Chapter 1

INTRODUCTION

Face recognition is a task so common to humans. In General People does not even notice the extensive number of times it is performed every day. For a Human it is a usual task which is done normally every day. Although research in automated face recognition has been conducted since the 1960's, it is difficult problem in the area of computer vision which has recently caught the attention of the scientific community. Many face analysis and face modeling techniques have progressed significantly in the last decade [1].However, the reliability of face recognition schemes still poses a great challenge to the scientific community.

In general a face recognition system can be defined as a computer application or a system to identifying or verifying a person automatically from a digital image or a video frame from a video source.

1.1 Motivation

Face recognition has recently received a blooming attention and interest from the scientific community as well as from the general public. The general public has the interest mostly due to the unusual events like terrorist attack around the world, so this becomes our need for useful security systems. Facial recognition applications are far from limited to security systems as

Falsification of identity cards or intrusion of physical and virtual areas by cracking alpha numerical passwords appears frequently in the media.

These problems of modern society have triggered a real necessity for reliable, user -friendly and widely acceptable control mechanisms for the identification and verification of the individual.

Biometrics, which is based on authentication on the intrinsic aspects of a specific human being, appears as a viable alternative to more traditional approaches (such as PIN codes or pass words). Among the oldest biometric techniques is finger print recognition. In 700 AD this technique was first used in China for official certification of contracts. After that, in the middle of the 19th century, it was used for identification of persons in Europe. A currently developed biometric technique is iris recognition [2]. This technique is now used instead of passport identification for frequent flyers in some air ports in United Kingdom, Canada and the Netherlands. As well as for access control of employees to restricted areas in Canadian airports and in the New Yorks J F K airport. These techniques are inconvenient due to the necessity of interaction with the individual who is to be identified or authenticated.

Face recognition on the other hand can be a nonintrusive technique. This is one of the reasons why this technique has caught an increased interest from the scientific community in the recent decade. Instead of that automated facial recognition can be used in a lot of areas other than Security oriented applications (access-control/verification systems, surveillance Systems), such as computer entertainment and customized computer -human interaction. Customized computer -human interaction applications will in the near Future be found in products such as buildings, vehicles, aids for disabled people etc. The interest for automated facial recognition and the amount of applications will most likely increase even more in the future. This could be due to increased penetration of technologies, such as the internet and digital cameras, and due to a larger demand for different security schemes.

1.2 Applications of face recognition

The two primary tasks where Face recognition can be used is Verification and Identification. Verification is just like one-to-one matching process. When it presents with a unknown face image with a claim of identity, ascertaining whether the face is who he/she claims to be. On the other hand Identification is one-to-many matching process. Suppose we have an image of an unknown individual it is used to determining that person's identity by comparing that image with a database of images of known individuals.

There are various application areas in which face recognition can be used for these two purposes, which are given below.

• In Security system to access control to buildings, ATM machines, seaports/airports, and border checkpoints computer/ network security.

• In video Surveillance system a huge number of CCTVs monitor to identify the unusual event, any prime suspect, etc.

• General identity verification eg. Drivers' licenses, Electoral registration, National IDs, Internet banking, Electronic commerce, Identifying newborns, Passports, Employee IDs etc.

• "Smart Card" applications (in lieu of maintaining a database of facial images, a smart card can be stored the face-print, magnetic stripe, authentication of which is performed by matching the live image and the stored template).

• Multi-media environments with adaptive human-computer interfaces (context-aware systems or part of ubiquitous, behavior monitoring at childcare, recognizing a customer).

Based on this area of uses the application can be divided in to two parts as follows:

Government Use

- Law Enforcement. Minimizing victim trauma by narrowing mug shot searches,

-Verifying identifies for hospital records, and to identify the child molesters by school surveillance camera.

- Security/Counter terrorism. Comparing surveillance images to known terrorists.

- Immigration. Rapid progression through Customs.

- Legislature. Verify identity of Congressmen prior to vote.

Commercial Use.

- Day Care. Verify identity of individuals picking up the children.

- Missing Children/Runaways. Search surveillance images and the internet for missing children and runaways.

- Gaming Industry. Find card counters and thieves.

- Residential Security. Alert home owners of approaching personnel.

- Internet, E-commerce. Verify identity for Internet purchases.

- Healthcare. Minimize fraud by verifying identity.

- Benefit payments. Minimize fraud by verifying identity.

- Voter verification. Minimize fraud by verifying identity.

- Banking. Minimize fraud by verifying identity.

In addition to these applications, the current face recognition technology have also been modified and used for related applications such as gender classification [3], expression recognition [4] and facial feature recognition and tracking, each has an advantage such as expression recognition is utilized in the field of medicine for intensive care monitoring. Facial feature recognition and detection can be used for tracking a vehicle driver's eyes and thus monitoring his fatigue and stress detection. Another application of Face recognition is in conjunction with other biometrics such as speech, iris, fingerprint, and ear and gait recognition in order to enhance the recognition performance of these methods.

1.3 Difficulties in face recognition

There are many variations in the facial images that are considered for face recognition. These are due to changes in light intensity conditions, change in pose direction, Different face expression and aging effect etc.

1.3.1 Pose variations

Pose variations of any face image is one of the problem in recognition. when the Angle of rotation is small the pose problem cannot be considered but it can be more difficult when there are large pose variations along with illumination variations. The pose problem is illustrated in the following figure.

Figure 1.1: Variation in facial image due to different viewing angle

In the above figure, the same face appears differently due to changes in viewing condition. This makes face recognition difficult.

1.3.2 Light intensity variations

The Light intensity or illumination problem is illustrated in Figure 1.2, where the same face appears differently due to the change in lighting. the changes induced by intensity could be larger than the differences between individuals, This results a systems based on comparing images to misclassify the identity of the input image.

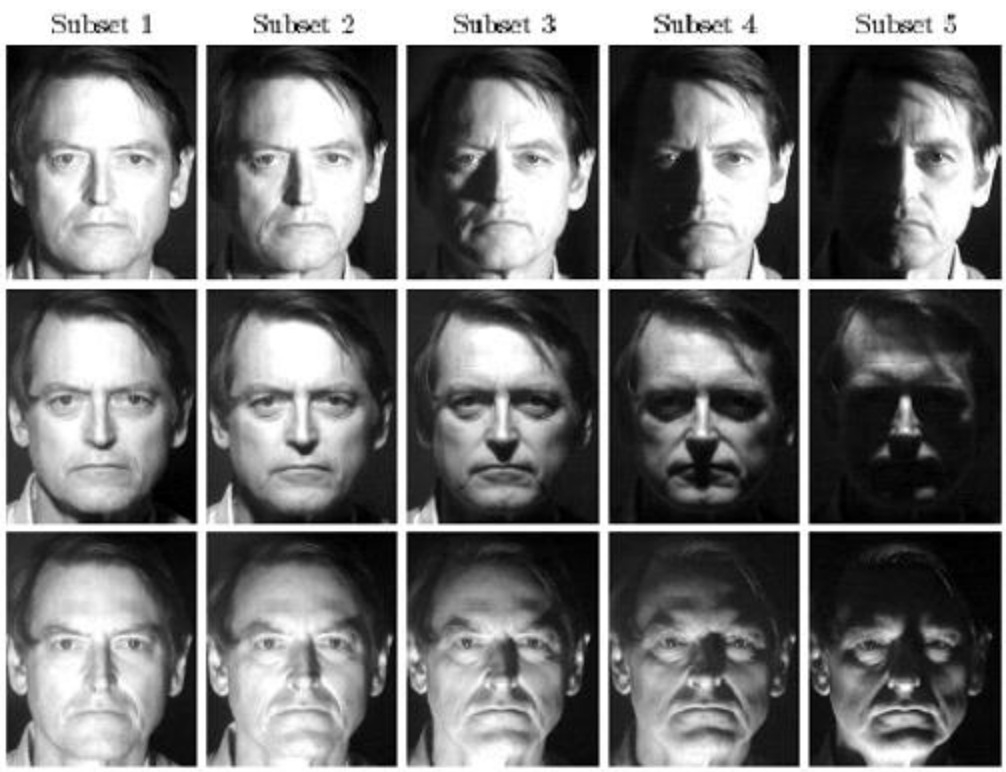


Figure 1.2: The same person seen under varying light conditions can appear dramatically different

In the above figure, we see that due to different lighting conditions the same face appears different. This poses a problem for face recognition systems.

