1 Definition**
- **2.2 Introduction**
- **2.3 Operation of a Face Detection System**
- **2.4 Recent Advances**
- **2.5 Quantifying Performance**
- **2.6 Applications**

2. FACE DETECTION

2.1 Definition

Face detection is concerned with finding whether or not there are any faces in a given image (usually in gray scale) and, if present, return the image location and content of each face. This is the first step of any fully automatic system that analyzes the information contained in faces (e.g., identity, gender, expression, age, race and pose). While earlier work dealt mainly with upright frontal faces, several systems have been developed that are able to detect faces fairly accurately with in-plane or out-of-plane rotations in real time. Although a face detection module is typically designed to deal with single images, its performance can be further improved if video stream is available.

2.2 Introduction

The advances of computing technology has facilitated the development of real-time vision modules that interact with humans in recent years. Examples abound, particularly in biometrics and human computer interaction as the information contained in faces needs to be analyzed for systems to react accordingly. For biometric systems that use faces as non-intrusive input modules, it is imperative to locate faces in a scene before any recognition algorithm can be applied. An intelligent vision based user interface should be able to tell the attention focus of the user (i.e., where the user is looking at) in order to respond accordingly. To detect facial features accurately for applications such as digital cosmetics, faces need to be located and registered first to facilitate further processing. It is evident that face detection plays an important and critical role for the success of any face processing systems. The face detection problem is challenging as it needs to account for all possible appearance variation caused by change in illumination, facial features, occlusions, etc. In addition, it has to detect faces that appear at different scale, pose, with inplane rotations. In spite of all these difficulties, tremendous progress has been made in the last decade and many systems have shown impressive real-time performance. The recent advances of these algorithms have also made significant contributions in detecting other objects such as humans/pedestrians, and cars.

2.3 Operation of a Face Detection System

Most detection systems carry out the task by extracting certain properties (e.g., local features or holistic intensity patterns) of a set of training images acquired at a fixed pose (e.g., upright frontal pose) in an off-line setting. To reduce the effects of illumination change, these images are processed with histogram equalization or standardization (i.e., zero mean unit variance). Based on the extracted properties, these systems typically scan through the entire image at every possible location and scale in order to locate faces. The extracted properties can be either manually coded (with human knowledge) or learned from a set of data as adopted in the recent systems that have demonstrated impressive results. In order to detect

faces at different scale, the detection process is usually repeated to a pyramid of images whose resolution are reduced by a certain factor from the original one. Such procedures may be expedited when other visual cues can be accurately incorporated (e.g., color and motion) as pre-processing steps to reduce the search space. As faces are often detected across scale, the raw detected faces are usually further processed to combine overlapped results and remove false positives with heuristics (e.g., faces typically do not overlap in images) or further processing (e.g., edge detection and intensity variance). Numerous representations have been proposed for face detection, including pixel-based, parts-based, local edge features, Haar wavelets and Haar-like features. While earlier holistic representation schemes are able to detect faces, the recent systems with Haar-like features have demonstrated impressive empirical results in detecting faces under occlusion. A large and representative training set of face images is essential for the success of learning-based face detectors. From the set of collected data, more positive examples can be synthetically generated by perturbing, mirroring, rotating and scaling the original face images. On the other hand, it is relatively easier to collect negative examples by randomly sampling images without face images. As face detection can be mainly formulated as a pattern recognition problem, numerous algorithms have been proposed to learn their generic templates (e.g., eigenface and statistical distribution) or Discriminant classifiers (e.g., neural networks, Fisher linear Discriminant, sparse network of Winnows, decision tree, Bayes classifiers, support vector machines, and AdaBoost). Typically, a good face detection system needs to be trained with several iterations. One common method to further improve the system is to bootstrap a trained face detector with test sets, and re-train the system with the false positive as well as negatives. This process is repeated several times in order to further improve the performance of a face detector. A survey on these topics can be found in, and the most recent advances are discussed in the next section.

2.4 Recent Advances

The AdaBoost-based face detector by Viola and Jones demonstrated that faces can be fairly reliably detected in real-time (i.e., more than 15 frames per second on 320 by 240 images with desktop computers) under partial occlusion. While Haar wavelets were used in for representing faces and pedestrians, they proposed the use of Haar-like features which can be computed efficiently with integral images and with four types of Haar-like features that are used to encode the horizontal, vertical and diagonal intensity information of face images at different position and scale. Given a sample image of 24 by 24 pixels, the exhaustive set of parameterized Haar-like features (at different position and scale) is very large (about 160,000). Contrary to most of the prior algorithms that use one single strong classifier (e.g., neural networks and support vector machines), they used an ensemble of weak classifiers where each one is constructed by thresholding of one Haar-like feature. The weak classifiers are selected and weighted using the AdaBoost algorithm. As there are large numbers of weak classifiers, they presented a method to rank these classifiers into several cascades using a set of optimization criteria. Within each stage, an ensemble of several weak classifiers is trained using the AdaBoost algorithm. The motivation behind the cascade of classifier is that simple classifiers at early stage can filter out most negative

examples efficiently, and stronger classifiers at later stage are only necessary to deal with instances that look like faces. The final detector, a 38 layer cascade of classifiers with 6,060 Haar-like features, demonstrated impressive real-time performance with fairly high detection and low false positive rates. Several extensions to detect faces in multiple views with in-plane ration have since been proposed. Despite the excellent run-time performance of boosted cascade classifier, the training time of such a system is rather lengthy. In addition, the classifier cascade is an example of degenerate decision tree with an unbalanced data set (i.e., a small set of positive examples and a huge set of negative ones). Numerous algorithms have been proposed to address these issues and extended to detect faces in multiple views. To handle the asymmetry between the positive and negative data sets, Viola and Jones proposed the asymmetric AdaBoost algorithm which keeps most of the weights on the positive examples.

In, the AdaBoost algorithm is used to select a specified number of weak classifiers with lowest error rates for each cascade and the process is repeated until a set of optimization criteria (i.e., the number of stages, the number of features of each stage, and the detection/false positive rates) is satisfied.

As each weak classifier is made of one single Haar-like feature, the process within each stage can be considered as a feature selection problem. Instead of repeating the feature selection process at each stage, Wu et al. presented a greedy algorithm for determining the set of features for all stages first before training the cascade classifier. With the greedy feature selection algorithm used as a pre-computing procedure, they reported that the training time of the classifier cascade with AdaBoost is reduced by 50 to 100 times.

For learning in each stage (or node) within the classifier cascade, they also exploited the asymmetry between positive and negative data using a linear classifier with the assumptions that they can be modeled with Gaussian distributions. The merits and drawbacks of the proposed linear asymmetric classifier as well as the classic Fisher linear Discriminant were also examined in their work. Recently, Pham and Cham proposed an online algorithm that learns asymmetric boosted classifiers with significant gain in training time. In, an algorithm that aims to automatically determine the number of classifiers and stages for constructing a boosted ensemble was proposed. While a greedy optimization algorithm was employed in, Brubaker et al. proposed an algorithm for determining the number of weak classifiers and training each node classifier of a cascade by selecting operating points within a receiver operator characteristic (ROC) curve.

The solved the optimization problem using linear programs that maximize the detection rates while satisfying the constraints of false positive rates. Although the original four types of Haar-like features are sufficient to encode upright frontal face images, other types of features are essential to represent more complex patterns (e.g., faces in different pose). Most systems take a divide-and-conquer strategy and a face detector is constructed for a fixed pose, thereby covering a wide range of angles (e.g., yaw and pitch angles). A test image is either sent to all detectors for evaluation or to a decision module with a coarse pose estimator for selecting the appropriate trees for further processing.

The ensuing problems are how the types of features are constructed, and how the most important ones from a large feature space are selected. More generalized Haar-like features are defined in which the

rectangular image regions are not necessarily adjacent, and furthermore the number of such rectangular blocks is randomly varied. Several greedy algorithms have been proposed to select features efficiently by exploiting the statistics of features before training boosted cascade classifiers . There are also other fast face detection methods that demonstrate promising results, including the component-based face detector using Naive Bayes classifiers, the face detectors using support vector machines, the Anti-face method which consists of a series of detectors trained with positive images only, and the energy-based method that simultaneously detects faces and estimates their pose in real time.

2.5 Quantifying Performance

There are numerous metrics to gauge the performance of face detection systems, ranging from detection frame rate, false positive/negative rate, number of classifier, number of feature, number of training image, training time, accuracy and memory requirements. In addition, the reported performance also depends on the definition of a "correct" detection result. The most commonly adopted method is to plot the ROC curve using the de facto standard MIT + CMU data set which contains frontal face images. Another data set from CMU contains images with faces that vary in pose from frontal to side view. It has been noticed that although the face detection methods nowadays have impressive real-time performance, there is still much room for improvement in terms of accuracy. The detected faces returned by state-of-the-art algorithms are often a few pixels (around 5) off the "accurate" locations, which is significant as face images are usually standardized to 21 by 21 pixels. While such results are the trade-offs between speed, robustness and accuracy, they inevitably degrade the performance of any biometric applications using the contents of detected faces. Several post-processing algorithms have been proposed to better locate faces and extract facial features (when the image resolution of the detected faces is sufficiently high).

2.6 Applications

As face detection is the first step of any face processing system, it finds numerous applications in face recognition, face tracking, facial expression recognition, facial feature extraction, gender classification, clustering, attentive user interfaces, digital cosmetics, biometric systems, to name a few. In addition, most of the face detection algorithms can be extended to recognize other objects such as cars, humans, pedestrians, and signs, etc.

Chapter 3

---

3. STRUCTURE OF DETECTOR SYSTEM

3.1 Individual Face Detection Networks

There are two stages of operation for the system. First it applies a set of neural network-based detectors to an image, and then uses an arbitrator to combine the output. The individual detectors examine each location in the image at several scales, looking for locations that might contain a face. The arbitrator then merges detections from individual networks and eliminates overlapping detections. The first component of our system is a neural network that receives as input a 20X20 pixel region of the image, and generates an output ranging from 1 to -1, signifying the presence or absence of a face, respectively. To detect faces anywhere in the input, the network is applied at every location in the image. To detect faces larger than the window size, the input image is repeatedly reduced in size (by sub-sampling), and the detector is applied at each size. This network must have some invariance to position and scale. The amount of invariance determines the number of scales and positions at which it must be applied. For the work presented here, I apply the network at every pixel position in the image, and scale the image down by a factor of 1.1 to 1.2 for each step in the pyramid. After a 20x20 pixel window is extracted from a particular location and scale of the input image pyramid, it is preprocessed using the lighting correction and histogram equalization. The preprocessed window is then passed to a neural network. The network has retinal connections to its input layer. The input window is broken down into smaller pieces, of four 10x10 pixel regions, sixteen 5x5 pixel regions, and six overlapping 20X5 pixel regions. Each of these regions will have complete connections to a hidden unit. The network has a single, real-valued output, which indicates whether or not the window contains a face.

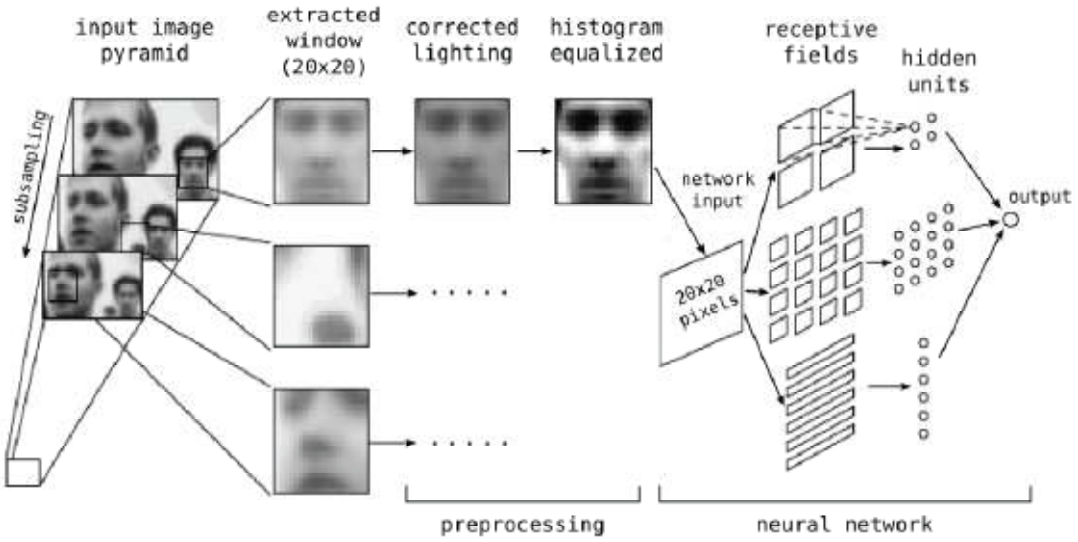


Figure 1: The basic algorithm used for face detection

3.2 Merging Overlapping Detections

The raw output from a single network will contain a number of false detections. The result can be improved by merging overlapping detections from the network. Most faces are detected at multiple nearby positions or scales, while false detections often occur with less consistency. This observation leads to a heuristic which can eliminate much false detection.

For each location and scale at which a face is detected, the number of detections within a specified neighborhood of that location can be counted. If the number is above a threshold, then that location is classified as a face. The centroid of the nearby detections defines the location of the detection result, thereby collapsing multiple detections. In the experiments section, this heuristic will be referred to as thresholding. If a particular location is correctly identified as a face, then all other detection locations which overlap it are likely to be errors, and can therefore be eliminated. Based on the above heuristic regarding nearby detections, I preserve the location with the higher number of detections within a small neighborhood, and eliminate locations with less detection.

Later, in the discussion of the experiments, this heuristic is called overlap elimination. There are relatively few cases in which this heuristic fails; however, in which one face partially occludes another. If a particular location is correctly identified as a face, then all other detection locations which overlap it are likely to be errors, and can therefore be eliminated. Based on the above heuristic regarding nearby detections, I preserve the location with the higher number of detections within a small neighborhood, and eliminate locations with fewer detections.

Each detection by the network at a particular location and scale is marked in an image pyramid, labeled the “output" pyramid. Then, each location in the pyramid is replaced by the number of detections in a specified neighborhood of that location. This has the effect of spreading out the detections. Normally, the neighborhood extends an equal number of pixels in the dimensions of scale and position, but for clarity detections are only spread out in position. A threshold is applied to these values, and the centroids (in both position and scale) of all above threshold regions are computed. All detections contributing to the centroids are collapsed down to single points. Each centroid is then examined in order, starting from the ones which had the highest number

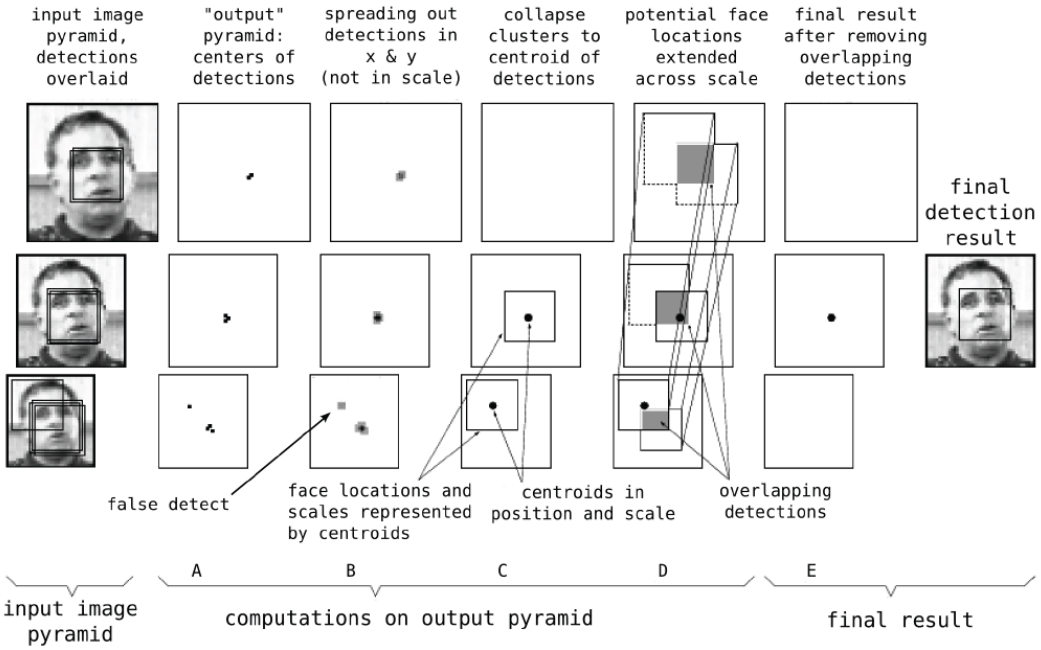


Figure 2: The framework used for merging multiple detections from a single network : (A) The detections are recorded in an image pyramid. (B) The detections are spread out and a threshold is applied. (C) The centroids in scale and position are computed, and the regions contributing to each centroid are collapsed to single points. In the example shown, this leaves only two detections in the output pyramid. (D) The final step is to check the proposed face locations for overlaps, and (E) to remove overlapping detections if they exist. In this example, removing the overlapping detection eliminates what would otherwise be a false positive.

If any other centroid locations represent a face overlapping with the current centroid, they are removed from the output pyramid. All remaining centroid locations constitute the final detection result.

3.3 Face Extraction Using Adaboost and Cascaded Detector

Cascaded face detector by the use of AdaBoost algorithm is based on the paper by Paul Viola and Michael J. Jones. The paper describes a face detection framework that is capable of processing images extremely rapidly while achieving high detection rates. There are three key contributions. The first is the introduction of integral image for rapid computation of the features used in the detector. The second is the AdaBoost algorithm for selection of efficient classifiers from a large population of potential classifiers. Third is the method for combining the classifiers generated by AdaBoost algorithm into a cascade, which has the property of removing most of the non-face images in the early stage by simple processing, and focus on complex face like regions in the later stages which take higher processing time. This algorithm is selected for the face detection because of its property of detecting faces extremely rapidly which would help in our project in processing real time videos.

3.4 Features

AdaBoost algorithm classifies images based on the value of simple features. The simple features are the reminiscent of Haar basis functions. In this method of face detection, I have used three kinds of features: two rectangle feature, three rectangle feature and four rectangle feature. Two rectangle feature is the difference between the sum of the pixels within two rectangular regions. The regions have the same size and are horizontally or vertically adjacent. Similarly three rectangle feature is the value obtained by subtracting the sum of pixels of outer rectangles from the center rectangle. Finally, four rectangle feature is the difference between the diagonal pairs of rectangles. In short, the feature value is the difference between the sum of pixels of the white region to the dark region of the features.

3.4.1 Computing Features

Given the base resolution of the detector sub-window is 24x24, the exhaustive set of rectangle features is quite large, more than 160,000. In simple words, the features have to be generated such that it starts from every possible pixel of the detector sub-window and also should cover all the widths and heights possible. This should be done for all types of features (two rectangle, three rectangle and four rectangle features). The algorithm for

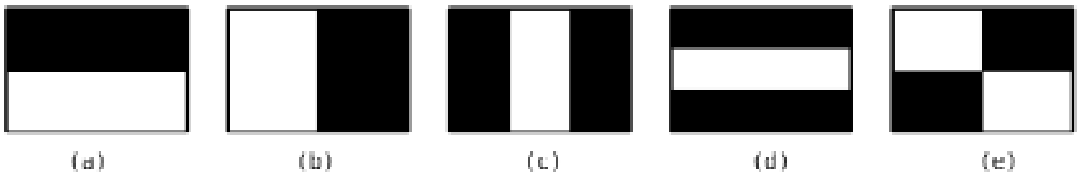


Figure 3: Features used in AdaBoost algorithm. (a) and (b) are two rectangle features, (a) being two vertically stacked rectangular feature. Similarly, (c) and (d) are three rectangle feature and (e) is the four rectangle feature

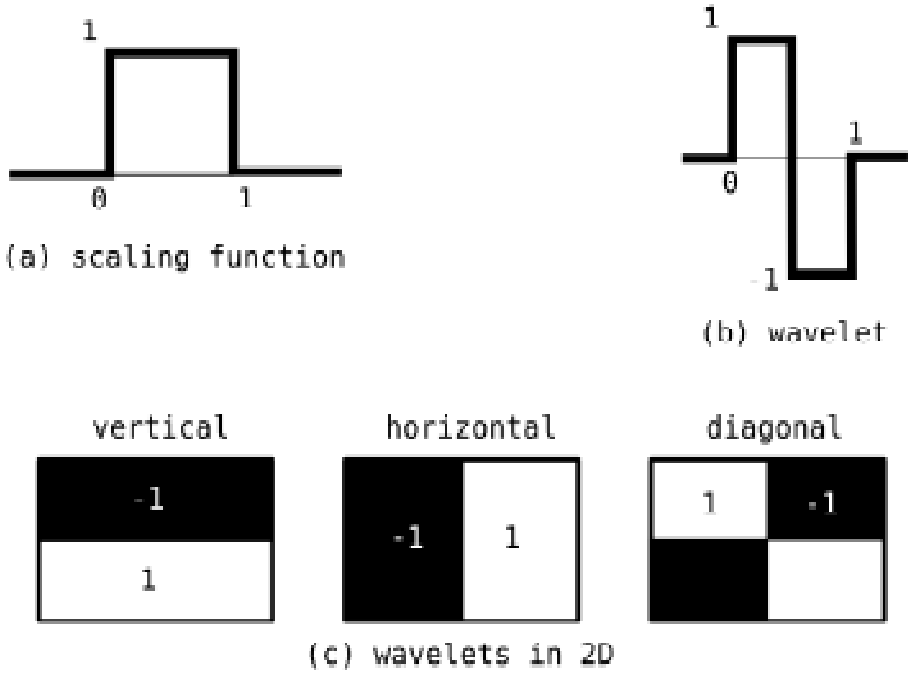


Figure 4: The Haar wavelet framework: (a) and (b) the Haar scaling function and wavelet, (c) the three types of 2-dimensional non-standard Haar wavelets: vertical, horizontal and diagonal

the feature generation is given below:

i. Start from the initial position of the sub-window, (1; 1). Create a rectangular feature of size such that 1 pixel represents the black region and 1 pixel represents the white region

ii. Move the feature throughout the sub-window

iii. Increase the size of the feature such that both the black region and white region size increases by 1 in horizontal direction. Goto step 2, until the width of the feature equals the width of the sub-window

iv. Increase the size of the feature such that both the black region and dark regionsize increases by 1 in vertical direction. Goto step 2 until feature height equals the height of the detector sub-window

v. For all the starting position and all the width and height of the feature (from steps 1 to 4), store the co-ordinates as (x, y, w, h) where x= starting x-coordinate, y = starting y-coordinate, w = width and h = height of the feature

vi. Repeat steps 1 to 5 for all types of rectangular features (i.e. two rectangle, three rectangle and four rectangle feature)

Total number of features generated is 162,336. The feature count of different features are given below.

- Two vertically stacked rectangle feature : 43,200
- Two horizontally stacked rectangle feature: 43,200
- Three vertically stacked rectangle feature : 27,600
- Three horizontally stacked rectangle feature: 27,600
- Four rectangle feature: 20,736

Generation of three horizontally stacked features and achieved by the pseudo-code given below:

```
H=24; W=24;
        for (h=1; h<=H; h++)
                for (w=3; w<=W; w+=3)
                        for (y=0; y<=H-h; y++)
                                for (x=0; x<=W-w; x++)
                                        Save(x, y, w, h)
```

This pseudo-code saves only the starting position with the width and height. To distinguish between the type of feature (like two vertically stacked, two horizontally stacked, three horizontally stacked etc), it is distinguished by saving a type as well. So, during the calculation, the type shows that this feature is a horizontally stacked three rectangle feature and the middle rectangle has to be subtracted from the outer rectangles.

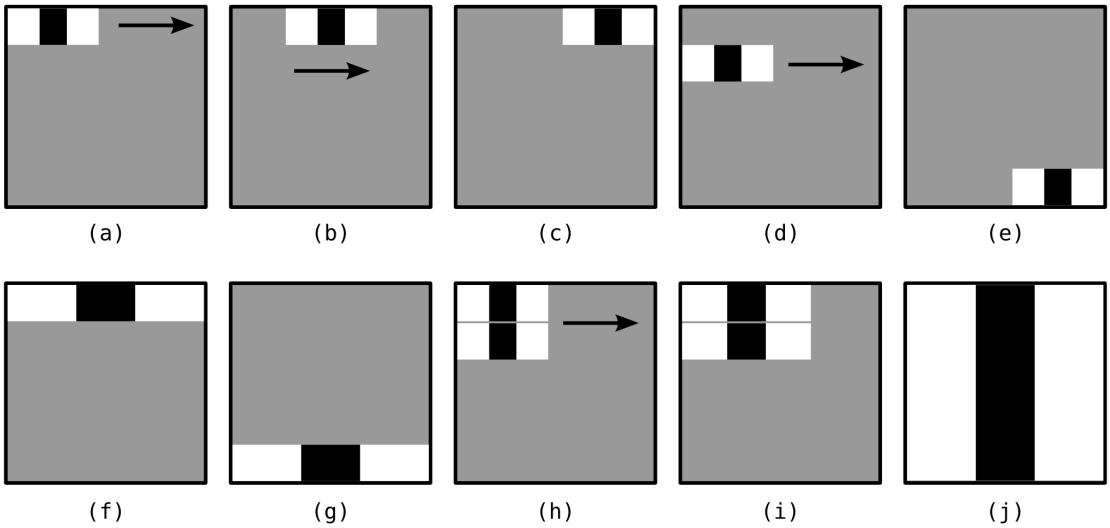


Figure 5: Process of generation of three horizontally stacked rectangle features : (a) Initially 1 pixel each of white and black region is at the initial position (1, 1). It is then moved throughout the window (a to e). The feature is then increased in width by one pixel each time and moved throughout the window until the width of the feature become equal to the width of the window (f to g). Then the feature height is increased by one pixel (h) each time and moved throughout the window until the width of the feature become equal to the height of the window (i). This process continues until the whole feature covers the window (j)

3.5 Integral Image

For the fast computation of feature values, integral image of an image can be used. Integral image size is the same size of the image, but the value at any location x,y contains the sum of the pixels above and to the left of x, y

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y')$$

where, $ii(x, y)$ is the integral image and $i(x, y)$ is the original image. So, for the computation of $ii(x, y)$, following pair of recurrences can be used:

$$s(x, y) = s(x, y - 1) + i(x, y)$$
$$ii(x, y) = ii(x - 1, y) + s(x, y)$$

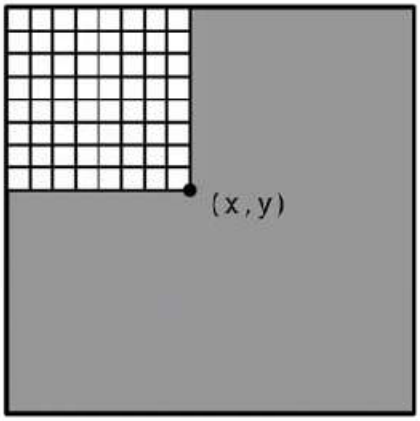


Figure 6: Integral image is the sum of the pixels at the left and above of the point (x,y) inclusive

where $s(x, y)$ is the cumulative row sum and $s(x,-1) = 0$, and $ii(-1, y) = 0$. For the

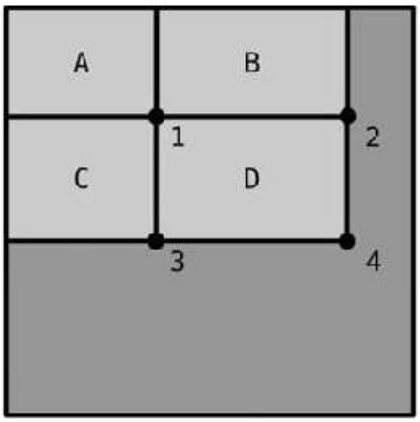


Figure 7: The sum of the pixels within rectangle D can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle A. The value at location 2 is A + B, at location 3 is A + C, and at location 4 is A + B + C + D. The sum within D can be computed as 4 + 1 − (2 + 3)

computation of features using integral image, only few array references has to be done.

For two rectangle feature, 6 array references has to be done. For three rectangle feature, 8 array references has to be done and for four rectangle features, 9 array references has to be done. Few simple operations like addition and subtraction has to be done for the feature evaluation, so the feature value computation is very fast using the technique of integral image. Initial computation of the integral image takes more time, but it has to be only computed once, and after its computation any feature can be computed rapidly using the same integral image with the maximum of 9 array references per feature. The computation of feature value for three horizontally stacked rectangle

feature is shown in the Figure. For the three horizontally stacked rectangle feature, eight points are computed as shown. Then the integral image value at that point $ii(x, y)$ is found at each point. The feature value is computed as $p1+p8-(p5+p4)-2*(p2+p7-(p3+p6))$, where pN is the integral image value at point N. This value is computed using the technique

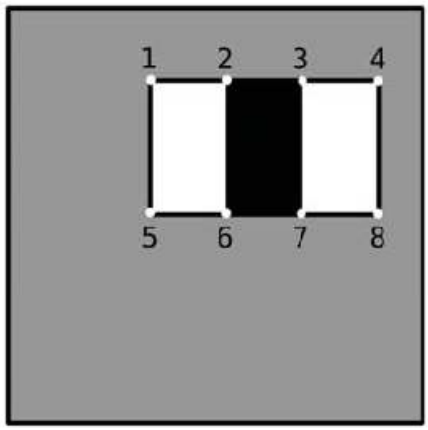


Figure 8: Feature evaluation using integral image

shown in figure 8 for computing value for a single rectangle.

For the face detection faces at multiple levels, other techniques use the method of image pyramid. Using the method of integral image, the detection of faces at multiple level is much faster than the image pyramid method. This is because in integral image method, the feature rectangle can be increased in size instead of computing the image pyramid. The same integral image can be used for the feature evaluation requiring same computation time as that at the original feature scale. But in case of image pyramid method, the scaling of the image itself takes a lot of time and further the detection time is high in all level of image pyramids.

3.6 Adaboost Algorithm

In a single subwindow of size 24×24, there are more than 160,000 features. Computation of all these features in detection can be very expensive in time. But there are efficient classifiers among these large number of features which when efficiently combined gives a robust classifier. The selection and combination of these efficient classifiers from among the large set is done by AdaBoost algorithm. Viola and Jones have given a variant of AdaBoost algorithm to train the classifiers. AdaBoost algorithm is given a feature set with large number of features and the training images (labeled faces and non-faces that is supervised learning). AdaBoost learning algorithm is used to boost the classification performance of simple learning algorithm (eg. a simple perceptron). It does this by combining a collection of weak classification functions to form a strong classifier. In the language of boosting the weak classification functions are called weak learners, because I do not expect even the best weak learner to classify the

training data well (that is, for a given problem even the best perceptron may classify the training data correctly only 51% of time). The selection of the best classifier for each round is the one which gives lowest classification error in the training data. Boosting is the method of combining the weak classifiers to form a strong classifier. In the later round of learning, AdaBoost finds the new weak classifier after re-weighting the training examples such that it emphasizes on the examples incorrectly classified by the previous weak classifiers. The weak classification function selects a single rectangular feature which best separates the positive and negative examples, determines the threshold and gives a weak classifier. So, weak classifier ($hj(x)$) consists of a rectangular feature (fj), threshold (θj) and a polarity (pj) indicating the direction of inequality:

$$h_j(x) = \begin{cases} 1 & \text{if } p_j \cdot f_j(x) < p_j \cdot \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Boosting algorithm which when run for T number of rounds selects T number of best weak classifiers and combines these to form a strong classifier also finding the threshold for the strong classifier. The boosting algorithm used for the selection of the rectangular feature and combining those to form a strong classifier is given below:

1. Given example images $(x_1, y_1), \ldots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively
2. Initialize weights $W_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the numbers of negatives and positives respectively
3. For $t = 1, \ldots, T$:
   (a) Normalize the weights
   $w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$ so that w_t is a probability distribution
   (b) For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t
   $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$
   (c) Choose the classifier, h_t, with the lowest error ϵ_t
   (d) Update the weights:
   $w_{t+1,i} = w_{t,i} \beta_t^{1-\epsilon_i}$
4. The final strong classifier is:

$$h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

The description of how the algorithm works is now presented. Labeled faces and non faces (labeled 1 and 0) are provided. The initial weight of all the examples (training faces and non faces) is calculated . Then the boosting iteration starts. In each iteration, the weights are normalized and for each feature a threshold is computed which then gives a weak classifier (computation of threshold is explained later). For each weak classifier, its error of classification in the training examples is computed by weighted sum. Among the more than 160,000 features, the classifier with the minimum of the errors is evaluated, and this is the weak classifier selected by AdaBoost algorithm for this round of boosting.

For the next round of boosting, the weights are updated such that the weight for the examples wrongly classified by previous classifier is higher than the ones that are rightly classified. This way, in the next round the classifier is selected which focuses on these examples for finding the efficient classifier and the classifiers selected for different round of boosting will be different. The final strong classifier is created by combining the weak classifiers generated and setting the threshold as half the weight given to the classifiers.