1.3.3 Age variations

Apart from illumination conditions and pose variations, age is another challenge in face recognition. There are many significant changes in human face as a person grows older. The features of the face vary for every person and are affected by many factors such as exposure to sunlight, inherent genetics, and nutrition.

The color of the skin might also change with age. The performance of face recognition systems cannot cope with the dynamics of temporal metamorphosis over a period of time. Figure 1.3 shows the variations in face due to age.

Fig 1.3: Age cross-section: One of the subjects at age 3, 4, 19, 20, 22, 24 & 26.

In the above figure, we see that the variations in face as the age progresses are large. This makes identification of the face difficult. Apart from the above mentioned challenges, there are other factors also. The face images have similar geometrical features which make discriminating one face from the other in the database a challenging task. The results make it difficult to represent face images with distinct feature vectors that are invariant to transformation. The extracted feature vectors may possess overlapping characteristics, but the problem may be easily solved if there exists a feature extraction method which can generate distinct features for each class of images or a classification technique capable of discriminating the overlapping features of the images

1.3.4 Occlusion effect:

Occlusion is a common difficulty encountered in applications of automatic face recognition. There are many way of being occlusion such as sunglasses, hats, eyeglasses, or scarves, as well as objects such as cell phones placed in front of the face. Moreover, even in the absence of an occluding object, violations of an assumed model for face appearance may act like occlusions: e.g., shadows due to extreme illumination violate the assumption of a low dimensional linear illumination model. Robustness to occlusion is therefore essential to practical face recognition.

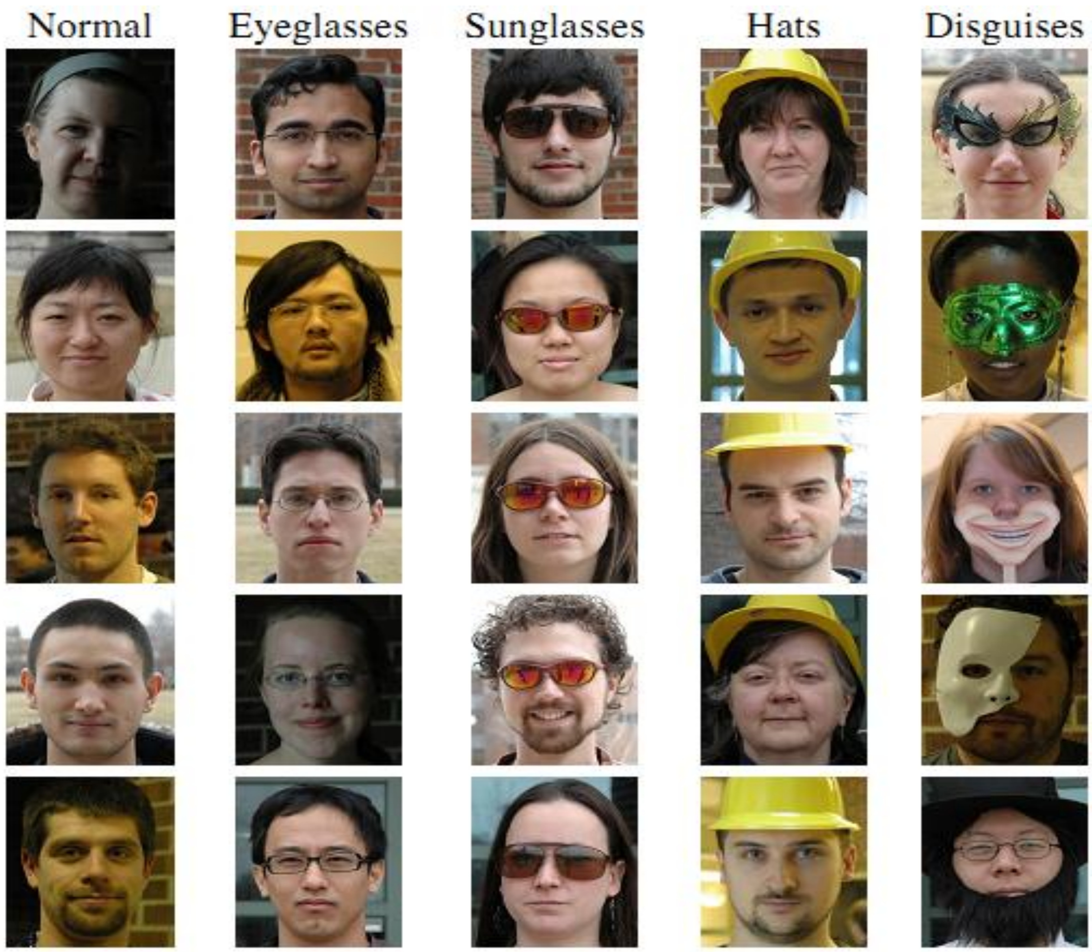


Figure 1.4 Occlusion images from the different test categories.

1.4 Different approaches for face recognition

Face recognition techniques can be classified in many ways. One possible classification at high-level is:

1.4.1 Region based approach

These methods are more effective than holistic methods. They work on the principle of applying different processing methods to distinct face regions and hence filter out those regions that are mostly affected by expression changes or spurious elements. In other words they partition the face surface into regions and extract appropriate descriptors for each of them. But, these are also sensitive to face alignment which makes useful face regions hard to detect automatically. Also, local features and differences in resolution determine their performance.

1.4.2 Holistic approach

In holistic methods, the face as a whole is taken as input data. One of the main algorithms that come under this category is the Eigen face method. Eigen face method is based on the implementation of Principal Component Analysis (PCA). In this technique, we find the features of the studied images by looking for the maximum deviation of each image from the mean image. To obtain this variance we get the eigenvectors of the covariance matrix of all the images. The Eigen face space can be obtained by applying the Eigen face method to the training images. After that, we project the training images into the Eigen face space. Later, we project the test image into this new space and the distance of the projected test image to the training images is used to classify the test image. Other examples of holistic methods are fisher faces and support vector machines.

1.4.3 Hybrid and multimodal approach

The idea of this method comes from how human vision system sees both face and local features which includes eyes, lips and nose). There are many examples in hybrid approach eg. Modular Eigen faces and component-based methods [5]. Even though there are wide range of algorithms available for both face detection and recognition. Matching these algorithms on to our embedded system will be a real challenge.

1.5 History of Face Recognition

Face recognition is a task to find out the interesting features of the face and compare these to the same features on other faces. Some of the earliest work on face recognition was done by Darwin and Galton. The Analysis of the different facial expressions due to different emotional states is done by Darwin, whereas Galton studied facial profiles. However, in the late 1960's and early 1970's the first real attempts to develop semi-automated facial recognition systems began, and were depend on geometrical information. Here, landmarks were placed on photographs locating the major facial features, such as eyes, ears, noses, and mouth corners. Relative distances and angles were computed from these landmarks to a common reference point and compared to reference data. In Goldstein et al. (1971) a system is created of 21 subjective markers, such as hair color and lip thickness. These markers proved very hard to automate due to the subjective nature of many of the measurements still made completely by hand.

Fischler et al. [6] (1973) and later by Yuille et al. (1992) proposed a more consistent approach to do facial recognition. In this approach they measured the facial features using templates of single facial features and mapped these onto a global template.

In summary, most of the developed techniques during the first stages of facial recognition focused on the automatic detection of individual facial features. The greatest advantages of these geometrical feature-based methods are the insensitivity to illumination and the intuitive understanding of the extracted features. However, even today facial feature detection and measurement techniques are not reliable enough for the geometric feature-based recognition of a face and geometric properties alone are inadequate for face recognition.

There were some drawback in the geometric feature-based recognition, the technique has gradually been abandoned and an effort has been made in researching holistic color -based techniques to obtain the better results Holistic color–based techniques align a set of different faces to obtain a correspondence between pixels intensities, a nearest neighbor classifier can be used to classify new faces when the new image is first aligned to the set of already aligned images. By the appearance of the Eigen faces technique [7], a statistical learning approach, this coarse method was enhanced. Instead of directly comparing the pixel intensities of the different facial images, the dimension of the input intensities were first reduced by a Principal Component Analysis (PCA) in the Eigen face technique. Eigen faces is a basis component of many of the image based facial recognition schemes used today. One of the current techniques is Fisher faces.