For the computation of the threshold for each of the features, a table has to be maintained. The examples are sorted according to the feature value. For each element in the sorted list, four sums are maintained and evaluated: the total sum of positive example weights T^+, the total sum of negative example weights T^-, the sum of positive weights below the current example S^+ and the sum of negative weights below the current example S^-. The error for a threshold which splits the range between the current and previous example in the sorted list is:

$$e = min(S^+ + (T^- - S^-), S^- + (T^+ - S^+))$$

or the minimum of the error of labeling all examples below the current example negative and labeling the examples above positive versus the error of the converse. Then the feature value is selected as threshold for that feature which has the minimum of these error values. After finding the threshold, I also need to find the polarity of the classifier. Polarity gives the direction of inequality, that is if the feature value of an example is greater than the threshold, should it be labeled as a face or a non face. To find the polarity, following relation is used for the example from which threshold is found:

$$p = \begin{cases} 1 & \text{if } S^+ + (T^- - S^-) < S^- + (T^+ - S^+) \\ -1 & \text{otherwise} \end{cases}$$

A sample of the table maintained to compute the threshold for each feature is shown in Table. All the examples are sorted according to the feature value. The values of $T^+ = 0.045$ and $T^- = 0.031$ can be computed only once for each feature, and hence are not shown in the table. Here, the threshold for this feature is computed as 23 and the polarity as -1.The difference between the intensity of the region of eyes and the sum of region of forehead and cheek is higher in most of the faces and not always high in non faces. So, this feature gives lower classifying error and AdaBoost selects this as a feature. AdaBoost

algorithm takes a very long time for the training. This is because more than 160,000 features are to be processed in selecting one of the features in one round of boosting. A lot of time is taken for finding the threshold and evaluating error of each

Table 1: Table maintained to compute the threshold for each feature

error	weight	example (+/-)	feature value	S^+	S^-
0.028	0.017	1	-30	0.028	0.031
0.008	**0.02**	**1**	**23**	**0.008**	**0.031**
0.0235	0.012	0	89	0	0.019
0.031	0.019	0	97	0	0
0.031	0.008	1	105	0	0

Figure 9: One of the first features selected by AdaBoost algorithm feature.

Also, the number of training examples is also very large.

3.7 Cascade of Classifiers

AdaBoost algorithm is capable of making a good detector with the use of many features which can result in high detection rate. But a monolithic detector made by using AdaBoost only has low detection speed. This is because the feature values have to be evaluated and classified for all the sub-windows, regardless of its complexity. To increase the detection speed, several stages of the classifiers are made which reject most of the sub-windows at the early stage and give more time on less number of sub-windows which are complex and have face-like regions. The cascade is made to reject the sub-windows because the number of non face sub-windows is far greater than the number of face subwindows in an image. Each stage (strong classifier) is trained by AdaBoost algorithm. When a subwindow is passed to the cascade, most of the simple sub-windows not containing faces are rejected by simple processing of two or three features. Computation time increases as the stage number increases because of the complexity of the sub-window. More and more non face sub-windows are rejected as the sub-window moves further into the cascade. The sub-windows which are classified as face by all the stages are the final detections.

3.7.1 Training of Cascade of Classifiers

For the training of the cascade of classifiers, each stage is generated by using AdaBoost algorithm. To achieve high detection rate and low false positive rate of the final detector, each stage is made such that it has the detection rate greater or equal to a given value and false positive rate less than or equal to a given value. If this condition is not met, the stage is again trained by specifying larger number of classifiers than the previous. The

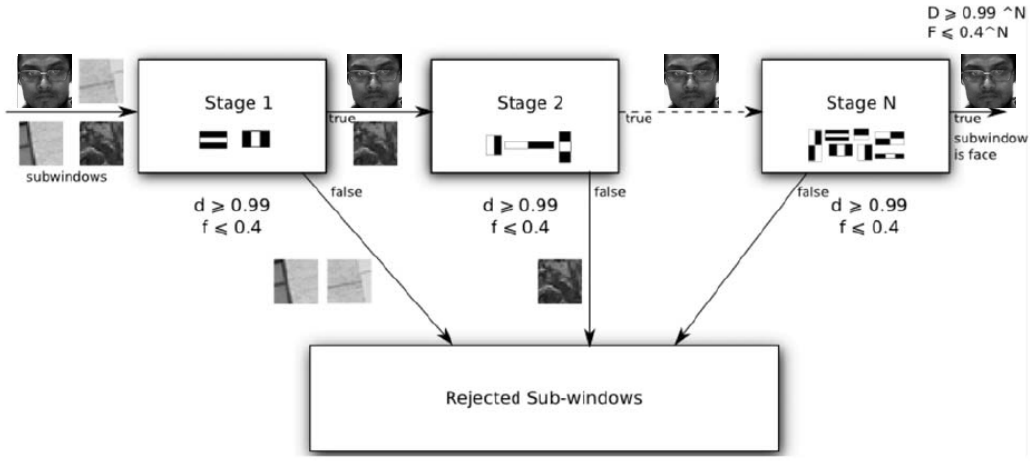


Figure 10: Cascade of classifiers for the face detection

final false positive rate and the detection rate of the cascade is given as:

$$F = \prod_{i=1}^{K} f_i$$

where, *F* is the false positive rate of the cascaded classifier, *fi* is the false positive rate of the ith stage and *K* is the number of stages. The detection rate is:

$$D = \prod_{i=1}^{K} d_i$$

Where *D* is the detection rate of the cascaded classifier, *di* is the detection rate of ith stage and *K* is the number of stages. Every face detection system requires high detection rate and a low false positive rate. Given the concrete goal for overall false positive and detection rates, target rates can be determined for each stage in the cascade process. For example, detection rate of 0.9 can be achieved by 10 stages with the detection rate of each stage *di* = 0.99(0.9910 = 0.9). The detection rate of 0.99 for each stage can be found for a high false positive rate. But for 10 stages, if the false detection rate of each stage is 0.4, then the final false detection rate will be 0.000105(0.410).

For the training of the cascade, initially faces and non-faces for the training and faces and non-faces for the validation are provided. AdaBoost algorithm is designed to find a strong classifier which best classifies the faces from the non-faces and is not designed to produce high detection rate with high false detection rate. So, to achieve high detection rate of say 0.99, the stage has to be checked with the validation image to find the current detection rate, and the threshold of the strong classifier has to be reduced to achieve the detection rate of 0.99. In doing so, the false positive rate also increases, and if it is greater than the desired false positive rate (say 0.4), then the stage has to be trained again with more number of features included. After the generation of one stage, the non-face training images for the next stage are generated by running the detector up to the previous stage on a set on non-face containing images. These false detections are then used as non-face training images for the training of the new stage. This way, the complexity of the non-face increases as the number of stages increases, resulting in more and more face-like non-face training images in later stages. So, the number of classifiers in the stage also increases as the stage number increases.

3.7.2 Training Algorithm For Cascade Detector

The training algorithm is as follows:

- User selects values of f, the maximum acceptable false positive rate per layer and d, the minimum acceptable detection rate per layer
- User selects target overall false positive rate, $Ftarget$
- P = set of positive examples
- N = set of negative examples
- $F0 = 1.0; D0 = 1.0$
- $i = 0$
- while $Fi > Ftarget$

  – $i \leftarrow i + 1$

  – $ni = 0$; $Fi = Fi-1$

  – while $Fi > f \times Fi-1$

  ∗ $ni \leftarrow ni + 1$

  ∗ Use P and N to train a classifier with ni features using AdaBoost

  ∗ Evaluate current cascade classifier on validation set to determine Fi and Di

  ∗ Decrease threshold for the ith classifier until the current cascade classifier has a detection rate of at least $d \times Di-1$ this also affects Fi)

– $N \leftarrow \varnothing$

– If $Fi > Ftarget$

then evaluate the current cascade detector on the set of non-face images and put any false detections into the set *N* The algorithm for the generation of cascade of classifiers is shown in the above.

First, the maximum acceptable false positive rate per stage *f*, minimum acceptable detection rate per layer *d* and overall target false positive rate of the final cascade *Ftarget* has to be defined. Also, the training set of positive *P* and negative examples *N*, set of validation positive and negative examples, and the non face source images from which the non faces for the later stages of the cascade will be generated, has to be provided. Then the stages of the cascade are generated using AdaBoost algorithm. Once the stage is generated, its threshold is reduced to meet the requirement of minimum detection rate. If reduction of threshold in doing so does not meet the requirement for the maximum false positive rate, more classifiers are added to the stage using AdaBoost algorithm. After these conditions are met, all the non face training images are removed and new ones are generated from the non face source images using the cascade generated so far. Then the new stage generation starts. This continues until the final false positive rate is less than or equal to the target false positive rate.

3.8 Learning Results

While details on the training and performance of the final system are presented , several simple results merit discussion. Initial experiments demon started that a classifier constructed from 200 features would yield reasonable results.

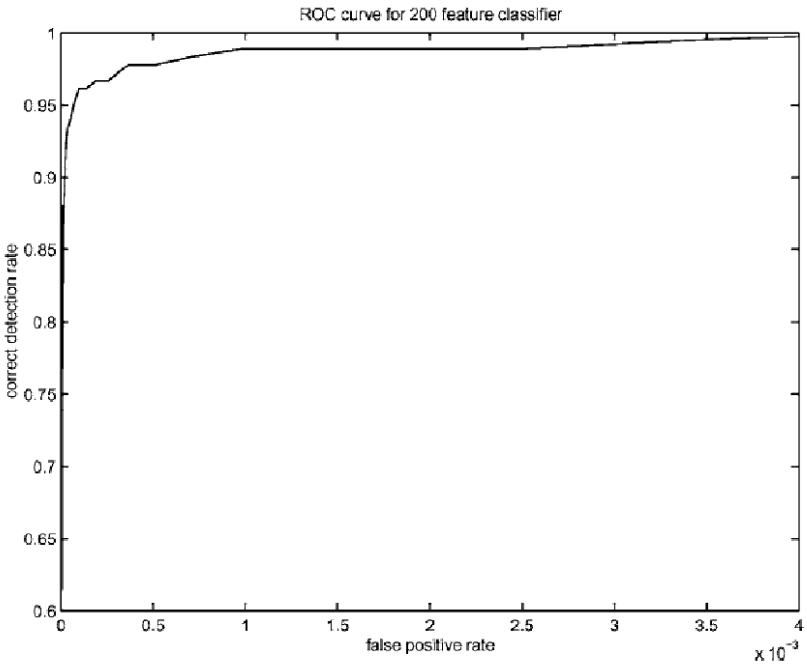


Figure 11: Receiver operating characteristic (ROC) curve for the 200 feature classifier.

Given a detection rate of 95% the classifier yielded a false positive rate of 1 in 14084 on a testing dataset. This is promising, but for a face detector to be practical for real applications, the false positive rate must be closer to 1 in 1,000,000.

For the task of face detection, the initial rectangle features selected by AdaBoost are meaningful and easily interpreted. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. This feature is relatively large in comparison with the detection sub-window, and should be somewhat insensitive to size and location of the face. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.

In summary the 200-feature classifier provides initial evidence that a boosted classifier constructed from rectangle features is an effective technique for face detection. In terms of detection, these results are compelling but not sufficient for many real-world tasks. In terms of computation, this classifier is very fast, requiring 0.7 seconds to scan an 384 by 288 pixel image. Unfortunately, the most straightforward technique for improving detection performance, adding features to the classifier, directly increases computation time.

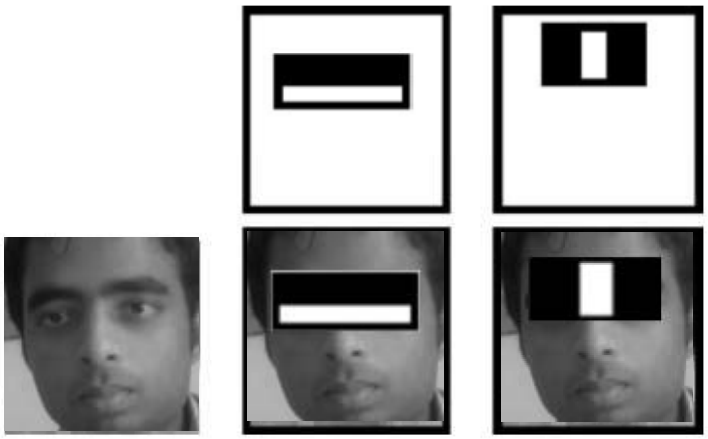


Figure 12: The first and second features selected by AdaBoost. The two features are shown in the top row and then overlayed on a typical training face in the bottom row. The first feature measures the difference in intensity between the region of the eyes and a region across the upper cheeks. The feature capitalizes on the observation that the eye region is often darker than the cheeks. The second feature compares the intensities in the eye regions to the intensity across the bridge of the nose.

3.9 The Attentional Cascade

This section describes an algorithm for constructing a cascade of classifiers which achieves increased detection performance while radically reducing computation time. The key insight is that smaller, and therefore more efficient, boosted classifiers can be constructed which reject many of the negative sub-windows while detecting almost all positive instances. Simpler classifiers are used to reject the majority of sub-windows before more complex classifiers are called upon to achieve low false positive rates.Stages in the cascade are constructed by training classifiers using AdaBoost. Starting with a two-feature strong classifier, an effective face filter can be obtained by adjusting the strong classifier threshold to minimize false negatives. The initial AdaBoost threshold, is designed to yield a low error rate on the training data. A lower threshold yields higher detection rates and higher false positive rates. Based on performance measured using a validation training set, the two-feature classifier can be adjusted to detect 100% of the faces with a false positive rate of 50%. The detection performance of the two-feature classifier is far from acceptable as a face detection system. Nevertheless the classifier can significantly reduce the number of sub-windows that need further processing with very few operations:

i. Evaluate the rectangle features (requires between 6 and 9 array references per feature).
ii. Compute the weak classifier for each feature (requires one threshold operation per feature).
iii. Combine the weak classifiers (requires one multiply per feature, an addition, and finally a threshold).

A two feature classifier amounts to about 60 microprocessor instructions. It seems hard to imagine that any simpler filter could achieve higher rejection rates. By comparison, scanning a simple image template would require at least 20 times as many operations per sub-window.

The overall form of the detection process is that of a degenerate decision tree, what I call a "cascade" (Quinlan, 1986). A positive result from the first classifier triggers the evaluation of a second classifier which has also been adjusted to achieve very high detection rates. A positive result from the second classifier triggers a third classifier, and so on. A negative outcome at any point leads to the immediate rejection of the sub-window. The structure of the cascade reflects the fact that within any single image an overwhelming majority of sub-windows are negative. As such, the cascade attempts to reject as many negatives as possible at the earliest stage possible. While a positive instance will

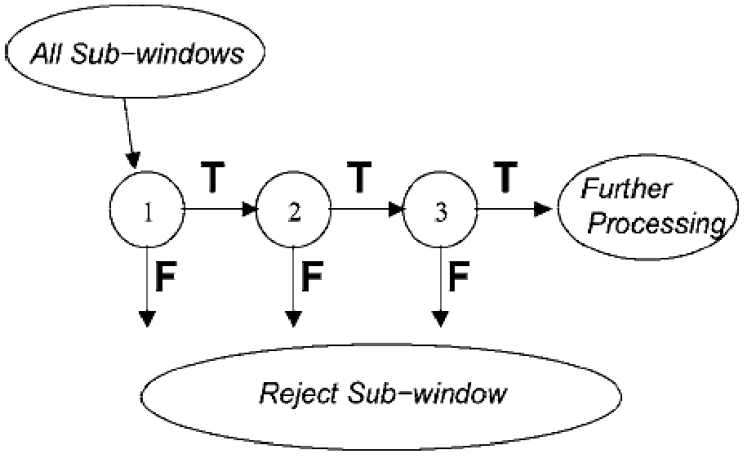


Figure 13: Schematic depiction of a the detection cascade. A series of classifiers are applied to every sub-window.

The initial classifier eliminates a large number of negative examples with very little processing. Subsequent layers eliminate additional negatives but require additional computation. After several stages of processing the number of sub-windows have been reduced radically. Further processing can take any form such as additional stages of the cascade (as in our detection system) or an alternative detection system.

Much like a decision tree, subsequent classifiers are trained using those examples which pass through all the previous stages. As a result, the second classifier faces a more difficult task than the first. The examples which make it through the first stage are "harder" than typical examples. The more difficult examples faced by deeper classifiers push the entire receiver operating characteristic (ROC) curve downward. At a given detection rate, deeper classifiers have correspondingly higher false positive rates.

3.10 Detector Cascade

There is a hidden benefit of training a detector as a sequence of classifiers which is that the effective number of negative examples that the final detector sees can be very large. One can imagine training a single large classifier with many features and then trying to speed up its running time by looking at partial sums of features and stopping the computation early if a partial sum is belowthe appropriate threshold. One drawback of such an approach is that the training set of negative examples would have to be relatively small (on the order of 10,000 to maybe 100,000 examples) to make training feasible.With the cascaded detector, the final layers of the cascade may effectively look through hundreds of millions of negative examples in order to find a set of 10,000 negative examples that the earlier layers of the cascade fail on. So the negative training set is much larger and more focused on the hard examples for a cascaded detector. A notion similar to the cascade appears in the face detection system described by Rowley et al. (1998). Rowley et al. trained two neural networks. One network was moderately complex, focused on a small region of the image, and detected faces with a low false positive rate. They also trained a second neural network which was much faster, focused on larger regions of the image, and detected faces with a higher false positive rate. Rowley et al. used the faster second network to prescreen the image in order to find candidate regions for the slower more accurate network. Though it is difficult to determine exactly, it appears that Rowley et al.'s two network face system is the fastest existing face detector. Our system uses a similar approach, but it extends this two stage cascade to include 38 stages.

The structure of the cascaded detection process is essentially that of a degenerate decision tree, and as such is related to the work of Amit and Geman (1999). Unlike techniques which use a fixed detector, Amit and Geman propose an alternative point of view where unusual co-occurrences of simple image features are used to trigger the evaluation of a more complex detection process. In this way the full detection process

need not be evaluated at many of the potential image locations and scales. While this basic insight is very

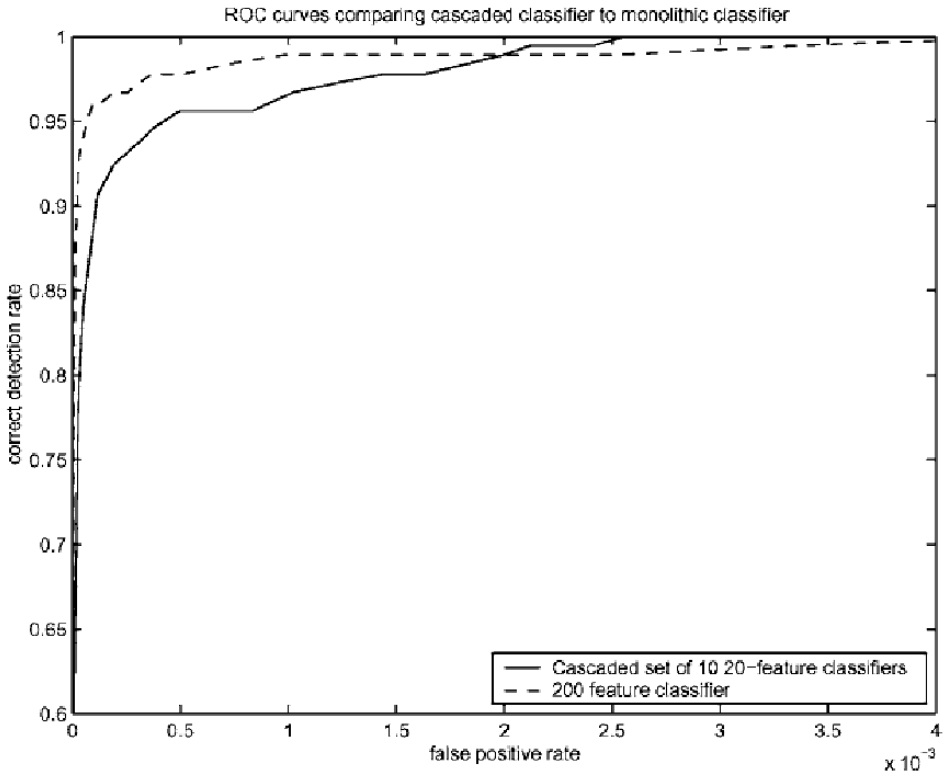


Figure 14: ROC curves comparing a 200-feature classifier with a cascaded classifier containing ten 20-feature classifiers. Accuracy is not significantly different, but the speed of the cascaded classifier is almost 10 times faster.

valuable, in their implementation it is necessary to first evaluate some feature detector at every location. These features are then grouped to find unusual co-occurrences. In practice, since the form of our detector and the features that it uses are extremely efficient, the amortized cost of evaluating our detector at *every scale and location* is much faster than finding and grouping edges throughout the image.

In recent work Fleuret and Geman (2001) have presented a face detection technique which relies on a "chain" of tests in order to signify the presence of a face at a particular scale and location. The image properties measured by Fleuret and Geman, disjunctions of fine scale edges, are quite different than rectangle features which are simple, exist at all scales, and are somewhat interpretable. The two approaches also differ radically in their learning philosophy. Because Fleuret and Geman's learning process does not use negative examples their approach is based more on density estimation, while our detector is purely discriminative.

Finally the false positive rate of Fleuret and Geman's approach appears to be higher than that of previous approaches like Rowley et al. and this approach. In the published paper the included example images each had between 2 and 10 false positives. For many practical Tasks, it is important that the expected number of false positives in any image be less than one (since in many tasks the expected number of true positives is less than one as well). Unfortunately the paper does not report quantitative detection and false positive results on standard datasets.

Chapter 4

4. FACE RECOGNITION USING SUBSPACE LDA AND EIGENFACES

- **4.1 Principle Component Analysis (PCA)**
- **4.2 Recognition Operation in PCA**
- **4.3 Linear Discriminant Analysis (LDA)**
  - **4.3.1 Training Operation in LDA**
  - **4.3.2 Recognition Operation in LDA**
  - **4.3.3 Subspace LDA**
- **4.4 Eigenface**
  - **4.4.1 Overview**
  - **4.4.2 Pattern Classes and Patterns**
  - **4.4.3 Fundamental Problems in Pattern Recognition System Design**
  - **4.4.4 Training and Learning**
  - **4.4.5 Supervised and Unsupervised Pattern Recognition**
  - **4.4.6 Outline of a Typical Pattern Recognition System**
- **4.5 Face Recognition**
  - **4.5.1 Background and Related Work**
  - **4.5.2 Outline of a Typical Face Recognition System**
  - **4.5.3 Problems during Face Recognition**
  - **4.5.4 Feature Based Face Recognition**
  - **4.5.5 Introduction to Feature Based Face Recognition**
  - **4.5.6 Effective Feature Selection**

4. FACE RECOGNITION USING SUBSPACE LDA AND EIGENFACES

There are basically three approaches to face recognition :

Holistic approach This approach works on the whole face region as the raw input to the recognition system. It doesn't consider individual features like eye, nose, mouth etc but works on the whole face for comparing similarity. Algorithms like PCA (Principle Component Analysis) and LDA(Linear Discriminant Analysis) are examples of holistic approach based face recognition algorithms.

Feature based approach Unlike holistic approach, feature based approach considers individual features like eyes, nose, mouth, mole etc and compares the similarity between the individual features for a match. Algorithms like Elastic Bunch Graph Matching (EBGM), Face recognition using Gabor Wavelet Transform.

Hybrid approach Hybrid approach is the mixture of holistic approach and feature based approach. Compared to the individual holistic and feature based approach, hybrid approach is considered to have the best performance.
Subspace Linear Discriminant Analysis (LDA) is a hybrid algorithm for face recognition. It was initially proposed in followed by. It consists of two separate face recognition algorithms which are:

- Principle Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

4.1 Principle Component Analysis (PCA)

PCA is one of the most primitive algorithms used for face recognition proposed by Pentland and Turk. It is a holistic approach based face recognition algorithm. It is also known as Eigenfaces. Any image (suppose of dimension $N_x x N_y$) can be thought of as a point in a P(=N_x X N_y) dimensional image space having values in the range of pixel values. For the case of gray scale images, in each dimension the image could have a value in between 0 and 255. An image can be thought as a point in the image space by converting the image to a long vector by concatenating each column of the image one after the other.
The Eigenface method tries to find a lower dimensional space for the representation of the face images by eliminating the variance due to non-face images; that is, it tries to focus on the variation just coming out of the variation between the face images. Eigenface method is the implementation of Principal Component Analysis (PCA) over images. In this method, the features of the studied images are obtained by looking for the maximum deviation of each image from the mean image. This variance is obtained by getting the eigenvectors of the covariance matrix of all the images.
The Eigenface space is obtained by applying the eigenface method to the training images. Later, the training images are projected into the Eigenface space. Next, the test image is projected into this new space and the distance of the projected test image to the training images is used to classify the test

image. The train image projection with the minimum distance from the test image projection is the correct match for that test face.

Training operation in PCA

Let I be an image of size (N_x, N_y) pixels, then the training operation of PCA algorithm can be expressed in mathematical terms as

i. Convert the training image matrix I of size (N_x, N_y) pixels to the image vector τ of size (P X 1) where P = N_x X N_y (ie: the train image vector τ is constructed by stacking each column of train image matrix I).

ii. Create a training set of training image vectors such that its size is P X M_t where M_t is the number of training images $\tau_{\text{P X Mt}} = [\tau_1\ \tau_2 \ldots\ \tau_{\text{Mt}}]$ where, τ_i represents the image vector of i^{th} training images.

iii. Compute arithmetic average (mean face Ψ) of the training image vectors at each pixel point given by:

$$\Psi_{P\times 1} = \frac{1}{M_t}\sum_{i=1}^{M_t}\Gamma_i$$

iv. Obtain mean subtracted vector (Φ) by subtracting the mean face from the each training image vector as given below:

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

v. Create the difference matrix (A) which is the matrix of all the mean subtracted vectors (Φ) and is given by:

$$A_{P\times M_t} = [\Phi_1 \Phi_2 \ldots \Phi_{M_t}]$$

vi. Compute the covariance matrix (X) given by:

$$X_{PXP} = A\ X\ A^T$$

7. Compute the eigenvector and eigenvalue of the covariance matrix (X). The dimension of covariance matrix (X) is $P \times P = (N_x \cdot N_y) \times (N_x \cdot N_y)$. For an image of typical size, the task of computing eigenvector of a matrix of such huge dimension is a computationally intensive task. However a way around this problem exists and is described below: Let us define a matrix $Y_{M_t \times M_t}$ such that

$$Y = A^T \cdot A$$

$$\begin{aligned}
Y \cdot \nu_i &= \mu_i \cdot \nu_i \\
(A^T \cdot A) \cdot \nu_i &= \mu_i \cdot \nu_i && (\because Y = A^T \cdot A) \\
(A \cdot A^T \cdot A) \cdot \nu_i &= A \cdot \mu_i \cdot \nu_i && \text{(premultiplying both sides by matrix A)} \\
A \cdot A^T \cdot A \cdot \nu_i &= \mu_i \cdot (A \cdot \nu_i) && \text{(as } \mu_i \text{ is scalar, we can rearrange the terms)} \\
X \cdot (A \cdot \nu_i) &= \mu_i \cdot (A \cdot \nu_i) && (\text{where } X = A \cdot A^T) \\
X \cdot \vartheta_i &= \mu_i \cdot \vartheta_i && (\text{where } \vartheta_i = (A \cdot \nu_i))
\end{aligned}$$

which is of the form $B \cdot I = \lambda I$ where I and λ are the eigenvector and eigenvalue of B. Hence, I conclude that ϑ and μ_i is one of the the eigenvectors and eigenvalues of matrix $X = A \cdot AT$ respectively. Thus I obtain the eigenvector $\vartheta_i (= A \cdot \vartheta_i)$ of covariance matrix $X = A \cdot AT$ by first computing the eigenvector (ϑ_i) of matrix $Y = AT \cdot A$. This formulation brings substantial computational efficiency. Some example images and mean of the images from the ORL database are given below. The eigenfaces are in fact $(P \times 1)$ vectors for the computations; in order to see what they look like, they are rearranged as $(Nx \times Ny)$ matrices. Instead of using Mt

Figure 15 : Sample Faces of ORL Database

Figure 16: Sample Mean Face

Of the eigenfaces, $M' \leq Mt$ of the eigenfaces can be used for the eigenface projection. This is achieved to eliminate some of the eigenvectors with small eigenvalues, which contribute less variance in the data. Eigenvectors can be considered as the vectors

Figure 17: Some example faces of Eigenface

pointing in the direction of the maximum variance and the value of the variance the eigenvector represents is directly proportional to the value of the eigenvalue (i.e, the larger the eigenvalue indicates the larger variance the eigenvector represents).

Hence, the eigenvectors are sorted with respect to their corresponding eigenvalues. The eigenvector having the largest eigenvalue is marked as the first eigenvector, and so on. In this manner, the most generalizing eigenvector comes first in the eigenvector matrix. In the next step, the training images are projected into the

eigenface space and thus the weight of each eigenvector to represent the image in the eigenface space is calculated. This weight is simply the dot product of each image with each of the eigenvectors.

vii. Determine the projection of a training image on each of the eigenvectors as given below:

$$\omega_k = \vartheta_K^T \cdot \Phi = \vartheta_K^T \cdot (\Gamma - \Psi) \qquad k = 1, 2, \ldots, M_t$$

viii. Determine the weight matrix Ω - which is the representation of the training images in eigenface space

$$\Omega_{M' \times 1} = [\omega_1 \ \omega_2 \ . \ . \ . \ \omega_{M'}]^T$$

The training operation of PCA algorithm ends with the computation of the weight matrix. At this point, the images are just composed of weights in the eigenface space, simply like they have pixel values in the image space. The important aspect of the eigenface transform lies in this property. Each image is represented by an image of size ($Nx \times Ny$) in the image space, whereas the same image is represented by a vector of size ($M' \times 1$) in the Eigenface space.

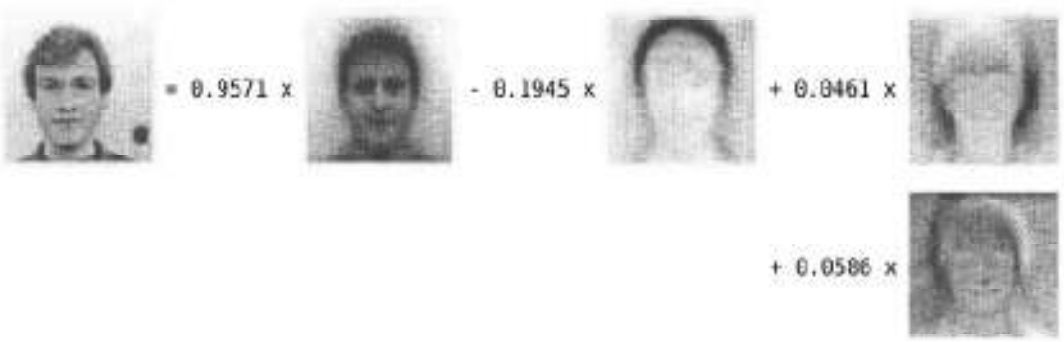


Figure 18: representation of training face images as weighed sum (given by Ω) of eigenfaces

4.2 Recognition Operation in PCA

When a new probe(test) image is to be classified, it is also mean subtracted and projected onto the eigenface space and the test image is assumed to belong to the nearest class by calculating the Euclidean distance of the test image projections to that of the training image projections.Let T be an image of size (Nx,Ny) pixels, then the recognition operation of PCA algorithm can be expressed in mathematical terms as:

i. Convert the test image matrix T of size (Nx,Ny) pixels (*the size of test image must be same as that of the training images*) to the image vector Γ_T of size ($P \times 1$) where $P = Nx \times Ny$ (ie: the test image vector ΓT is constructed by stacking each column of test image matrix T)

ii. Obtain mean subtracted test image vector (ΦT) by subtracting the mean face (computed during the training session) from the test image vector as given below:

$$\Phi_T\,(P \times 1) = \Gamma_T - \Psi$$

iii. Determine the projection of a test image on each of the eigenvectors as given below:

$$\omega_k = \vartheta_K^T \cdot \Phi = \vartheta_K^T \cdot (\Gamma - \Psi) \qquad k = 1, 2, \ldots, M_t$$

iv. Determine the weight matrix Ω - which is the representation of the test image in eigenface space

$$\Omega_{M' \times 1} = [\omega_1\ \omega_2 \ldots \omega_{M'}]^T$$

v. Compute the value of similarity function of given test image for each training image. Similarity of test image with *ith* training image is defined as:

$$\delta_i = \|\Omega_T - \Omega_{\Psi_i}\| = \sqrt{\sum_{k=1}^{M'} (\Omega_{T_k} - \Omega_{\Psi_{i_k}})^2}$$

where (δi) is the Euclidean distance (L2 norm) between projections of images in face space.

The training face image which has the minimum distance (δ) is the face that is closely matching with the test face.

4.3 Linear Discriminant Analysis (LDA)

LDA is another holistic approach based face recognition method proposed by Etemad and Chellapa. However unlike Eigenfaces which attempts to maximize the scatter of the training images in face space, it attempts to maximize the between class scatter, while minimizing the within class scatter. In other words, moves images of the same face closer together, while moving images of different faces further apart. Overall it tries to increase the ratio of the between class scatter to within class scatter. The difference in the space chosen by LDA and PCA (figure 16). In mathematical terms

$$J(T) = \frac{|T^T \cdot S_b \cdot T|}{|T^T \cdot S_w \cdot T|}$$

where, S_b and S_w are between-class and within-class scatter matrix.