This technique is widely used and referred [6]. It combines the Eigen faces with Fisher Linear Discriminant Analysis (F L DA) to obtain a better separation of the individual faces. In Fisher faces, the dimension of the input intensity vectors is reduced by PCA and then FLDA is applied to obtain an optimal projection for separation of the faces from different persons. After development of the Fisher face technique, many related techniques have been introduced. These new techniques provide an even better projection for separation of the faces from different persons. They try to strengthen the robustness in coping with differences in illumination or image pose. Techniques like Kernel Fisher face [8], Laplacian faces [1] or discriminative common vectors can be found among these approaches.

1.6 Scope of the work

In this work, we have focused on the problem of face recognition and have derived a novel way for recognizing faces using Scale-invariant feature transform. We have the standard ORL Database it is divided in two parts. The first part is training image and Second part is the test image. The SIFT features are generated for every training image and then k nearest neighbors classifier is used for the matching scheme. The recognition results demonstrate its robust performance under different expression conditions, Pose variation, illumination changes and partial occlusion.

For all the experiments conducted the ORL (Olivetti Research Laboratories) database has been used.

1.7 Organization of the thesis

The remaining part of this thesis is organized in the following chapters:

Chapter 2: Literature survey

In this section, the critical points of current knowledge on face recognition have been reviewed. Substantive findings as well as theoretical and methodological contributions in the field of face recognition have been included. This section summarizes the work done in the field of face recognition using various techniques in recent times.

Chapter 3: Proposed face recognition system

In this section, image preprocessing has been discussed. Also, detailed description about the method of feature extraction used has been provided. The Scale-invariant feature transform is used for feature extraction. Along with this, the methods of classification also have been discussed. These include k nearest neighbor’s classifier etc.

Chapter 4: Experiments and results

In this section, initially the dataset used for the work has been mentioned. Thereafter, all the experiments conducted along with the analysis of the results have been explained in detailed. The section is divided into two parts, feature extraction and classification. Figures and tables have been provided wherever necessary.

Chapter 5: Conclusion and Future Scope

In this section we will discuss the conclusion of the thesis work and the future scope of the work.

References: This section gives the reference details of the thesis. Appendix A: Abbreviations

Appendix B: Introduction to Image Processing in MATLAB

Chapter 2

LITERATURE SURVEY

Although face recognition systems are known for decades, there are many research work has been done on this topic. The Face recognition system is divided into three parts;

1. Detection

2. Recognition

3. Detection & Recognition

Face detection is the first step of face recognition system. Output of the detection can be location of face region with facial features such as eyes, mouth, eyebrow, nose etc. and location of whole face region and Most of the algorithms are combination of methods for detecting faces to increase the accuracy so detection methods are difficult for the classification, because. Mainly, detection can be classified into two groups as Knowledge-Based Methods and Image-Based Methods. The methods for detection are given in Figure 2.1.

Knowledge-Based methods use information about Facial Features, Skin Color or Template Matching. Facial Features play an important role to find mouth, eyes, nose or other facial features to detect the human faces. Skin color is unique and it is different from other colors, and its characteristics are invariant with respect to changes in pose angle and occlusion. Skin color is modeled in each color spaces like RGB (Red-Green-Blue), YCbCr (Luminance-Blue Difference Chroma-Red Difference Chroma), HSV (Hue-Saturation-Value), YUV (Luminance-Blue Luminance Difference-Red Luminance Difference), and in statistical models. Face has a unique property to differentiate from other objects and hence a template can be generated to scan and detect faces. Facial features are important information for human faces and standard images can be generated using these information. In literature, many detection algorithms which are based on facial features are available.

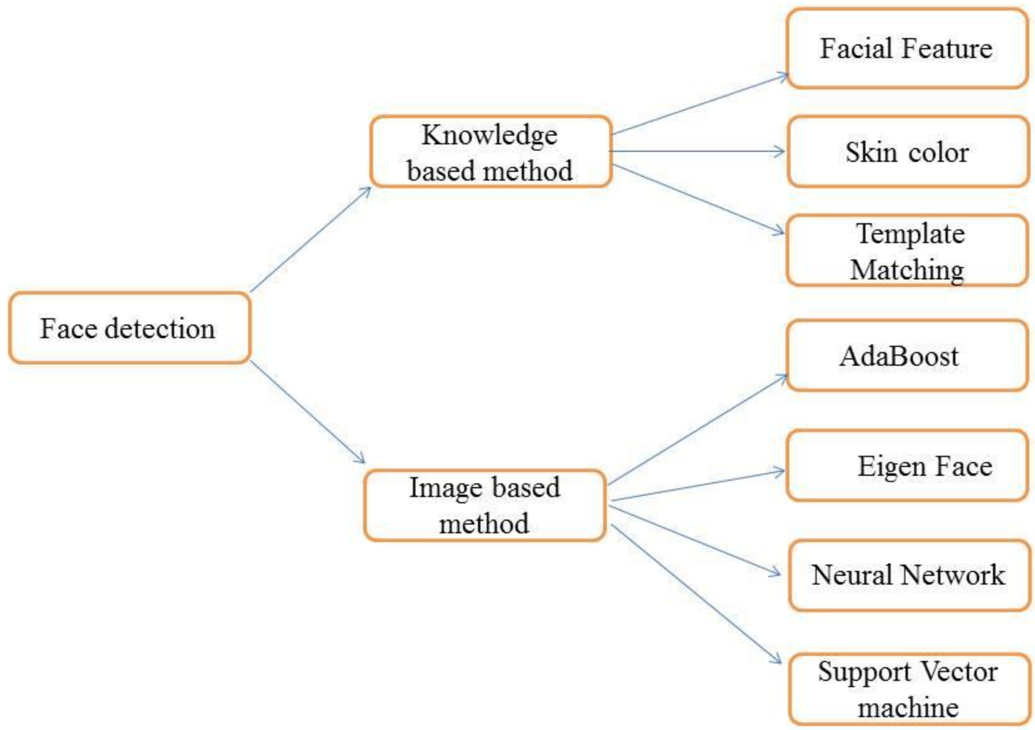


Figure 2.1 Methods for Face Detection

Zhi-fang et al. [9] detect faces and facial features by extraction of skin like region with YCbCr color space and edges are detected in the skin like region. Then, eyes are found with Principal Component Analysis (PCA) on the edged region. Finally, Mouth is found based on geometrical information. Another approach extracts skin like region with Normalized RGB color space and face is verified by template matching. To find eyes, eyebrows and mouth, color snakes are applied to verified face image [10]. Ruan and Yin [11] segment skin regions in YCbCr color space and faces are verified with Linear Support Vector Machine (SVM). For final verification of face, eyes and mouth are found with the information of Cb and Cr difference. For eye region Cb value is greater than Cr value and for mouth region Cr value is greater than Cb value. There is another application which segments skin like regions with statistical model. Statistical model is made from skin color values in Cb and Cr channel in YCbCr color space. Then, face candidates are chosen with respect to rectangular ratio of segmented region. At last, we can verify candidates with eye & mouth map [12]. Also, RGB color space can be used to segment skin like region and skin color like region is extracted to be face candidate. Candidate is verified by finding facial features. Eyes and mouth are found based on isosceles triangle property. Two eyes and one mouth create an isosceles triangle and also distance between two eyes and distance from mid-point of eyes to mouth are equal. After eyes and mouth is found, Feed Forward Neural Network (FFNN) is used for final verification of face candidate [13]. Bebar et al. [14] segment with YCbCr color space and eyes & mouth are found on the combination of segmented image and edged image. For final verification, horizontal and vertical profiles of the images are used to verify the position of the eyes and mouth. These all methods are using firstly skin segmentation to eliminate non-face objects in the images to save computational time.