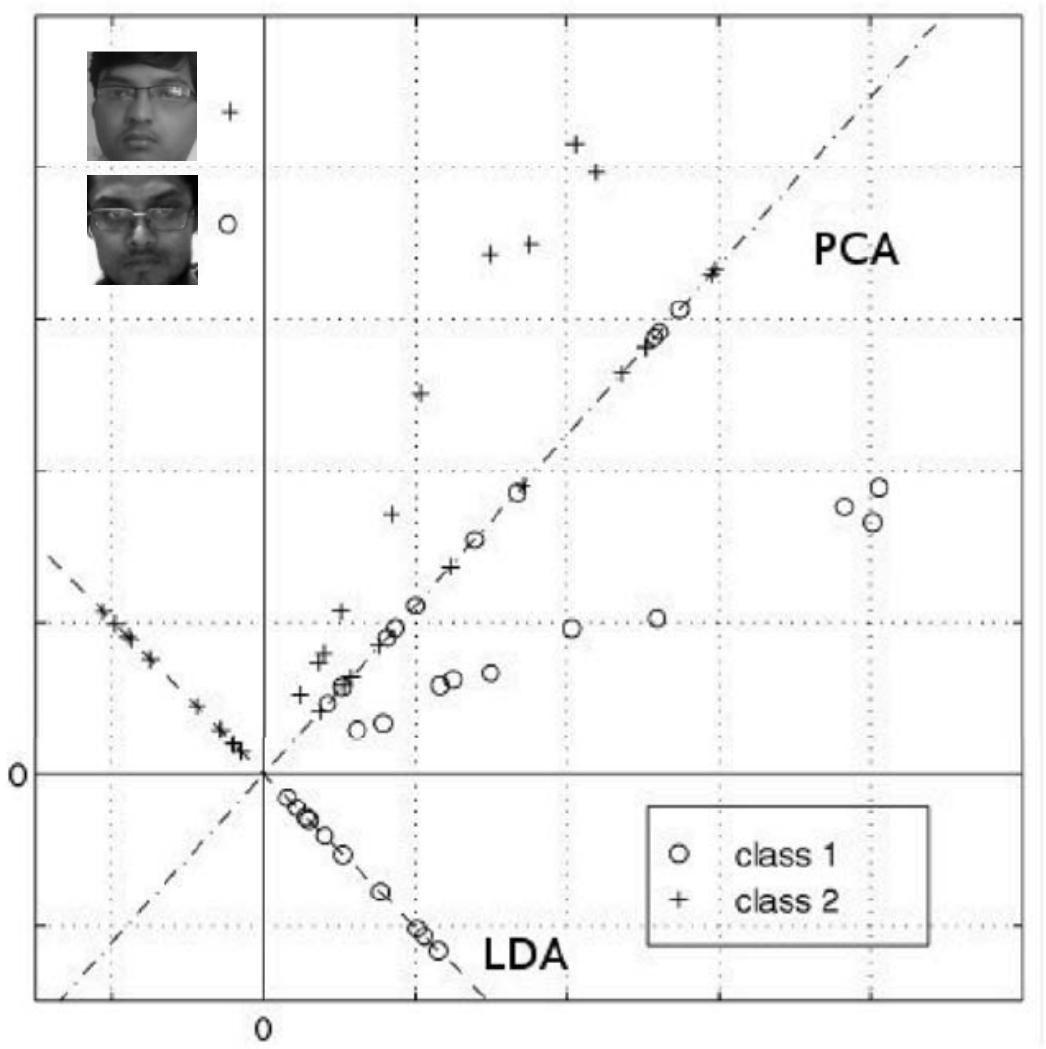


Figure 19: Difference in the space chosen by LDA and PCA

4.3.1 Training Operation in LDA

Let I be an image of size (Nx,Ny) pixels, then the training operation of PCA algorithm can be expressed in mathematical terms as

i. Convert the training image matrix I of size (Nx,Ny) pixels to the image vector Γ of size ($P \times 1$) where $P = Nx \times Ny$ (ie: the train image vector Γ is constructed by stacking each column of train image matrix I)

ii. Create a training set of training image vectors such that its size is $P \times Mt$ where Mt is the number of training images $\Gamma_P \times M_t = [\Gamma_1\ \Gamma_2\ .\ .\ .\ \Gamma_{Mt}\ ]$ where, Γ_i represents the image vector of i^{th} training images.

iii. Compute class mean face Ψ_{Ci} which is the arithmetic average of the training image vectors corresponding to the same individual at each pixel point for each class; and its size is (P ×1). In LDA, mean face images are also calculated for each face class; this is due to need for the calculation of each face classes inner variation. Hence, for each of C_L individuals having q_i training images in the database, the class mean face is given by.

$$\Psi_{C_i} = \frac{1}{q_i}\sum_{k=1}^{q_i} \Gamma_k \qquad i = 1, 2, \ldots, C_L \text{ (total no. of classes)}$$

iv. Compute arithmetic average (mean face Ψ) of the training image vectors at each pixel point given by:

$$\Psi_{P\times 1} = \frac{1}{M_t}\sum_{k=1}^{M_t} \Gamma_k$$

v. Obtain mean subtracted vector (Φ) by subtracting the class mean face from the each training image vector of that class as given below:

$\Phi i = \Gamma i - \Psi ci$

vi. Compute within class scatter matrix S_w for C_L individuals each having q_i training images. S_w represents the average scatter Σi of the image vectors of different individuals C_i around their respective class means Ψ_{Ci}.

$$S_{w\,(P\times P)} = \sum_{i=1}^{C_L} P(C_i)\,\Sigma_i$$

$$\text{where, } P(C_i) = \frac{1}{C_L}$$

Average scatter (Σi) = $E[\Phi_i \cdot \Phi_{Ti}] = E[(\Gamma_i - \Psi C_i) \cdot (\Gamma_i - \Psi C_i)^T]$

vii. Compute the "between class scatter matrix" S_b which represents the scatter of each class mean Ψ_{Ci} around the overall mean vector Ψ.

$$S_{b\,(P\times P)} = \sum_{i=1}^{C_L} P(C_i)(\Psi_{C_i} - \Psi) \cdot (\Psi_{C_i} - \Psi)^T$$

The objective is to maximize J(T); in other words finding an optimal projection W which maximizes between-class scatter and minimizes within class scatter.

$$W = arg\ \max_T (J(T)) \Rightarrow max(J(T)) = \left.\frac{|T^T \cdot S_b \cdot T|}{|T^T \cdot S_w \cdot T|}\right|_{T=W}$$

W can be obtained by solving the generalized eigenvalue problem

$$S_b \cdot W = S_w \cdot W \cdot \lambda_w$$

From the generalized eigenvalue equation, only C_{L-1} or less of the eigenvalues come out to be nonzero. Only the eigenvectors coming out with these nonzero eigenvalues can be used in forming the $W_P \times (C_{L-1})$ matrix.

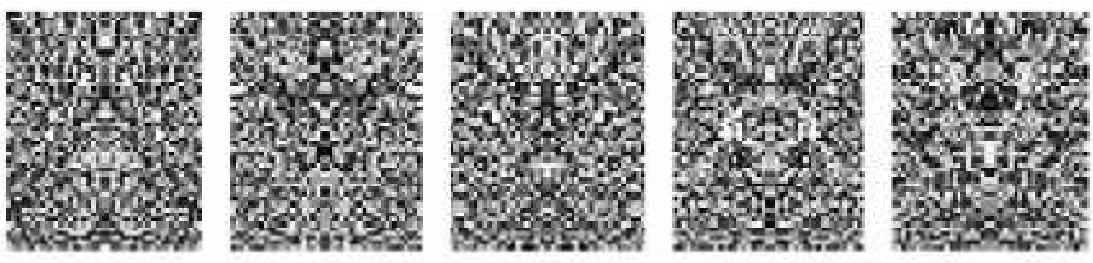


Figure 20: LDA bases

viii. Project training images vector to the classification space by the dot product of the optimum projection W and the image vector as follows:

$$g(\Phi_i)(C_{L-1}) \times 1 = W_T \cdot \Phi_i \qquad i = 1, 2, \ldots, M_t$$

4.3.2 Recognition Operation in LDA

The test image vector (Γ_T) is projected to the classification space in a similar way

$$g(\Phi_T)(C_{L-1}) \times 1 = W_T \cdot \Phi^T$$

Finally, the distance between the projections is calculated by the Euclidean distance between the training and test classification space projections. Distance measure is given by:

$$d_{T_i} = \|g(\Phi_T) - g(\Phi_i)\| = \sqrt{\sum_{k=1}^{C_L-1} \left(g_k(\Phi_T) - g_k(\Phi_i)\right)^2} \qquad i = 1, 2, \ldots, M_t$$

The test image is assumed to be in the class whose distance is the minimal among all other class distances.

4.3.3 Subspace LDA

Subspace LDA is a hybrid algorithm. It uses both PCA and LDA. However only LDA is used as the ultimate classifier; PCA is only used as a dimension reduction step. So in subspace LDA algorithm I initially create a PCA subspace using the training images as described above in the PCA section.

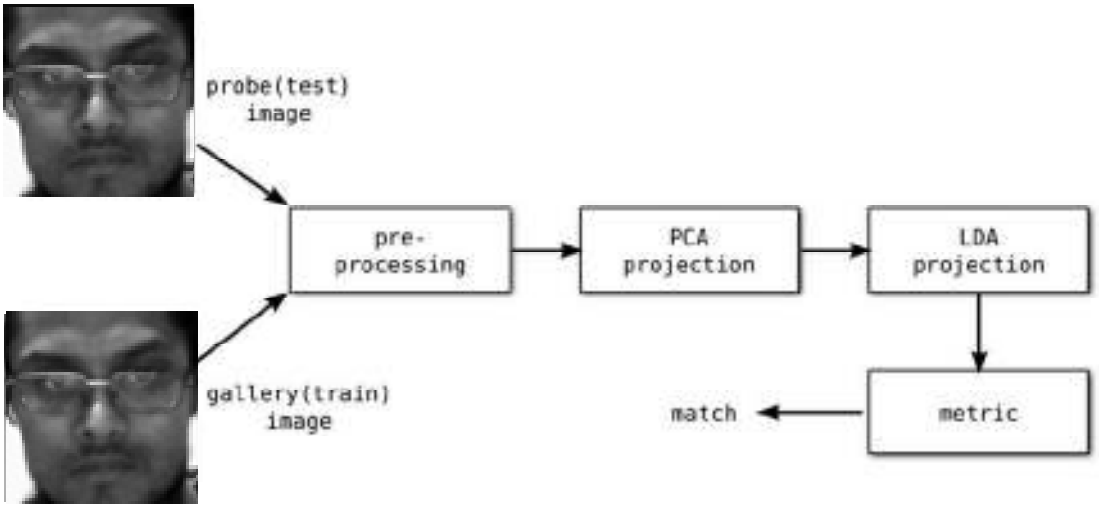


Figure 21: Basic block diagram of subspace LDA implementation for face recognition

All the training images are then projected into the PCA subspace. These projections are then input to the LDA classifier. LDA again creates a new subspace from these PCA projections as described above in the LDA section. The PCA projections are again projected into the LDA subspace. For classification, a test face is projected first into the PCA subspace followed by LDA subspace. Then a suitable distance metric can be used to compare the distance between the individual projections and classify the test face.

Figure 22: Subspace LDA bases

4.4 Eigenface

4.4.1 Overview

The need for improved information systems has become more conspicuous, since information is an essential element in decision making, and the world is generating increasing amounts of information in various forms with different degrees of complexity. One of the major problems in the design of modern information systems is automatic pattern recognition.

Recognition is regarded as a basic attribute of human beings, as well as other living organisms. A pattern is the description of an object. A human being is a very sophisticated information system, partly because he possesses a superior pattern recognition capability. According to the nature of the patterns to be recognized, recognition acts can be divided into two major types:

- **Recognition of concrete items.** This may be referred to as sensory recognition, which includes visual and aural pattern recognition. This recognition process involves the identification and classification of spatial and temporal patterns. Examples of spatial patterns are characters, fingerprints, physical objects, and images. Temporal patterns include speech waveforms, time series, electrocardiograms and target signatures.
- **Recognition of abstract items.** On the other hand, an old argument, or a solution to a problem can be recognized. This process involves the recognition of abstract items and can be termed conceptual recognition.

Recognition of concrete patterns by human beings may be considered as a psychophysiological problem which involves a relationship between a person and a physical stimulus. Human recognition is in reality a question of estimating the relative odds that the input data can be associated with one of a set of known statistical populations which depend on our past experience and which form the clues and the a priori information for recognition. Thus, the problem of pattern recognition may be regarded as one of discriminating the input data between populations via the search for features or invariant attributes among members of a population.

4.4.2 Pattern Classes and Patterns

Pattern recognition can be defined as the categorization of input data into identifiable classes via the extraction of significant features or attributes of the data from a background of irrelevant detail.

A **pattern class** is a category determined by some given common attributes or features. The features of a pattern class are the characterizing attributes common to all patterns belonging to that class. Such features are often referred to as intraset features. The features which represent the differences between pattern classes may be referred to as the interset features.

A **pattern** is the description of any member of a category representing a pattern class. For convenience, patterns are usually represented by a vector such as:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ .... \\ .... \\ x_n \end{bmatrix}$$

where each element x_j, represents a feature of that pattern. It is often useful to think of a pattern vector as a point in an n-dimensional Euclidean space.

4.4.3 Fundamental Problems in Pattern Recognition System Design

The design of an automatic pattern recognition systems generally involves several major problem areas:

- First of all, I have to deal with the representation of input data which can be measured from the objects to be recognized. This is the sensing problem. Each measured quantity describes a characteristic of the pattern or object. In other words, a pattern vector that describes the input data has to be formed. The pattern vectors contain all the measured information available about the patterns. The set of patterns belonging to the same class corresponds to an ensemble of points scattered within some region of the measurement space. A simple example of this is shown in Figure 20 for two pattern classes denoted by w_1 and w_2.

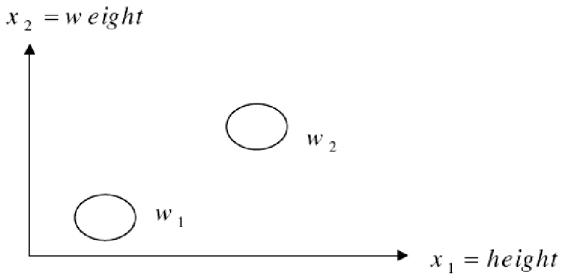


Figure 23: Two disjoint pattern classes. Each pattern is characterized by two measurements: height and weight. The pattern vector therefore is in the form of $x = \{x_1, x_2\}^T$.

- The second problem in pattern recognition concerns the extraction of characteristic features or attributes from the received input data and the reduction of the dimensionality of pattern vectors. This is often referred to as the pre-processing and the feature extraction problem. The elements of intraset features which are common to all pattern classes under consideration carry no discriminatory information and can be ignored. If a complete set of discriminatory features for each pattern class can be determined from the measured data, the recognition and classification of patterns will present little difficulty. Automatic recognition may be reduced to a simple matching process or a table look-up scheme. However, in most pattern recognition problems which arise in practice, the determination of a complete set of discriminatory features is extremely difficult, if not impossible.

- The third problem in pattern recognition system design involves the determination of the optimum decision procedures, which are needed in the identification and classification process. After the observed data from patterns to be recognized have been expressed in the form of pattern points or measurement vectors in the pattern space, I want the machine to decide to which pattern class these data belong. Let the system be capable of recognizing M different pattern classes. Then the pattern space can be considered as consisting of M regions, each of which encloses the pattern points of a class. The recognition problem can now be viewed as that of generating the decision boundaries which separate the M pattern classes on the basis of the observed measurement vectors. These decision boundaries are generally determined by decision functions.

4.4.4 Training And Learning

The decision functions can be generated in a variety of ways. When complete a priori knowledge about the patterns to be recognized is available, the decision function may be determined with precision on the basis of this information. When only qualitative knowledge about the patterns is available, reasonable guesses of the forms of the decision functions can be made. In this case the decision boundaries may be far from correct, and it is necessary to design the machine to achieve satisfactory performance through a sequence of adjustments.

The more general situation is that there exists little, if any, a priori knowledge about the patterns to be recognized. Under these circumstances pattern recognizing machines are best designed using a training or learning procedure. Arbitrary decision functions are initially assumed, and through a sequence of iterative training steps these decision functions are made to approach optimum or satisfactory forms.

It is important to keep in mind that learning or training takes place only during the design (or updating) phase of a pattern recognition system. Once acceptable results have been obtained with the training set of patterns, the system is applied to the task of actually performing recognition on samples drawn from the environment in which it is expected to operate. The quality of the recognition performance will be largely determined by how closely the training patterns resemble the actual data with which the system will be confronted during normal operation.

4.4.5 Supervised and Unsupervised Pattern Recognition

In most cases, representative patterns from each class under consideration are available. In these situations, supervised pattern recognition techniques are applicable. In a supervised learning environment, the system is taught to recognize patterns by means of various adaptive schemes. The essentials of this approach are a set of training patterns of known classification and the implementation of an appropriate learning procedure.

In some applications, only a set of training patterns of unknown classification may be available. In these situations, unsupervised pattern recognition techniques are applicable. As mentioned above, supervised pattern recognition is characterized by the fact that the correct classification of every training pattern is known. In the unsupervised case however, one is faced with the problem of actually learning the pattern classes present in the given data. This problem is also known as "learning without a teacher".

4.4.6 Outline of a Typical Pattern Recognition System

In Figure 21, functional block diagram of an adaptive pattern recognition system is shown. Although the distinction between optimum decision and pre-processing or feature extraction is not essential, the concept of functional breakdown provides a clear picture for the understanding of the pattern recognition problem.

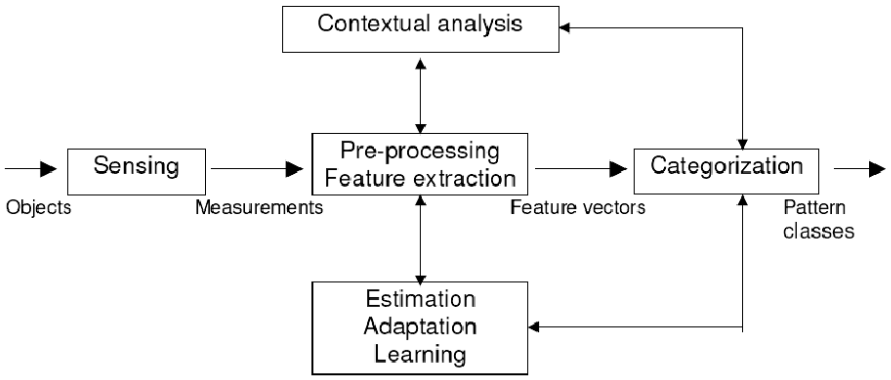


Figure 24: Functional block diagram of an adaptive pattern recognition system.

Correct recognition will depend on the amount of discriminating information contained in the measurements and the effective utilization of this information. In some applications, contextual information is indispensable in achieving accurate recognition. For instance, in the recognition of cursive handwritten characters and the classification of fingerprints, contextual information is extremely desirable. When I wish to design a pattern recognition system which is resistant to distortions, flexible under large pattern deviations, and capable of self-adjustment, I are confronted with the adaptation problem.

4.5 Face Recognition

Face recognition is a pattern recognition task performed specifically on faces. It can be described as classifying a face either "known" or "unknown", after comparing it with stored known individuals. It is also desirable to have a system that has the ability of learning to recognize unknown faces.

Computational models of face recognition must address several difficult problems. This difficulty arises from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces. Faces pose a particularly difficult

problem in this respect because all faces are similar to one another in that they contain the same set of features such as eyes, nose, mouth arranged in roughly the same manner.

4.5.1 Background and Related Work

Much of the work in computer recognition of faces has focused on detecting individual features such as the eyes, nose, mouth, and head outline, and defining a face model by the position, size, and relationships among these features.

Such approaches have proven difficult to extend to multiple views and have often been quite fragile, requiring a good initial guess to quide them. Research in human strategies of face recognition, moreover, has shown that individual features and their immediate relationships comprise an insufficient representation to account for the performance of adult human face identification. Nonetheless, this approach to face recognition remains the most popular one in the computer vision literature. Bledsoe was the first to attempt semi-automated face recognition with a hybrid human-computer system that classified faces on the basis of fiducial marks entered on photographs by hand. Parameters for the classification were normalized distances and ratios among points such as eye corners, mouth corners, nose tip, and chin point. Later work at Bell Labs developed a vector of up to 21 features, and recognized faces using standard pattern classification techniques.

Fischler and Elschlager, attempted to measure similar features automatically. They described a linear embedding algorithm that used local feature template matching and a global measure of fit to find and measure facial features. This template matching approach has been continued and improved by the recent work of Yuille and Cohen. Their strategy is based on deformable templates, which are parameterized models of the face and its features in which the parameter values are determined by interactions with the face image. Connectionist approaches to face identification seek to capture the configurational nature of the task. Kohonen and Kononen and Lehtio describe an associative network with a simple learning algorithm that can recognize face images and recall a face image from an incomplete or noisy version input to the network. Fleming and Cottrell extend these ideas using nonlinear units, training the system by backpropagation.

Others have approached automated face recognition by characterizing a face by a set of geometric parameters and performing pattern recognition based on the parameters. Kanade's face identification system was the first system in which all steps of the recognition process were automated, using a top-down control strategy directed by a generic model of expected feature characteristics. His system calculated a set of facial parameters from a single face image and used a pattern classification technique to match the face from a known set, a purely statistical approach depending primarily on local histogram analysis and absolute gray-scale values.

Recent work by Burt uses a smart sensing approach based on multiresolution template matching. This coarse to fine strategy uses a special purpose computer built to calculate multiresolution pyramid images quickly, and has been demonstrated identifying people in near real time.

4.5.2 Outline of a Typical Face Recognition System

In Figure 22, the outline of a typical face recognition system is given. This outline heavily carries the characteristics of a typical pattern recognition system that was presented in Figure 21.

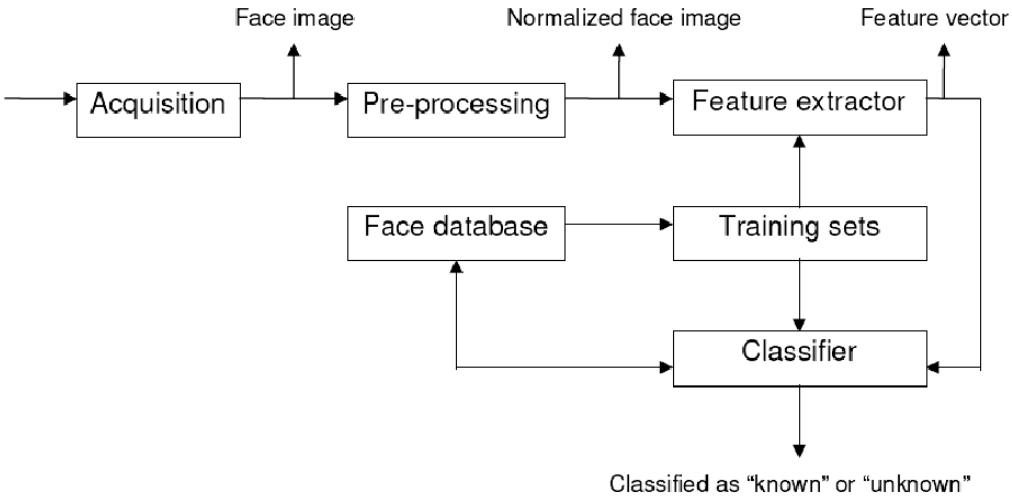


Figure 25: Outline of a typical face recognition system.

There are six main functional blocks, whose responsibilities are given below:

i. **The acquisition module.** This is the entry point of the face recognition process. It is the module where the face image under consideration is presented to the system. Inother words, the user is asked to present a face image to the face recognition system in this module. An acquisition module can request a face image from several different environments: The face image can be an image file that is located on a magnetic disk, it can be captured by a frame grabber or it can be scanned from paper with the help of a scanner.

ii. **The pre-processing module.** In this module, by means of early vision techniques, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system. Some or all of the following pre-processing steps may be implemented in a face recognition system:

a. **Image size normalization**. It is usually done to change the acquired image size to a default image size such as 128 x 128, on which the face recognition system operates. This is mostly encountered in systems where face images are treated as a whole like the one proposed in this thesis.

b. **Histogram equalization.** It is usually done on too dark or too bright images inorder to enhance image quality and to improve face recognition performance. It modifies the dynamic range (contrast range) of the image and as a result, some important facial features become more apparent.

c. **Median filtering.** For noisy images especially obtained from a camera or from a frame grabber, median filtering can clean the image without loosing information.

d. **High-pass filtering.** Feature extractors that are based on facial outlines, may benefit the results that are obtained from an edge detection scheme. High-pass filtering emphasizes the details of an image such as contours which can dramatically improve edge detection performance.

e. **Background removal.** Inorder to deal primarly with facial information itself, face background can be removed. This is especially important for face recognition systems where entire information contained in the image is used. It is obvious that, for background removal, the preprocessing module should be capable of determining the face outline.

f. **Translational and rotational normalizations.** In some cases, it is possible to work on a face image in which the head is somehow shifted or rotated. The head plays the key role in the determination of facial features. Especially for face recognition systems that are based on the frontal views of faces, it may be desirable that the preprocessing module determines and if possible, normalizes the shifts and rotations in the head position.

g. Illumination normalization. Face images taken under different illuminations can degrade recognition performance especially for face recognition systems based on the principal component analysis in which entire face information is used for recognition. A picture can be equivalently viewed as an array of reflectivities r(x). Thus, under a uniform illumination I, the corresponding picture is given by

$$F(x) = Ir(x)$$

The normalization comes in imposing a fixed level of illumination I_0 at a reference point x_0 on a picture. The normalized picture is given by

$$\Phi(X) = \frac{I_0 \Phi(X)}{I(X_0)}$$

iii. **The feature extraction module.** After performing some pre-processing (if necessary), the normalized face image is presented to the feature extraction module inorder to find the key features that are going to be used for classification. Inother words, this module is responsible for composing a feature vector that is well enough to represent the face image.

iv. **The classification module.** In this module, with the help of a pattern classifier, extracted features of the face image is compared with the ones stored in a face library (or face database). After doing this comparison, face image is classified as either known or unknown.

v. **Training set.** Training sets are used during the "learning phase" of the face recognition process. The feature extraction, and the classification modules adjust their parameters inorder to achieve optimum recognition performance by making use of training sets.

vi. **Face library or face database.** After being classified as "unknown", face images can be added to a library (or to a database) with their feature vectors for later comparisons. The classification module makes direct use of the face library.

4.5.3 Problems during Face Recognition

Due to the dynamic nature of face images, a face recognition system encounters various problems during the recognition process. It is possible to classify a face recognition system as either "robust" or "weak" based on its recognition performances under these circumstances. The objectives of a robust face recognition system is given below:

- **Scale invariance.** The same face can be presented to the system at different scales as shown in Figure 2.4-b. This may happen due to the focal distance

between the face and the camera. As this distance gets closer, the face image gets bigger.

- **Shift invariance.** The same face can be presented to the system at different perspectives and orientations as shown in Figure 2.4-c. For instance, face images of the same person could be taken from frontal and profile views. Besides, head orientation may change due to translations and rotations.

- **Illumination invariance.** Face images of the same person can be taken under different illumination conditions such as, the position and the strength of the light source can be modified like the ones shown in Figure 2.4-d.

- **Emotional expression and detail invariance.** Face images of the same person can differ in expressions when smiling or laughing. Also, like the ones shown in Figure 2.4-e, some details such as dark glasses, beards or moustaches can be present.

- **Noise invariance.** A robust face recognition system should be insensitive to noise generated by frame grabbers or cameras. Also, it should function under partially occluded images.

A robust face recognition system should be capable of classifying a face image as "known" under even above conditions, if it has already been stored in the face database.

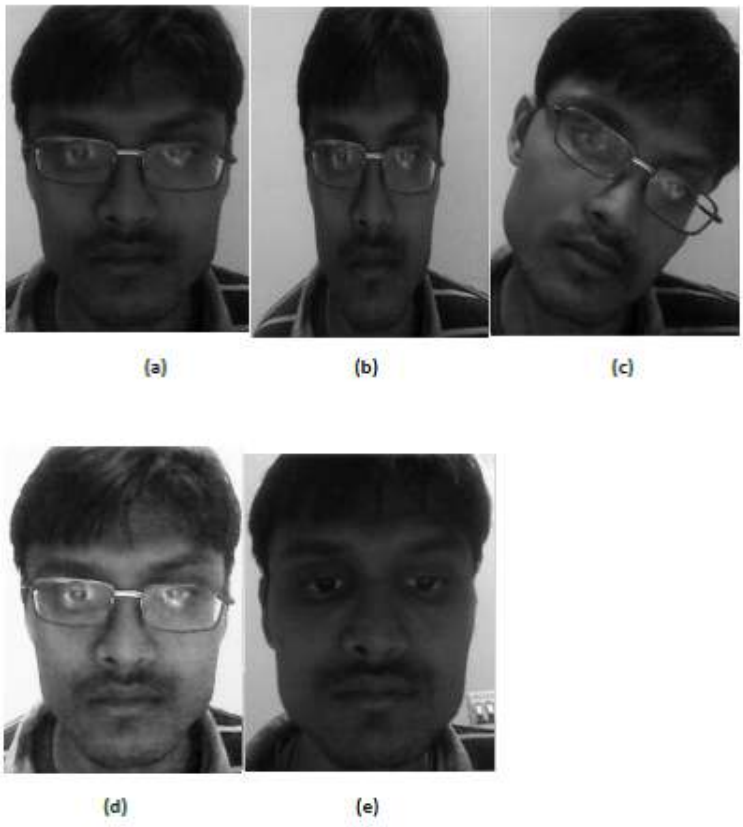


Figure 26: (a) Original face image. (b) Scale variance. (c) Orientation variance. (d) Illumination variance. (e) Presence of details.

4.5.4 Feature based Face Recognition

It was mentioned before that, there were two basic approaches to the face recognition problem: Feature based face recognition and principal component analysis methods. Although feature based face recognition can be divided into two different categories, based on frontal views and profile silhouettes, they share some common properties and I will treat them as a whole. In this section, basic principles of feature based face recognition from frontal views are presented.

4.5.5 Introduction to Feature based Face Recognition

The first step of human face identification is to extract the features from facial images. In the area of feature selection, the question has been addressed in studies of cue salience in which discrete features such as the eyes, mouth, chin and nose have been found important cues for discrimination and recognition of faces. After knowing what the effective features are for face recognition, some methods should be utilized to get contours of eyes, eyebrows, mouth, nose, and face. For different facial contours, different models should

be used to extract them from the original portrait. Because the shapes of eyes and mouth are similar to some geometric figures, they can be extracted in terms of the deformable template model. The other facial features such as eyebrows, nose and face are so variable that they have to be extracted by the active contour model. These two models can be illustrated in the following:

- **Deformable template model.** The deformable templates are specified by a set of parameters which uses a priori knowledge about the expected shape of the features to guide the contour deformation process. The templates are flexible enough to change their size and other parameter values, so as to match themselves to the data. The final values of these parameters can be used to describe the features. This method works well regardless of variations in scale, tilt, and rotations of the head. Variations of the parameters should allow the template to fit any normal instance of the feature. The deformable templates interact with the image in a dynamic manner. An energy function is defined which contains terms attracting the template to salient features such as peaks and valleys in the image intensity, edges and intensity itself. The minima of the energy function corresponds to the best fit with the image. The parameters of the template are then updated by steepest descent.

- **Active contour model (Snake).** The active contour or snake is an energy minimizing spline guided by external constraint forces and influenced by image forces that pull it toward features such as lines and edges. Snakes lock onto nearby edges, localizing them accurately. Because the snake is an energy minimizing spline, energy functions whose local minima comprise the set of alternative solutions to higher level processes should be designed. Selection of an answer from this set is accomplished by the addition of energy terms that push the model toward the desired solution. The result is an active model that falls into the desired solution when placed near it. In the active contour model issues such as the connectivity of the contours and the presence of corners affect the energy function and hence the detailed structure of the locally optimal contour. These issues can be resolved by very high-level computations.

4.5.6 Effective Feature Selection

Before mentioning the facial feature extraction procedures, I have the following two considerations:

- The picture-taking environment must be fixed inorder to get a good snapshot.
- Effective features that can be used to identify a face efficiently should be known.

Despite the marked similarity of faces as spatial patterns I are able to differentiate and remember a potentially unlimited number of faces. With sufficient familiarity, the faces of any two persons can be discriminated. The skill depends on the ability to extract invariant structural information from the transient situation of a face, such as changing hairstyles, emotional expression, and facial motion effect.

Features are the basic elements for object recognition. Therefore, to identify a face, I need to know what features are used effectively in the face recognition process. Because the variance of each feature associated with the face recognition process is relatively large, the features are classified into three major types:

- **First-order features values.** Discrete features such as eyes, eyebrows, mouth, chin, and nose, which have been found to be important in face identification and are specified without reference to other facial features, are called first-order features. Important first-order features are given in Table 2.1.

- **Second-order features values.** Another configural set of features which characterize the spatial relationships between the positions of the first-order features and information about the shape of the face are called second-order features. Important second-order features are given in Table 2.2. Second order features that are related to nose, if nose is noticeable are given in Table 2.3.

- **Higher-order feature values.** There are also higher-level features whose values depend on a complex set of feature values. For instance, age might be a function of hair coverage, hair color, skin tension, presence of wrinkles and age spots, forehead height which changes because of receding hairline, and so on. Variability such as emotional expression or skin tension exists in the higher-order features and the complexity, which is the function of first-order and second-order features, is very difficult to predict. Permanent information belonging to the higher-order features can not be found simply by using first and

second-order features. For a robust face recognition system, features that are invariant to the changes of the picture taking environment should be used. Thus, these features may contain merely first-order and second-order ones. These effective feature values cover almost all the obtainable information from the portrait. They are sufficient for the face recognition process. The feature values of the second-order are more important than those of the first-order and they are dominant in the feature vector. Before mentioning the facial feature extraction process, it is necessary to deal with two preprocessing steps:

- **Threshold assignment**. Brightness threshold should be known inorder to discriminate the feature and other areas of the face. Generally, different thresholds are used for eyebrows, eyes, mouth, nose, and face according to the brightness of the picture.

- **Rough Contour Estimation Routine (RCER).** The left eyebrow is the first feature that is to be extracted. The first step is to estimate the rough contour of the left eyebrow and find the contour points. Generally, the position of the left eyebrow is about one-fourth of the facial width. Having this a priori information, the coarse position of the left eyebrow can be found and its rough contour can be captured. Once the rough contour of the left eyebrow is established, the rough contours of other facial features such as left eye, right eyebrow, mouth or nose can be estimated by RCER. After the rough contour is obtained, its precise contour will be extracted by the deformable template model or the active contour model.