Skin color is one of the most significant features of human face. Skin color can be modeled with parameterized or non-parameterized methods. Skin color region can be identified in terms of elliptical modeling, threshold region, statistical modeling (i.e. Gaussian Modeling). Skin color can be described in all color spaces like RGB, YCbCr, and HSV. RGB is sensitive to light changes but YCbCr and HSV are not sensitive to changes of light. Because these two color spaces have separate intensity and color channel. In literature there are many algorithms which are based on skin color available. Kherchaoui and Houacine [15] modeled skin color using Gaussian Distribution Model with Cb and Cr channel in YCbCr color space. Then skin like region is chosen as a face candidate with respect to the bounding box ratio of the region and candidates are verified with template matching. Another method preprocesses the given image to remove background part as a first step. It is done by applying edge detection on the Y component of YCbCr color space. Then, the closed region is filled to take it as foreground part. After this, segmentation of skin is done on YCrCb color space with conditions. The segmented parts are corresponding to candidate and verification is done by calculating the entropy of the candidate image and use thresholding to verify face candidate [16]. Qiang-rong and Hua-lan [17] applied white balance correction before detecting faces. The color value is important for segmentation, so while acquiring the image colors may reflect false color. To overcome this, white balance correction should be done as a first step. Then, skin color like regions are segmented using elliptical model in YCbCr. After skin regions are found, they combined with edged images to gray scale image. Finally, the combined regions are verified as face by checking bounding box ratio and area inside the bounding box. Another method is to segment skin like region with threshold value in Cb, Cr, Normalized r and Normalized g. Then candidate for face is chosen with respect to area bounding box, bounding box ratio and ratio of area inside and minimum area of the region. After candidates are found, then Ada Boosting method is applied to find face candidates. The verification can be done by combining both results from skin like region and AdaBoosting [18]. Also, skin color can be modeled in elliptical region in Cb and Cr channel in YCbCr color space. Skin like region is segmented if the color value is inside elliptic region and candidate regions are verified using template matching [19]. Peer et al. [20] detect faces using only skin segmentation in YCbCr color space and researchers generate the skin color conditions in RGB color space as well. Second way for skin color modeling is done by Self Organizing Map (SOM) Neural Network (NN). After skin segmentation is applied, each segment is taken as candidate and verified if they can fit into elliptic region or not.

Significant information about the human face detection is pattern of human face. Template matching can be applied over segmented region or window scanning technique. Scanning technique is done with small size window like 20x20 or 30x30 pixel window. This method scans the entire original image, and then decreases the image size with some iteration suitable to re-scanning. Decreasing the size is important to locate the large or medium size faces. However, this requires excessive computational time to locate faces. Template matching in segmented region requires much less computational time than scanning, because it only considers the matching of segmented part. In literature many applications are available by using template matching, i.e.,[21]. Chen et al, use half-face template, instead of full-face template. Computational time is decreased by this method. And this half-face can be adopted to face orientations. Another approach uses abstract templates which are not really a image like but composed of some parameters (i.e. size, shape, color, and position). Regions with Skin like are segmented with respect to YCbCr color space. Then the abstract templates of eye and eye pair are applied to the segmented region. Firstly templates locate the eyes region and second templates locate the each eye. Second the orientation of eyes is also determined by template. Then Texture template is applied to verify the face candidate region.

Image-Based methods use training/learning methods to make comparison between face and non-face images. For this technique, a large number of images of face and non-face should be trained to increase the accuracy of the system. AdaBoost, Eigen Face, Neural Networks and Support Vector Machines are kind of methods that are used commonly in face detection algorithms. Face and non-face images are described in terms of wavelet feature in AdaBoost method. Principal Component Analysis (PCA) is used to generate the feature vector of face and non-face image in Eigen Face method. Also, the given information vector is compressed by PCA method. Kernel function is applied to describe face and non-face images in Support Vector Machines (SVM). Face and non-face images are also classified by artificial neuron structure in Neural Networks (NN).

AdaBoost is an algorithm that constructs strong classifier from weak classifiers. Face candidates are found by applying AdaBoost algorithm. Then, verification is done with Cascade Classifier. This algorithm can suitable for faces; left, left+45, front, and right+45, right pose.

As mentioned above PCA is used to generate feature vector of face and non-face images in Eigen Face method. Window scanning technique and segment-based analysis can be used for this purpose. The applications of Eigen Face are available in literature. Wang and Yang [22] apply template matching to find face candidates. Then, 2D PCA is used to extract the feature vector. The image matrix is directly given to the 2D PCA instead of vector. That decreases computational time for calculating covariance matrix. After applying PCA, Minimal Distance Classifier is used to classify the PCA data for either face or non-face cases. Another algorithm uses NN and PCA. PCA method is introduced to the given image to extract the face candidates at first. Then, candidate images are classified with NN to eliminate non-face images. Last face candidates are verified with geometrical distribution of edges in the face regions. Also, PCA and AdaBoost applications are applied with window scanning technique. First PCA is applied, that resultant feature vector is applied to input of Ada Boost technique. PCA feature vectors generate Strong classifier from for AdaBoost method [23].

Neural Networks (NNs) are good at pattern recognition and classification problem, while face and non-face images are two classes in face detection. Again window scanning technique or segment-based implementation can be used to classify the images. Many types of NN used for detection problem i.e., Perceptron, Probabilistic NN, Back Propagation NN, Radial Basis NN, Self Organizing Competitive NN. Anijiantis[24] apply image preprocessing seeding the NN, and then two-layer perceptron is used to determine whether this input is face or non-face. Window scanning technique is used to detect faces. Anagnostopoulos et al. [25] segments skin like region with Fuzzy Logic in RGB color space and segmented regions are sent to Probabilistic NN to verify the face region. Another approach extracts skin like regions in YUV color space to decrease computational time with respect to window scanning technique. Then, segmented regions are sent to Back Propagation (BP) NN to generate face candidates. Finally, face candidates are verified with Bayesian Decision [26]. Skin like region segmentation with YCbCr is applied. Then, feature vector is generated on segmented region by wavelet invariant moment. Then, feature vector is classified via Self Organizing Competitive NN. This algorithm can detect frontal and profile faces up to 20 fps video images [27]. Also, HSV color space can be used to segment skin like region and feature vector of segment is calculated with 2D Discrete Cosine Transform (DCT). Finally, BPNN is used to classify the feature vectors that belong to skin like regions. On the other hand only using Radial Basis Neural Network (RBNN) can be used to detect faces with scanning technique [28].

For last method of image-based approach is the Support Vector Machines (SVM). SVM is trained with face and non-face images to construct a kernel function which classifies the face and non-face images. A Different approach is applied to find face candidates. The face candidate is found by generalized symmetry based on the location of eyes. Finally, face candidate is validated/classified with SVM. Also, SVM can be applied to detect faces with scanning technique [29]. Jee et al. [30] apply skin like region segmentation based on YCbCr and eye candidate are found with edge information inside the region. Then, SVM method verifies eyes. After verification, face candidate are extracted with respect to eye position. As a final verification, face candidate is sent to SVM for verification. Another approach for SVM is using one class based SVM. Only face class is generated instead of generating two classes that are face and non-face. Since, the non-face images are difficult to model.

The face detection performance is investigated but each algorithm use either their own library images or face detection database images. Because of that performance comparison between the algorithms is difficult and best algorithm cannot be decided.

For face recognition system, the other part is recognition part. The recognition can be achieved with 2D or 3D image data that have both advantages and disadvantages. 2D image data can be obtained easily and less expensive than 3D. On the other hand, 2D images are sensitive to light changes but 3D images are not. With 3D images, surface of the face can easily model but 2D images do not contain the depth data. Also, face recognition algorithms are performed over the face libraries. These libraries are made with standard face images, so the face recognition system should deal with this problem. The face recognition methods are given in Figure 2.2. Face recognition is a pattern recognition problem, so this training or learning algorithm should be used to make comparison between the faces. For 2D recognition, the methods of Linear or Nonlinear Projection and Neural Networks are used. Linear/Nonlinear Projection methods are PCA, Linear Discriminant Analysis (LDA), Gabor Wavelet, and Spectral Feature Analysis. Neural Network approaches are Wavelet NN, BPNN, RBNN, FFNN, and Multi-Layer Cluster NN. For 3D recognition application, Corresponding Point Measure, Average Half Face and 3D Geometric Measure are used.