Measurement	Facial Location
Area, Angle	left eyebrow right eyebrow left eye right eye mouth face
Distance	Length of left eyebrow Length of right eyebrow Length of left eye Length of right eye Length of mouth Length of face Height of face

Table 1 First Order Features

Measurement	Facial Location
Distance	left eyebrow <-> right eyebrow left eye <-> right eye left eyebrow <-> left eye right eyebrow <-> right eye left eyebrow <-> mouth right eyebrow <-> mouth left eye <-> mouth right eye <-> mouth eyebrow <-> side of face eye <-> side of face mouth <-> side of face mouth <-> lower part of face
Angle	Left eyebrow – left eye – left eyebrow Right eyebrow – right eye – right eyebrow Left eye – left eyebrow – left eye Right eye – right eyebrow – right eye Left eyebrow – mouth – right eyebrow Left eye – mouth – right eye Left eyebrow – left eye – mouth Right eyebrow – right eye – mouth

Table 2 Second Order Features

Measurement	Facial Location
Distance	Left nose <-> right nose Left eyebrow <-> left nose Right eyebrow <-> right nose Left eye <-> left nose Right eye <-> right nose Left nose <-> mouth Right nose <-> mouth
Angle	Left eyebrow – center of nose – right eyebrow Left eye – center of nose – right eye Left nose – mouth – right nose Left eyebrow – left eye – left nose Right eyebrow – right eye – right nose

Table 3 Features related to nose, if nose is noticeable.

Chapter 5

5 FRVS : IMPLEMENTATION AND ANALYSIS

5.1 Requirement

i. A Facial Recognition system that will verify and process real time (directly from camera) and stored videos.

ii. It must include database storage and retrieval.

iii. Database can store large amount of information in efficient manner.

iv. Results should be fast and accurate.

v. System stable enough to work 24x7 (system capable to recover after crash).

vi. System can have more than one input and output interface(s) working in parallel.

vii. It must be capable of integrating itself with current surveillance systems (like CCTV).

viii. System should be able to work even in minimal light conditions (minimum room light).

ix. Face recognition should work with different facial gestures.

x. Facial changes such as mustaches and beard should not affect functionality of the system.

xi. System should recognize all the faces (if more than one face exist) in a frame.

xii. Side posture of face should also be considered.

xiii. A log should be maintained of all activities of every interfacing unit.

xiv. In case of Occlusion(s) preventive measures should be taken.

xv. If a detected face doesn't exist then a option should be displayed to add it in database.

5.2 Requirement Analysis (FRVS)

Requirement : A Facial Recognition system that will verify and process real time (directly from camera) and stored videos.

Analysis : Our first requirement is to provide a system that can perform following operations precisely i.e. take input from an imaging device such as video camera, CCTVs etc., then it can detect the presence of any human face in the video or the particular video frame, then it should search for the detected face from the provided database and then the output must be the information of that detected face.

For this requirement our system must be capable of handling all types of cameras or the imaging devices that are being used i.e. it must be updated with the current technologies that are used for video acquisition and for this it is required that our system must be equipped with updated drivers. A proper networking should be there among the devices (used for video acquisition) because at the crowded place one device cannot cover all the angels so there must be more than one device to be used. In case of stored videos, format compatibility issue may occur for which I must first analyze the formats in which a video is being stored by the devices. Another issue that can arrive is quality of videos, for this I must be prepared with a module or a sub system that is capable video enhancement. It must be ensured that there is a proper detection because this is the initial phase and if this goes wrong then may result in wrong recognition too therefore system must take the input when maximum features of face are visible that will result in fast and accurate recognition.

Requirement : It must include database storage that is capable of storing large amount of information and information retrieval.

Analysis : Our second requirement is to provide a system that can interact with a database system i.e. capable of performing operations like storing information and its retrieval.

For this requirement first I need a huge amount of space where I can store our data. Our data must be well normalized so as to remove the redundancy of data. I must choose an efficient database engine that ensures stability, easy maintenance, and security. Since our database will be very large as it contains the training images, information related to those images therefore I must be ready with a backup plan that will be helpful at the time of any system crash and our backup database must be able to synchronize with the original database. Our database must contain the frontal view of face, side pose of face and the information related to the face separately that will result in fast search and information retrieval. The issues related to the database connectivity with our system must be checked before the installation.

Requirement : Results should be fast and accurate.

Analysis : Results provided by the system must be fast and as well as accurate too.

For this requirement the database must be well normalized in order to remove the redundancy of data that will result in faster and accurate search. There are certain standard databases for storing the training databases such as FERET (FAce REcognition Technology) database and ORL (Oracle Research Laboratory) Face Database are used as de-facto standards that allows the storing of images in many ways but did not deal

with the storage of information related to the image. Thus I have to design data base in such aa manner that it handle information also efficiently of the faces. To increase the detection and recognition rate I had to look for the fast algorithms or can achieve good rate by making changes to any algorithm. For example Eigenface algorithm and Gabor wavelet algorithm can be speed up by using the Fast-Fourier Transformations. Along with the above steps I also need good processing power because detection and recognition process requires thousand of calculations therefore I need good processors and RAM to make our system fast, that results in increase in hardware cost.

Requirement : System stable enough to work 24x7 (system capable to recover after crash).

Analysis : Our system must be stable because any unstable system can cause data loss and if there is any condition like this then it must recover itself or can be recovered by someone very soon.

For this requirement it must be ensured that the place where system or devices are being installed that place must be well equipped with power backup plans as it is seen that power-cut plays important role in system crash. If system crashes by itself then must be able to recover itself from crash, and at this moment I need a secondary or a backup database and system that starts working automatically without any delay and continues te work . When system is recovered then its database must get synchronized with the secondary database so that no information will be lost. In the case if there is some another reason for the system crash then it must be capable of informing aor alarms its failure ,in that case a technician will needed to recover the system that requires some more men to handle the system at the place of installation.

Requirement : System can have more than one input and output interface(s) working in parallel.

Analysis : System can be able to work with more than one input and output interface because as mentioned earlier that in a crowdy place it will be impossible to work only with one input and output interface i.e. system could have various input and output interfaces (video/image acquisition devices) including multiple cameras, video input from network, etc. These inputs from the interfaces can be handled in parallel by providing each slot in the software for interacting with that interface. The database of FRVS is central so many copies of software could work simultaneously with no problem.

For this requirement the first issue that can occur is the synchronization problem with the database because I are providing input to database from more than one interface therefore there may be a chance of data redundancy that must be removed and parallel many input are coming for the detection and recognition process therefore it must be handled in a ordered way or else can result into a unstable or a slow system. The output screen must show the different places or locations that will prevent data redundancy, for this the video-acquisition device must be installed at various locations. Now in order to control all the devices there must be a strong central system that can able to manage data coming from the devices in a order and this central system need some more hardware, that will increase hardware cost and more man power will be required to handle the central system

Requirement : It must be capable of integrating itself with current surveillance systems (like CCTV).

Analysis : Our system must be updated according to the current surveillance systems i.e. must be updated with the drivers and must be compatible.

For this requirement our system must be platform independent because in that case I don't have to change our system according to the devices as it is not possible to use same devices for surveillance at all the places. There must be common standard at which both system and the device should work like the format of video. It must also have the backward compatibility so that it can work on the old devices too (which is not so much necessary). There must be a common secure gateway for interaction between system and database i.e. there must be secure transfer of data .The uninstallation process must also be considered because it is sometimes easy to set or install anything but difficult to remove or uninstall it. This system requires a good maintenance hence increasing the maintenance cost. The main problem that could arise is of bad video quality required for the detection and recognition purpose. Thus integration could be done with ease with existing systems like CCTV with techniques like image enhancement, etc.

Requirement : System should be able to work even in minimal lightening conditions (minimum room light).

Analysis : The performance of the system is degraded for some daylight photos or photos taken in very dim light conditions. This is due to the combination of two factors. First, the variety of possible shades in under daylight and dim-light conditions is a challenge Second, the skin color can change with the light. These problems should be taken care of during the face detection process. But it is necessary for the system that there must be some light at time while it is performing detection and recognition operation.

For this requirement I had to look for new and effective algorithm like Adaboost algorithm . Here image enhancement algorithms such as Contrast Stretching Algorithm, Gray Scale Mapping Algorithm, Power Law Transformation Can also be used simultaneously in order to enhance our image or video quality. The another way is that I can work on the features such as curve of face, eyes etc. (that are visible clearly) and not on the features distance where I use to find the distance using the Euclidean Distance and Mahalanobis Distance. But it will be good for the performance of the system if some amount of light is available because it is not possible for system to work in complete dark conditions, then in such case I had to opt for good cameras that has capability of night vision which will be more costlier.

Requirement : Face recognition should work with different facial gestures.

Analysis : Different facial gestures have been a problem with face recognition techniques. For this algorithm Eigenface would be used which converts different images with different facial gestures into one Eigenface image which is the average of all the images.

For this requirement first I have to look for the available effective algorithms already available such as Eigenface Algorithm and Gabor Wavelet Algorithm and I have to analyze their capability for working with different facial gestures. If they are capable then I will use them else search for another new algorithms with this capability. The main aspect of this requirement is the training database. Our training images in database must contain facial gestures also so that it will be easier for the system to locate the face in database and

hence making the system fast and efficient. This requirement also needs more filtering of the images in the database.

Requirement : Facial changes such as mustaches and beard should not affect functionality of the system

Analysis : Moustaches and beards or other kinds of facial ornaments provide a great hindrance in the face recognition process. Measures will be taken with the use of Eigenface algorithm in which the faces are searched according to the facial features that don't change over a period.

For this requirement first I have to look for the available effective algorithms already available such as Eigenface Algorithm and Gabor Wavelet Algorithm and I have to analyze their capability for working with different facial changes. If they are capable then I will use them else search for another new algorithms with this capability. The main aspect of this requirement is also the training database (like in above condition). Our training images in database must contain some images with also so that it will be easier for the system to locate the face in database and hence making the system fast and efficient. This requirement also needs more filtering of the images in the database.

Requirement : System should recognize all the faces (if more than one face exist) in a frame and In case of Occlusion(s) preventive measures should be taken.

Analysis : System must be capable of detecting all the faces that appears in the video frame. The video for this must be of fair quality and if not then image enhancement techniques must be used to provide a better quality video. Most algorithms perform poorly if a face is partially hidden by another object. This happens because the correlation in the hidden part will be lower, as it does not correspond to a face, providing a total lower correlation value.

For this requirement, in order to recognize all face the main hinderence is the Background details and removing the background information is a problem in itself. It can be used by using the image processing techniques such as applying Logical operations on the image like AND, OR or NOT operations that are helpful in removing the background or the excess information like textures from the image that can increase calculations and operations that can slow down the system performance. But this can increase the computational cost as firstly removing the background information and then detecting and recognizing many numbers of faces.

Requirement : Side posture of face should also be considered.

Analysis : In present face recognition systems it is required that the training faces must be in frontal position and as well as must be centered due to which the systems are efficient to recognize face when they are at the same position i.e. are centered and in frontal positions but their preciseness lags when side view recognition is to be done. Therefore Different side postures have been a problem with face recognition techniques. For this algorithm Eigenface would be used which converts different images with different facial gestures into one Eigenface image which is the average of all the images.

For this requirement first I have to look for the available effective algorithms already available such as Eigenface Algorithm and Gabor Wavelet Algorithm and I have to analyze their capability for working with

different side postures of human face. If they are capable then I will use them else search for another new algorithms with this capability. The main aspect of this requirement is the training database. Our training images in database must contain facial gestures also so that it will be easier for the system to locate the face in database and hence making the system fast and efficient. This requirement also needs more filtering of the images in the database. I can also make different part in the database that will only store the side postures of the human face thus will increase the efficiency in terms of searching in database.

Requirement : A log should be maintained of all activities of every interfacing unit.

Analysis : A log is maintained about the whole face recognition process. The log contains details about all the processes involved in face recognition through which user can easily find out the error in a particular section of database

For this requirement it is needed to maintain a log file (which will be created by the system itself) that will contain all the activities performed by the system during the face detection and recognition process of the whole day as <ACTIVITY PERFORMED_TIME_DATE> and saved as <LOG_DATE>. It contains the activities such as how many faces are detected, out of them how many are recognized from the database, if any new information or the face is added or any information is edited etc.

Requirement : If a detected face doesn't exist then a option should be displayed to add it in database.

Analysis : If ,after the face detection during the face recognition process, the recognition fails then a option is given to the end user to manually insert the face and give it an relevant identity or to discard it.

To fulfill this requirement a module is needed that gives the option to the controller to save any face and the related information in the database if system does not able to find that face in database (that face is not present in database) as <Anonymous_datetime>. There may be a case when I get large number of faces that does not resides in our database instantly then those faces must stored in a temporary database and then can be processed later.

Requirement : System should be able to distinguish between similar shapes.

Analysis : In order to match a picture of an object to a similar picture in a data base I need to tackle the issue of **similarity**, a concept that it is quite difficult to quantify. Consider, for example, the three shapes in Figure

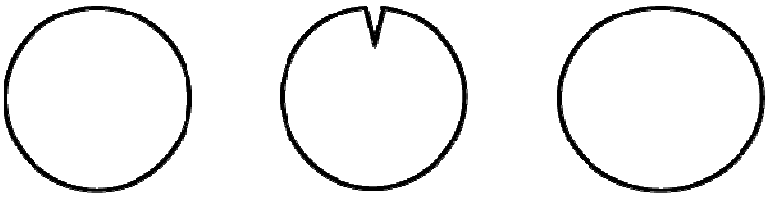


Figure 27: Illustration used to explain the difference between humanly perceived and mathematical similarity.

This is the case that may occur during the face recognition of twins that will increase the computational cost as this requirement will need more number of calculations and operations to be performed while detection and recognition of human face.

Requirement : Backup of database.

Analysis : Data loss or the information loss in such systems is equivalent to the damage because database is backbone of such systems.

For this requirement as I know that our database is very large therefore I must choose an efficient database engine that ensures stability, easy maintenance, and security. Our database contains huge amount of training images, information related to those images therefore I must be ready with a backup plan that will be helpful at the time of any system crash and our backup database must be able to synchronize with the original database. I can use technologies such RAID for the data backup.

5.3 FRVS System

This project aims at firstly detecting the human faces from a video source or a still image using face detection algorithm and next to recognize that detected faces from the training database. The algorithms used for these purpose are: Voila Jones Face detection algorithm as the name suggest for face detection purpose and Eigenfaces Face Recognition algorithm for the face recognition phase.

The project works on still image, video file, webcam as input and detects as well as recognizes faces in it.

Firstly, I started working on the face detection module in the still image. I used Voila Jones algorithm for the detection process. The algorithm is explained previously. The face was detected in the image and the object containing detected face was captured and coordinates of the detected face were taken and a rectangle was drawn specifying the boundary of the detected face.

The detected portion was then stored as a image in JPEG format.

Secondly, I worked the same procedure on video stream, Since the video stream is a collection of different frames, so this procedure for a single image can be used for the incoming frames of the videos. Now the face detection Voila Jones Algorithm was implemented on the frames and the results were good. The face was detected but there as false acceptance also, i.e. some regions in video stream were also detected which did not contain the face.

But since the next phase was recognition phase so the previous problem was not a big one as the recognition would have not effect but obviously a overhead was there in terms of time.

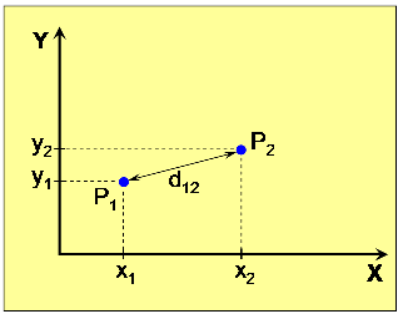
The faces were detected in the video file successfully, now further I extended the input method by taking the input from the webcam and successfully detected the faces from the webcam as input.

The face database for the recognition phase includes five 100*100 size face images of each person.

Now in the recognition phase I have used Eigenfaces face recognition algorithm for recognizing the faces.

Here's how recognition works: given example face images for each of several people, plus an unknown face image to recognize,

i. Compute a "distance" between the new image and each of the example faces
ii. Select the example image that's closest to the new one as the most likely known person
iii. If the distance to that face image is above a threshold, "recognize" the image as that person, otherwise, classify the face as an "unknown" person



There are two steps in recognition phase:

i. Training.
ii. Recognition.

Training: Training refers to the process of making mean face i.e. eigenfaces from the database. A set is created which includes the eigenvalues and eigenvectors and are used in recognition purpose.

The xml file is of format in which the values are stored:

<eigenValMat> >
<rows>1</rows>
<cols>2</cols>
<dt>f</dt>
<data>
14279064. 9614034.</data></eigenValMat>

Recognition: Testing refers to the process of matching the image with the eigenface(mean face) and coming out with closest match that is possible.

Eigenface uses the training images to "learn" a face model. This face model is created by applying a method called Principal Components Analysis (PCA) to reduce the "dimensionality" of these images. Eigenface defines image dimensionality as the number of pixels in an image.

The lower dimensionality representation that eigenface finds during the learning phase is called a subspace. In the recognition phase, it reduces the dimensionality of the input image by "projecting" it onto the subspace it found during learning. "Projecting onto a subspace" means finding the closest point in that subspace. After the unknown face image has been projected, eigenface calculates the distance between it and each training image. Its output is a pointer to the closest training image. You can then look up which person that was to see whom eigenface has identified.

5.4 Implementation

- Load Cascaded Filter (Like profile face, full body etc. based on Haar-like-feature).
- Allocate Memory storage for storing frames.
- Capture Input from avi/jpg file.
- If capture SUCESSFUL then
- Retrieve frames (video) one by one else store the image in 1 frame.
- Detect Faces using cvHaarDetectObjects function.
- Repeat until all faces are iterated.
- Create a new window.
- Create a temp Image.
- Find the coordinates of corner of rectangle which decides location of the face.
- Set Image ROI (Region of Interest) based on this rectangle.
- Extract ROI of parent image to temp image.
- Display temp Image in window.

The Algorithm was used for FACE detection – face was the object (Viola-Jones).
The cascade contained 32 classifiers.
Building the cascade consumed 80000 operations.
The cascade results – rapid average detection times.
Training Data base contained 507 faces and 75 million sub windows.
Computation consumed ~270 microprocessor instructions per sub-window. (according to Intel OpenCV)
(We had used classifiers, faces and subwindows provided in Intel OpenCV library).
Incremental Software development approach will be used as the primary objective is to reduce inherent project risk by breaking a project into smaller segments and providing more ease-of-change during the development process.

5.5 Steps for Building the Face Database using Webcam

i. Install OpenCV library.
ii. Start Visual Studio and open the project.
iii. Build the project.
iv. Run the project.
v. Images are taken using webcam as source.
vi. The samples are created at the same path as the executable.
vii. Store the samples in database/person folder and give proper numbering.

5.6 Steps Required for Executing the FRVS Code

i. Install OpenCV library.
ii. Start Visual Studio and open the project.
iii. For running the code for image edit the run.bat batch file and enter the image name along with the extension.
iv. For running the code for the video edit the run.bat batch file and enter the video name along with the extension.
v. For running the code with the webcam option edit the run.bat batch file and leave that input file field empty.
vi. Run the run.bat file.
vii. Output can be seen on the screen.

5.7 Performance and Testing

The two face recognition modules were tested on some standard face databases and the performance of the two face extraction modules was tested on a video shot by webcam at 640* 480 resolution. Testing was performed on a Dell laptop with the following specifications :

Processor : Intel(R) Core(TM)2 Duo CPU T7300 @ 2.00GHz
Memory : 1GB
Operating System : Windows Vista Home Basic
C++ compiler : Visual C++

The nature of several parameters were studied in order to assess the performance of FRVS when it is implemented and used in a real world scenario. All the performance related data presented here relate to natural lighting condition, almost no facial occlusion, large amount of facial expression variation and very small out of plane rotation. The face detection is done by Voila Jones algorithm.

Result of face Detection phase:
44.664698 1 face
63.982867 2 faces

There was false detection that can be seen in video frames also. False detection area included the non faces areas which were overhead for the recognition phase.

5.8 Training Dataset

For the training of the face recognition, faces of size 100*100 were used. The face used by Eigenfaces in their training was collected for the training purpose. For the training purpose, 10 face samples of the each person were used.

Training time was approx 564.23412 ms.

5.9 Implementation of the Recognition Process

The frames from the webcam were scaled down to 100*100 resolutions before the recognition process. The scaling factor increases the size of the scanning sub window and is used to detect the faces in the images at different scales. Then the recognition was done using the Eigenfaces face recognition algorithm and correspondingly the result was displayed in the recognition window. The respective result timings of the face recognition were displayed on the console part and can be viewed there.

5.10 Results of Recognition

The recognition on the webcam images are better than the images selected at random because the non-face source images used for the training was of the environment on which the detector would be run. The recognition of other images are good for the closeup images, but for other images there can be many false detections. This can be improved by training on a larger training set and training for larger stage.

The recognition time was:

Starting time:1782.035426

326.883375 1 face
346.612558 1 face
620.368499 2 faces
626.355375 2 faces

Chapter 7

7. CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

7.2 Future Work

7.2.1 OpenMPI Based Version of FRVS that utilizes Distributed Processing

7.2.2 Flowgraph Based GUI for FRVS

7.2.3 Use of Motion Tracking to Improve the Performance

6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

For the two Face Extraction (FE) algorithms in FRVS, I concluded that although Viola-Jones based approach can achieve high level of accuracy in face detection, its high detection time does not make it fit for real-time operation. Despite the very large training time (which is not a critical factor for realtime operation) of AdaBoost algorithm, it offers good performance with very small detection time (< 0:1 seconds).

Instead of using two different algorithms (Viola-Jones based approach and AdaBoost) for the task of realtime face extraction the implementation of three instances Adaboost algorithm, trained with different dataset, can provide appreciable real time performance. Results from the three instances can be considered as satisfactory.

Processing each frame in a video (30 fps) by AdaBoost algorithm to detect faces requires large amount of processing power and resources to achieve realtime operation. Use of algorithms like camshift, optical ow (for only detecting motion), etc can reduce the number of frames to be processed by the AdaBoost. These algorithm analyzes each video frame to detect presence of new faces in the video.

The two Face Recognition(FR) algorithms working in unison results in a hybrid approach to face recognition. Research has shown that visual processing pathway of human brain (an example for artificial face recognition systems) also employs a hybrid approach for face recognition. Both algorithms have high accuracy for recognition of faces in two different regions (there exists some overlap between the two regions) of the face space.

Subspace LDA performs well even for low resolution images but the performance degrades for images containing illumination variation and out of plane rotation of faces. However, Kepenekci algorithm is an illumination invariant technique having appreciable level of performance only for high resolution images.

An arbitrator function can be designed to assign appropriate weights (based on the nature of input image and face space region that an algorithm recognizes with high accuracy) to the result of recognition by each face recognition algorithm. A well chosen arbitrator function can improve the overall accuracy of the face recognition process. The absence of quantitative techniques to precisely define the nature of input images present a hurdle to the design of an optimal arbitrator function.

Nonlinearity and distortions introducted by image capture device can drastically reduce the accuracy of feature based techniques. Such errors have virtually no impact on the performance of holistic techniques like Subspace LDA. Hence, appropriate preprocessing should be performed before supplying the train/probe images. Image registration (an important image preprocessing technique) ensures that overall face structure and facial features location of train/probe images have least amount of variation. Also, a controlled environment(that can control parameters like illumination, possible poses of human face, etc) can improve the accuracy of FR algorithms. Controlling the environment

parameters (like illumination, possible facial poses, image resolution, etc) provides a proven technique for improving the accuracy of face recognition procedure. Preprocessing (histogram equalization, image registration, etc) of train/probe images can also ensure desirable level of accuracy.

6.2 Future Work

6.2.1 OpenMPI Based Version of FRVS that Utilizes Distributed Processing

OpenMPI1 can be used to implement a parallel processing version of FRVS that distributes task of face extraction/recognition to a number of computers. The results are aggregated by the MASTER node.

6.2.2 Flowgraph Based GUI for FRVS

A user interface that allows formation of custom processing pathway by adding components that implement different stages of FRVS. This type of architecture is also being developed by gnuradio companion2 for the gnuradio project. Present implementation of FRVS involves integration of different modules (modules belonging different stages) using a simple Visual C++ program. Such intuitive user interface would hide the complexity by allowing users to form processing pathway without requiring users to have the knowledge of interface used by each module for communication.

6.2.3 Use of Motion Tracking to Improve the Performance

Present implementation of FRVS does not use the information gained from motion tracking algorithms like camshift. It processes each frame of the video and the relationship between consecutive frames in not used during the face extraction/recognition procedure. Use of motion tracking algorithms can reduce the number of frames to be processed and hence improve the overall performance of FRVS.

BIBLIOGRAPHY

[1] Java Bluetooth Discussions and Tutorials. (2006). Retrieved July 14, 2007 from http://www.jsr82.com.

[2] R.S. Ivanov, "An application for Image Database Development for Mobile Face Detection and Recognition Systems"

[3] M. Turk, A. Pentland. "Eigenfaces for Recognition". Journal of Cognitive Neuroscience. Vol 3, No. 1. 71-86, 1991.

[4] Paul Viola, Michael J. Jones, Robust Real-Time Face Detection, International Journal of Computer Vision 57(2), 2004.

[5] http://jcp.org/en/jsr/detail?id=118

[6] http://jcp.org/en/jsr/detail?id=135

[7] http://www.gartner.com/it/page.jsp?id=498310

[8] W. Zhao, R. Chellappa, A. Rosenfeld, P.J. Phillips, Face Recognition: A Literature Survey, ACM Computing Surveys, 2003, pp. 399-458

[9] L. Torres, Is there any hope for face recognition?, Proc. of the 5th International Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2004, 21-23 April 2004, Lisboa, Portugal

[10] M. Turk, A Random Walk through Eigenspace, IEICE Transactions on Information and Systems, Vol. E84-D, No. 12, December 2001, pp. 1586-1595

[11] W.S. Yambor, Analysis of PCA-based and Fisher Discriminant-Based Image Recognition Algorithms, M.S. Thesis, Technical Report CS-00-103, Computer Science Department, Colorado State University, July 2000

[12] W.Y. Zhao, R. Chellappa, Image-based Face Recognition: Issues and Methods, Image Recognition and Classification , Ed. B. Javidi, M. Dekker, 2002, pp. 375-402

[13] M.-H. Yang, D.J. Kriegman, N. Ahuja, Detecting Faces in Images: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 24, No. 1, January 2002, pp. 34-58

[14] P.J. Phillips, H. Moon, S.A. Rizvi, P.J. Rauss, The FERET Evaluation Methodology for Face-Recognition Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, October 2000, pp. 1090-1104

[15] J.W. Tanaka, M.J. Farah, Parts and Wholes in Face Recognition, Quarterly Journal of Experimental Psychology Section A: Human Experimental Psychology, Vol. 46, No. 2, 1993, pp. 225-245

[16] Face recognition. http://www.biometrics.gov/Documents/FaceRec.pdf, August 2006. Subcommittee on Biometrics, National Science and Technology Council.

[17] Peter Belhumeur N., Joao Hespanha P., and Kriegman J. David. Eigenfaces vs. Fischerfaces: Recognition using class specfic linear projection. IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 19(7), 1997.

[18] Kamran Etemad and Rama Chellapa. Discriminant analysis for recognition of human faces images. volume 14, August 1997.

[19] R. C. Gonzales and R. E. Woods. Digital Image Processing. Addison-Wesley Publishing Comp., 1992.

[20] Burcu Kepenekci. Face recognition using gabor wavelet transform, September 2001.

[21] Javier Movellan R. Tutorial on gabor filters. http://mplab.ucsd.edu/tutorials/gabor.pdf.

[22] Henry Rowley A. Neural Network-Based Face Detection. PhD thesis, School of Computer Science, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, May 1999.

[23] Matthew Turk A. and Alex Pentland P. Face recognition using eigenfaces. In Journal of Cognitive Neuroscience, volume 3, page 7286, 1991.

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, December 2001.

[25] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In Conference on Computer Vision and Pattern Recognition (CVPR), page 7, 2001.

[26] W. Zhao, R. Chellappa, and P. J. Phillips. Subspace linear discriminant analysis for face recognition. Technical Report CAR-TR-914, Center for Automation Research, University of Maryland, College Park, MD., 1999.

[27] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. In ACM Computing Survey, volume 35, pages 399{458, December 2003}.

[28] Wenyi Zhao, Arvindh Krishnaswamy, Rama Chellapa, Daniel Swets L, and John Weng. Discriminant analysis of principle components for face recognition. 1998.

APPENDIX

A. FACE DATABASE

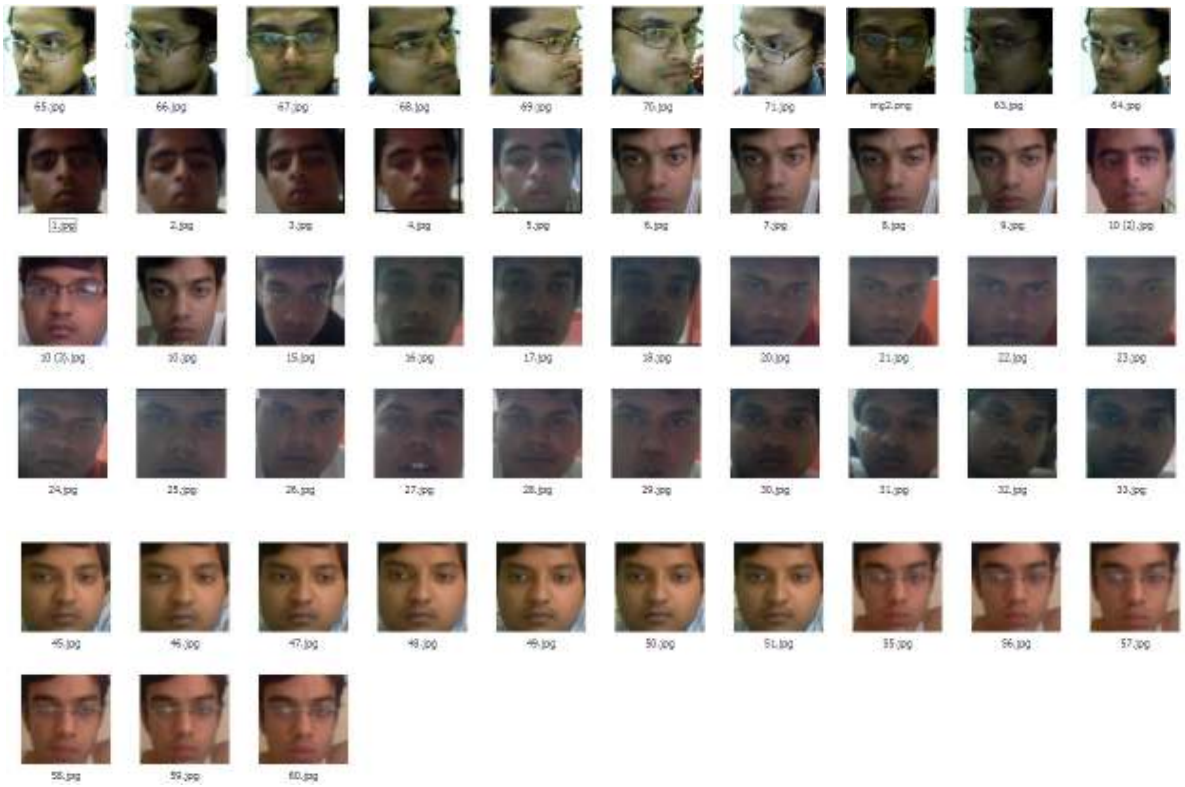
B. SNAPSHOTS

B1. Detection of a Face and its Recognition

B2. Detection of two Faces and their Recognition

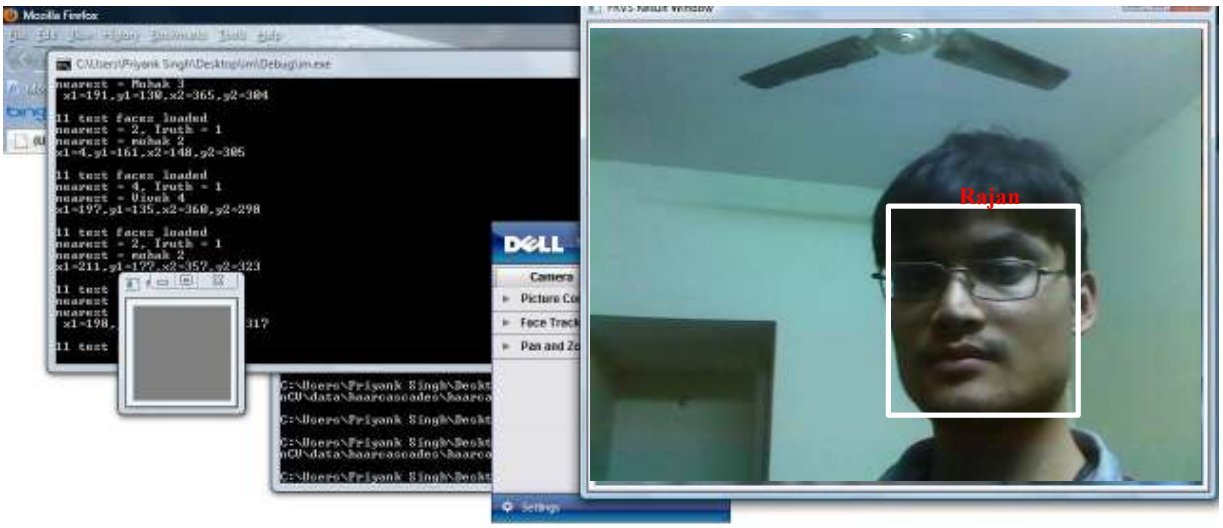
B3. Detection of three Faces and their Recognition

A. FACE DATABASE



B. SNAPSHOTS

B1. Detection of a Face and its Recognition



B2. Detection of two Faces and their Recognition