2D Linear/Nonlinear Projection methods generate feature vector for each person, after then classification is done for the input person inside the database. To generate the feature vector is also importance to reduce dimension of the input images. One method applies image enhancement to suppress the bad lighting condition before recognition process. Image enhancements are also called as logarithm transform and normalization. Then, feature extraction is done with Gabor Wavelets. Finally, Fisher Face is used to classify input face [31]. Song et al. [32] apply a different approach on image preprocessing/enhancement. Preprocessing is done before feature extraction to calculate the illumination difference between right and left part of face. If large amount of difference occur than take the mirror of average illuminated part. After face image is preprocessed, PCA is used for feature extraction. Euclidian Distance is applied for Classification of feature vector. Other implementation uses Layered Linear Discriminant Analysis (LDA) to classify faces, and the benefit of Layered LDA is using small sample size database [33]. Also, using Spectral Feature Analysis can reduce the sample size in database [34]. Extending the feature extraction can improve the performance of the system i.e. applying PCA, Gabor wavelet and then, Independent Component Analysis (ICA) on face image. After feature extraction, measure the cosine similarity and the nearest neighbor classification rule is used to recognize. Another approach for input vector uses facial distance. Dimension reduction is performed with PCA and classification is achieved with NN.

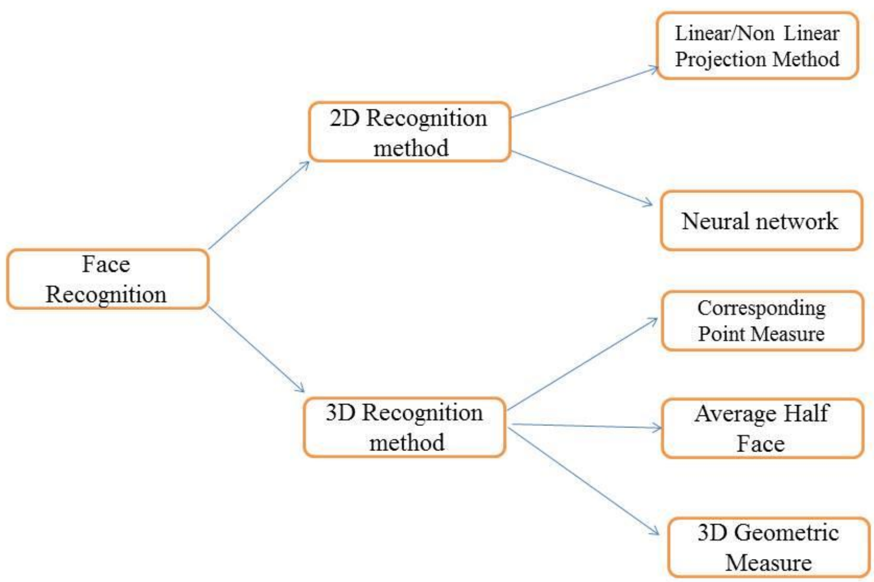


Figure 2.2: Methods for Face Recognition

Since face recognition is a kind of pattern recognition problem, neural networks are good at pattern recognition and can be capable of distinguish faces even for the large number of faces in a database. One possible way, Faces sent to NN as a vectored matrix image or feature-extracted image. The Algorithm which is based on feature-extracted can have more performance because of using small dimension vector. Wavelet NN, BPNN, RBNN, FFNN, and Multi-Layer Cluster NN are used in literature. Wavelet NN uses transfer function which is made from wavelet function series and this series avoids the blindness in structure design of BPNN [35]. On the other hand, instead of using gray scale images, color images can be used as input to the BPNN. R, G, B channels are inputs together and network feed with color information for easy discrimination [36]. Researchers also work on effects on using different type of networks and feature extractors. As network, BPNN, RBNN, and Multilayer Cluster NN used and as feature extractor, Discrete Wavelet Transform, Discrete Radon Transform, DCT, PCA. The best performance is reported as the combination of BPNN with DCT. Rida and Dr BoukelifAoued [37] implement FFNN with Log Sigmoid transfer function network to classify the given faces. Fast ICA is used as feature extractor and RBNN used as classifier, and only RBNN is used to classify the input images. Also, Singular Value Decomposition (SVD) is used as feature extractor in BPNN. Another type of image enhancement application is using the Laplacian of Gaussian filter, and then applying SVD to filtered image to get the feature vector. Finally, face image is classified by using FFNN [35].

3D face recognition methods use 3D data of face, which is composed of cloud of points. One implementation uses iterative closest point to align face. Then, as image enhancement, noises are reduced and spikes are removed. The nose tip detected and a sphere is cropped with origin of nose tip. This sphere is face feature. Then, using Corresponding Point Direction Measure the given face is classified [39]. Different approaches use half face instead of full face. The average half face is generated with Symmetry Preserving SVD. Then, feature extraction is completed with PCA and classification is achieved with Nearest Neighbor Classification [40]. There is applied input as facial profiles. Central Symmetry Profile and Cheek Profile are extracted for faces and Fourier Coefficients are found to generate feature extraction. At last Measure of Similarity is applied to classify the faces. The geometric information about facial features is used in [41]. Nose, eyes and mouth are found and 3D geometric measurements are applied. The measurements are straight-line Euclidean distance, area, angle curvature, distance, and volume. The classification is based on Similarity Measure.

Face recognition techniques are described in previous paragraphs. Selection of the best algorithm cannot be done because of nonstandard face libraries and non-universal face databases. Also, recognition and detection method can be affected by pose changes, illumination changes, facial expression and occlusion. All methods cannot handle all effects in the same algorithm. So, it is difficult to design an algorithm that handles all effects.

Our problem is to design a face recognition system which find faces in the images and extract faces. After that extracted face should be recognized. In literature some methods are available on this problem and combinations of methods are used. Lee and Liang [42] found first eyes and lip with AdaBoost method. Then, feature vector created based on eye, lip and nose region pixel values, and distance between facial features. The feature vector dimension is reduced with PCA and LDA. The classifier is done with RBNN. Embedded Hidden Markov Model (HMM) method is used both finding and classifying the faces. In another work, detection and classification is done with NN, and feature extraction is done with taking the Fast Fourier Transform of detected images. Also, polynomial NN is used to detect faces and pseudo 2D HMM is used to classify faces. Another implementation segments skin like regions with RGB, and then segmented parts are verified with template matching. Detected Faces that are feature extracted with PCA and classification is done with similarity measure. Zhang [43] segments skin like regions and face candidates are found with face symmetry verification. Then, eye template is used to determine the rotation of face. Then, with Haar Wavelet Feature, Gradient Based Directional Feature, and Gradient Based Directional Feature information are generated the feature vector. Finally, classification of feature vector is achieved with NN.

Some performance statistics of algorithms are given in Appendix section (App. 1). Detection/Recognition rates show the performance of correct detection of faces in the given image or recognition of given face image. Miss rate shows percentage of missed face in the given image. False rate gives the percentage of wrong detected face or wrong classified face.

Methods for face recognition system are investigated and possible solutions are studied extensively. Selected face detection method is skin segmentation and face candidate is verified with eyes and mouth finding. Then, extracted faces are classified with FFNN. The detail about the face recognition system will be explained in next chapter.

Chapter 3

PROPOSED FACE RECOGNITION SYSTEM

The entire face recognition system used in this work consists of three steps:

1) Image preprocessing

2) Feature extraction

3) Classification

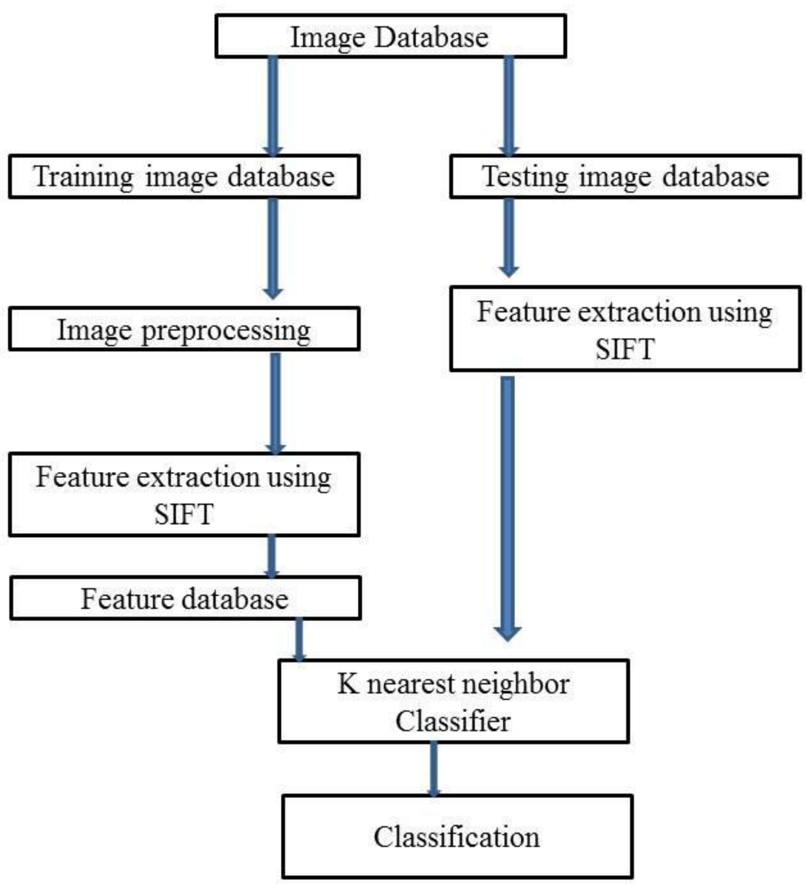


Figure 3.1: Proposed face recognition system

An image is extracted from database and the required preprocessing is done on it. Preprocessing helps in improving the results obtained from the feature extraction methods. Thereafter, features are extracted using the appropriate methods and feature vectors are formed. These feature vectors are then classified using various classifiers. All the above mentioned steps have been elaborated in this chapter.

3.1 Image preprocessing

Image preprocessing commonly comprises of a series of sequential operations including image enhancement or normalization, geometric correction, removal of noise etc. It is done before applying the feature extraction techniques. Using the above mentioned techniques, we alter the image pixel values permanently and use this improved data for later analyses.

3.1.1 Need for preprocessing

Image preprocessing more importantly is done for noise removal which is mainly added during image acquisition. Image acquisition is a very important step for the quality control since it provides the input data for the whole process. Digital image is acquired by an optical sensor which is always a video camera (with one line or a matrix of CCD). It provides accurate and noiseless image. Local illumination is also directly linked with the quality of image acquisition.

Sometimes there might be other reasons also. There may be a need to convert a colored image into gray scale. Image enhancement by various methods also comes under image preprocessing.

3.1.2 Various image preprocessing methods

i) Noise removal

Images taken with both digital cameras and conventional film cameras pick up noise. It is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene from a variety of sources. For further use of these images, it is required that the noise should be removed. For removal of noise various filters can be applied on images, like high pass filter, Gaussian filter etc.

ii) Contrast adjustment

The contrast of an image refers to the distribution of the dark and light pixels. A low-contrast image means there is a small difference between its light and dark pixel values. This in turn means that the histogram of a low-contrast image is narrow. Since the human eye is sensitive to contrast rather than absolute pixel intensities, a better image can be obtained by stretching the histogram of the image so that the full dynamic range of the image is used.

iii) Intensity adjustment

Image enhancement techniques are used to improve an image. These techniques can be used to increase the signal-to-noise ratio or make certain features easier to be seen by modifying the colors or intensities. Adjustment of Intensity is an image enhancement technique which maps the image intensity values to a new range.

iv) Histogram Equalization

Histogram equalization distributes evenly the occurrence of pixel intensities so that the entire range of intensities is occupied. This method usually increases the contrast of images globally, especially when the required data of the image is represented by close contrast values. By adjusting the histogram, the intensities can be better distributed. Due to this, the areas of lower local contrast gain a higher contrast. Histogram equalization effectively spreads out the most frequent intensity values.

v) Normalization

Image normalization is done in order to reduce the amount of computation so that the method is more efficient. The pixel values ranging from 1 to 255 are mapped to lie within 0 to 1.

3.2 Feature extraction

In image processing, feature extraction is a special form of dimensionality reduction. It simplifies the amount of resources required to describe a large set of data accurately. Image features can refer to the global properties of an image like average gray level, shape of intensity histogram etc. or the local properties of the image like shape of contours, elements composing textured region etc.

To choose the features to be extracted is critical and the following concerns should be kept in mind while selecting them.

1) Enough information about the image should be carried by the features and there should be no requirement of any domain specific knowledge for their extraction.

2) Their computation should be easy so that the approach is feasible for a large image database and rapid retrieval.

3) They should be meaningful i.e. they should be associated with interesting elements in the image formation process. They should be invariant to the image formation process like invariant to the viewpoint and illumination of the captured digital images.
4) They should relate well with the human perceptual characteristics since

the end users will finally determine the suitability of the images retrieved.

5) The extracted features should be able to be detected or located from the images with the help of algorithms. Also they should be easily described by a feature vector.

In this work, the features are extracted from images using Scale invariant Feature transform (SIFT) methods. The extracted features are then subjected to classification.

THEORY OF SIFT

The feature extraction technique, done by Scale invariant Feature transform has such advantages as scale invariability, rotation invariance, affine invariance, which could reduce mismatching because of occlusion, confusion, noise, and keep high matching rate. The detailed step of this method is as follows.

In first step Extreme value or key point is detected in the different scale of the image, secondly, the position of key-point is located by the filter of the key-points; thirdly, the key-point's gradient direction is determined by the key-point neighborhood; fourthly, the feature descriptor is determined by the feature of the key-point .

3.2.1. Extreme Value Detected in Scale Space

In [44] SIFT, the scale transformation is realized by the Gaussian convolution kernel, and the description $L(x, y, \sigma)$ of input image $I(x, y)$ in different scale can be expressed by the

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y)\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

The 2D Gaussian convolution kernel*(G(x, y, σ))* is described by (2)

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{x^2+y^2 / 2\sigma^2} \quad \text{.........................(2)}$$

According to (1) and (2), the description of *I(x, y)* in different scale is calculated by its convolution with a Gaussian filter, and the Gaussian function is only the kernel function. After the convolution, the calculated image is a Gaussian image. (*x*, *y*) represents the pixel's coordinate of the input image; *σ* is the scale factor; L(x, y, σ) is the spatial scale images. The image I(x, y) zoom with *σ*, and the smoothness of the image would change with de change of *σ*, and then a series of scale image could be obtained.

According to those scale images, the extreme point (key- point) will be detected. The difference of Gaussian (DoG) image between scale *kσ* and *σ* is as in

$$\begin{aligned} DoG(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \quad \text{..................(3)} \end{aligned}$$

By (3), the feature of image, such as contour, corner, edges, can be detected in different scale, and the candidate key-points of image are chosen by the extreme value of the features. That is: by comparing every point with its neighborhood which includes adjacent scale, the position and scale of local the extreme point are the key-points'. The schematic diagram is Fig. 3.2.

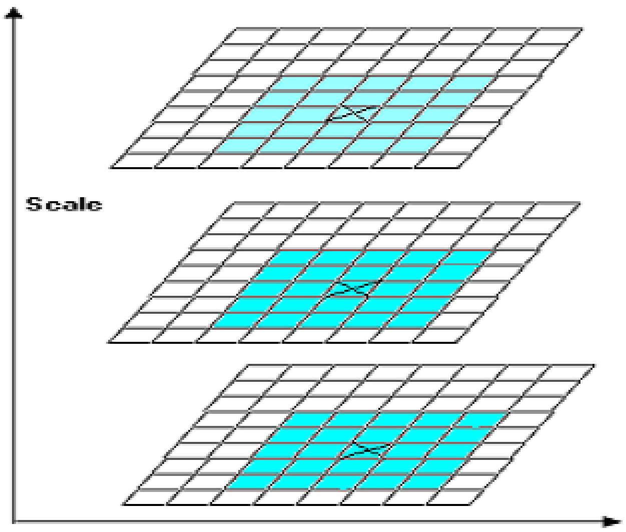


Figure 3.2. Key-point detection

3.2.2. Location of Key-points

The location of key-point is considered to filter the key- points which are sensitive to noise or have no edge effect in this process. The reference shows that, according to Taylor quadratic expansion, *DOG(x,y,σ)* can delete the extreme points which have lower contrast, and the value of Hessian vector and the ratio of determinant can reduce the edge effect.

3.2.3. Interpolation of nearby data for accurate position

First, for each candidate key point, To determine its position accurately, Interpolation of nearby data is used. The initial approach was to just locate each key point at the location and scale of the candidate key point. The new approach calculates the interpolated location of the extremum to improve matching and stability. For interpolation, Quadratic Taylor expansion of the Difference-of-Gaussian scale-space function is applied, $D(x, y, \sigma)$ with the candidate key point as the origin. This Taylor expansion is given below:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

Where D and its derivatives are evaluated at the candidate key point and X=(x, y, σ) is the offset from this point. By taking the derivative of this function with respect to X and setting it to zero The location of the extremum, $\hat{\mathbf{x}}$, is determined. If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, it indicates that the extremum lies closer to another candidate key point. In this case, the candidate key point is changed and the interpolation performed instead about that point. Otherwise the offset is added to its candidate key point to get the interpolated estimate for the location of the extremum. A similar sub pixel determination of the locations of scale-space extrema is performed in the real-time implementation based on hybrid pyramids developed by Lindeberg and his co-workers.

3.2.4. Discarding low-contrast key points

The value of the second-order Taylor expansion D(X) is computed at the offset $\hat{\mathbf{x}}$ to remove the key points with low contrast, If this value is less than 0.03, the candidate key point is removed. Otherwise it is kept, with final location $\mathbf{y} + \hat{\mathbf{x}}$ and scale σ, where y is the original location of the key point at scale σ.

3.2.5. Eliminating edge responses

Even if the candidate key point is not robust to small amounts of noise, the DoG function will have strong responses along edges. Therefore, in order to increase stability, we have to eliminate the key points that have poorly determined locations but have high edge responses.

For poorly defined peaks in the DoG function, the principal curvature across the edge would be much larger than the principal curvature along it. To Find these principal curvatures amounts to solving for the eigenvalues of the second-order Hessian matrix, **H**:

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigenvalues of **H** are proportional to the principal curvatures of D. It is the ratio of the two eigenvalues, let suppose $\alpha$is the larger one, and $\beta$the smaller one, with ratio $r = \alpha/\beta$, is sufficient for SIFT's purposes. The trace of **H**, i.e., $D_{xx} + D_{yy}$, is the sum of the two eigenvalues, while its determinant, i.e., $D_{xx}D_{yy} - D_{xy}^2$, is the product. The ratio $\mathrm{R} = \mathrm{Tr}\,(\mathbf{H})^2 / \mathrm{Det}\,(\mathbf{H})$can be equal to $(r+1)^2/r$, which depends only on the ratio of the eigenvalues rather than their individual values. R should be minimum when the eigenvalues are equal to each other. Therefore If the absolute difference between the two eigenvalues is high which is equivalent to a higher absolute difference between the two principal curvatures of D, the higher the value of R. It follows that, for some threshold eigenvalue ratio r_{th}, if R for a candidate keypoint is larger than $(r_{\mathrm{th}}+1)^2/r_{\mathrm{th}}$, that key point is poorly localized and hence rejected. The new approach uses $r_{\mathrm{th}} = 10$.[3]

This processing step for suppressing responses at edges is a transfer of a corresponding approach in the Harris operator for corner detection. The difference is that the measure for thresholding is computed from the Hessian matrix instead of a second-moment matrix (see structure tensor).

3.2.6. Direction of Key-point

After the position and scale of the key-point is determined, the next step is key-point’s direction, which can ensure the feature’s rotation invariance. The direction is calculated by the image information of key-points neighborhood. Firstly, we give the expression of the module value (4) and phase (5).

m(x, y) = $\sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$ ….(5)

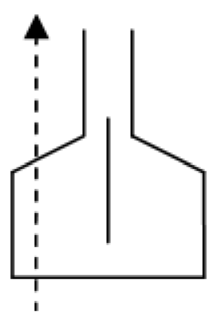
ɵ (x , y) = arctan [$\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}$] ……………………........(6)

Where, *L* is the scale of feature, and the calculating step is as follows:

- Choose a neighborhood ***M*** by the center of the key-point;
- Calculate the directions of points in ***M*** by (6);
- Obtain the direction distribution, and draw the statistical histogram;

4. The direction of key-point is the maximal component in the statistical histogram.

Since the result of (6) is a real number, we divide the real interval ([0,2π]) into 36 portions by π/18 so as to be convenient in obtaining the direction distribution, then the direction statistical histogram contains 36 phases, and the statistical rule is principle of proximity. Fig. 3.3 shows that the direction is determined by the statistical histogram, and the maximal component (π/2) is the key-point's direction.



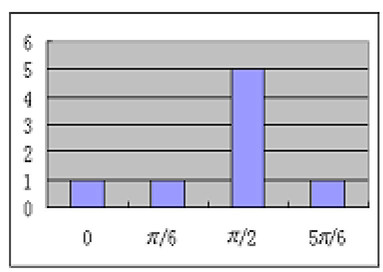


Figure3.3 Direction of key-point (a) Original image (b) Statistical histogram

3.2.8. Feature Description of Key-points

The position, scale and direction of key-point can only ensure the 2D geometric invariance, but can't ensure the lightness and view transformation invariance. So SIFT introduces feature description to solve those problems. The step is as follows:

- Select 16×16 pixel fields ***M*** in the neighborhood of key-point, and divide ***M*** into 16 subfields by 4×4;

- According to (5) and (6), calculate the direction and amplitude in every subfield, and then the direction distribution in the ranges of [0, 45, 90, 135, 180, 225, 270, 315] could be obtained.

- According to the direction statistical histogram of subfields, 8 direction descriptions will be calculated by the amplitude and Gaussian function;

- The feature description is obtained by connecting the direction descriptions of all subfields; the total of the direction descriptions is 16, so the length of the feature description is 128=16×8;
- In order to ensure the illumination invariance, the feature description should be normalized; and the Fig. 3.4 shows this process.

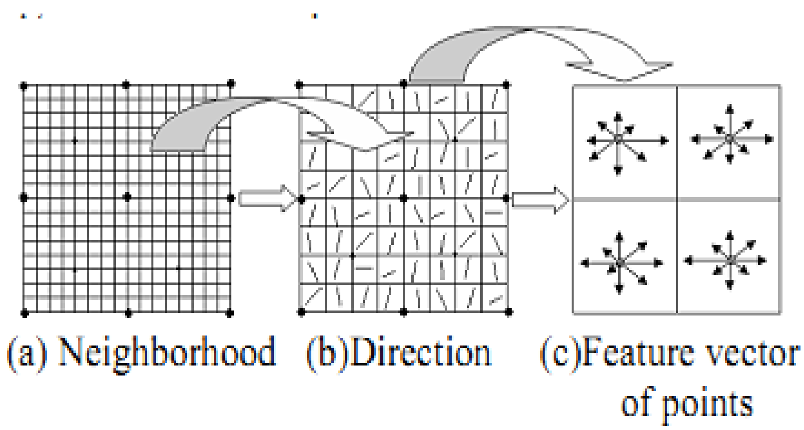


Figure 3.4 Feature descriptions of key-points

3.2.9. Feature Matching

The image matching is the feature matching. Firstly, the Euclidean distance between key-points from two pictures is calculated, and the two key-points with minimum distance is the candidate matching key-point pair. Secondly, calculate the ratio of the minimum distance: the second smallest distance; and if the ratio is greater than the preset threshold,

The candidate matching key-point pair is a good key-point pair. Thirdly, if the number of matching key-point is greater than the preset threshold, the two images are matched.

3.3 Classification

Image classification involves analyzing the numerical properties of various image features and organizing the data into categories. Usually, two phases of classification algorithms are employed: training and testing.

In the initial training phase, characteristic properties of typical image features are identified and, a unique description based on this of each classification category, i.e. training class, is created. The description of training classes is an extremely important component of the classification process.

The criteria for constructing training classes are that they are:

1. independent, i.e. a change in the description of one training class should not change the value of another,
2. discriminatory, i.e. different image features should have significantly different descriptions, and
3. Reliable, i.e. all image features within a training group should share the common definitive descriptions of that group.

In the next phase i.e. testing, accuracy of the classifier is measured. The accuracy can be determined by applying the classifier to an independent training set of objects with known classifications. Knowledge of the accuracy is necessary both in the application of the classifier and also in comparison of different classifiers.

A convenient way of building a parametric description of this sort is via a feature vector, where 'n' is the number of attributes which describe each image feature and training class. This allows us to consider each image feature is occupying a point, and a sub-space (i.e. a representative point surrounded by some spread, or deviation) is occupying by each training class, within the n-dimensional classification space. The classification problem is for determining to which sub-space class each feature vector belongs.

Classification can be linear or nonlinear. Linear classifiers are usually the fastest classifiers. It classifies the data on the basis of a linear combination of the characteristics based on which the classification is done. The classes are divided by a linear separator in the feature space. If the feature space is two dimensional, the separator is a line, it three dimensional the separator is a plane and if 'p' dimensional then the linear separator is a (p-1) dimensional hyper plane. Nonlinear classification is required when the class boundaries cannot be approximated well with linear hyper planes. Example of nonlinear classifier is K Nearest Neighbor.

3.3.1 Methods of classification

There are mainly two ways of classification, supervised and unsupervised. The difference between them lies in the fact that how the data is classified.

3.3.1.1 Supervised classification

In supervised classification, there are predetermined classes. Statistical processes (i.e. based on an a priori knowledge of probability distribution functions) or distribution free processes can be used to extract class descriptors. These classes can be regarded as a previously decided finite set. After classification, certain segment of data will be labeled with these classes. The task of the algorithm is to search for patterns and construct mathematical models. These models are then used to find out the measure of variance in the data and classify it. The examples of supervised classification are decision tree induction, naïve bayes classifier etc.

3.3.1.2 Unsupervised classification

Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In this type of classification, the classes are not pre decided. The basic task of the classifier is to develop the classes or the labels automatically. The algorithm is not told how the data is to be grouped; it is something it has to arrive at by itself. This is a difficult decision to make. The classifier looks for similarities between data and then determines which of these can form a group and can be classified under one label. The classes are also called clusters. The example of unsupervised classification is K-means classification. In K-means, the classifier is told in advance, the no of clusters have to be formed.

In unsupervised classification, since there is no defined strategy, the algorithm starts from one point and performs iterative repetitions to reach a stable configuration that makes sense. The results can vary widely and depend largely on the first few steps taken.

3.3.2 Examples of classifiers

The most common classifiers used are:

3.3.2.1 Support vector machine (SVM)

Support vector machine (SVM) [45] is an example of supervised learning classification. It is a binary linear classifier which takes an input and decides to which of the two classes it belongs. The classifier s first trained using a set of training examples. The training examples are pre marked as belonging to one of the two categories and based on these examples, the SVM classifier builds a model that assigns new examples (data to be tested) their suitable classes. The training examples are represented as points in space and are mapped such that there is a clear gap which divides the examples belonging to separate classes. The new examples are then mapped into the same space by analyzing to which of the two classes they suit better.

Apart from classification, a support vector machine can also be used for regression etc. It constructs a hyper plane that separates the two classes with gap as wide as possible. A hyper plane is regarded as ‘good’ if it has the largest distance with the nearest training data point of any class. This reduces the error of the classifier.

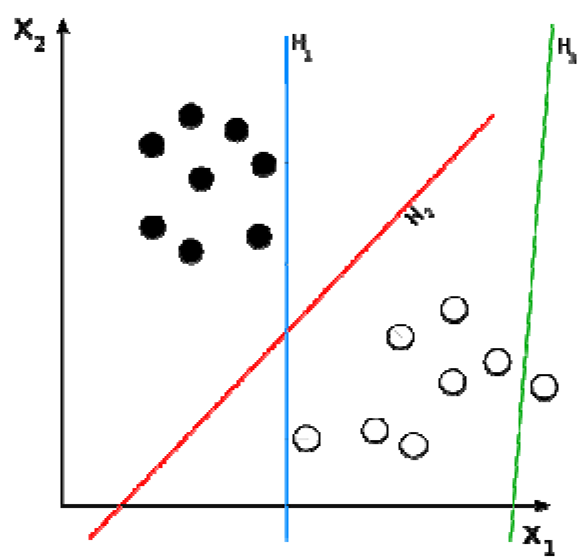


Figure 3.5 Hyper planes separating two classes

In the above figure, we see that although red and blue hyper planes both are separating the two classes entirely, the red hyper plane is doing it such that its distance from the nearest points of the two classes is maximum. Hence the red hyper plane is the most optimum hyper plane. The data points which are closest to the hyper plane are called support vectors.

Nonlinear SVM

Nonlinear SVM is applied in cases where the data sets have nonlinear decision boundaries. The trick applied here is to transform the data from its original coordinate space to a new space where a linear decision boundary can be used to separate the instances in the transformed space. Transformation of space may suffer from dimensionality problem which is associated with high dimensional data. Moreover, it is not always easy to find out what mapping must be used to ensure that a linear decision boundary can be constructed in the transformed space. This problem can be solved by 'Kernel' trick. It is a method for computing similarity in the transformed space using the original attribute set. The measure of similarity used is the dot product of the two input vectors. The similarity function computed in the original attribute space is called the 'kernel function'. The kernel trick eliminates the need to know the exact form of the mapping function.

Multi class SVM

Multi class SVM classifies data into more than one class. For multiclass case, we transform the problem into multiple binary classification problems.

3.3.2.2 Minimum distance classifiers

The minimum distance classifiers [46], as the name suggests, classify the data on the basis of the distance between the data points in the feature space. It classifies in such a manner, so as to minimize the distance between the data and the class in multi-feature space. The index of similarity here is the distance so that the minimum distance is identical to the maximum similarity. The most common distances often used in this procedure are:

Euclidian distance

The Euclidean distance is also known as the Euclidean metric that is distance between two points that is measured with a ruler, and is given by the Pythagoras formula. By using this formula as distance, Euclidean space or any other inner product space becomes a metric space. The norm of the Euclidian distance is called the Euclidean norm.

It is used in those problems where the variances of the population classes are different from each other. Theoretically the Euclidian distance is identical to the similarity index.

The Euclidean distance between two points 'p' and 'q' is the length of the line segment connecting them.

If p = (p_1, p_2,..., p_n) and q = (q_1, q_2,..., q_n) are two points in Euclidean n dimensional space, then the Euclidian distance from p to q is given by:

$$d(p, q) = d(q, p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 \dots \dots \dots \dots + (p_n - q_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \dots\dots\dots\dots\dots (7)$$

The position of a point in a Euclidean *n*-space is shown by a Euclidean vector. So, Euclidean vectors p and q, starting from the origin of the space. The Euclidean length, or Euclidean norm, or magnitude of a vector measures the length of this vector and is given by:

$$x(t), \frac{1}{\sqrt{A}} y\left(\frac{t-\tau}{A}\right), z(t) \dots\dots\dots\dots\dots\dots\dots\dots\dots (8)$$

Here the last equation involves the dot product.

Mahalanobis distance

Mahalanobis distance was introduced by P. C. Mahalanobis in 1936. Its basis is the correlations between variables by which different patterns can be identified and analyzed. It measures the similarity of an unknown sample set to a known one. It differs from Euclidean distance in the manner that it takes into account the correlations of the data set and is scale-invariant. To put in other words, it is a multivariate effect size.

The formal definition of the Mahalanobis distance of a multivariate vector

$$x=(x_1, x_2, \ldots \ldots \ldots, x_n)^T \text{is}$$

$$D_M(x) = \sqrt{(x-\mu)^T\, S^{-1}(x-\mu)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(9)$$

Where, μ is the mean vector represented as: $\mu = (\mu_{1,}\mu_{2,} \ldots \ldots \ldots \mu_{n,})^T$ and S is the covariance matrix.

Mahalanobis distance also called "generalized squared interpoint distance". Its squared value is defined as a dissimilarity measure between two random vectors *x* and *y* of the same distribution with the covariance matrix *S* as:

$$d\ (\bar{x},\bar{y}) = \sqrt{(\bar{x}-\bar{y})^T\, S^{-1}(\bar{x}-\bar{y})} \ldots\ldots\ldots\ldots\ldots\ldots(10)$$

If the covariance matrix is the identity matrix, the Mahalanobis distance becomes same as the Euclidean distance. On the other hand, if the covariance matrix is diagonal, the resulting distance measure is called the normalized Euclidean distance.

3.3.2.3 Artificial Neural networks (ANN)

Artificial neural network classifiers [47] are inspired by biological neural systems. The nerve cells in human brain are called neurons. These are linked with other neurons via strands of fiber called axons. Whenever the neurons are stimulated, axons transmit nerve impulses from one neuron to another. Extensions from the cell body of the neuron are called dendrites. Dendrites connect one neuron to the axons of other neurons. The connection between a dendrite and an axon is called a synapse. It has been discovered that the human brain learns by changing the strength of the synaptic connection between neurons when stimulated repeatedly by the same impulse.

The ANN has a structure analogous to the human brain. It is composed of an interconnected assembly of nodes and direct links. These models can be trained for the purpose of classification. The simplest model is called perceptron.

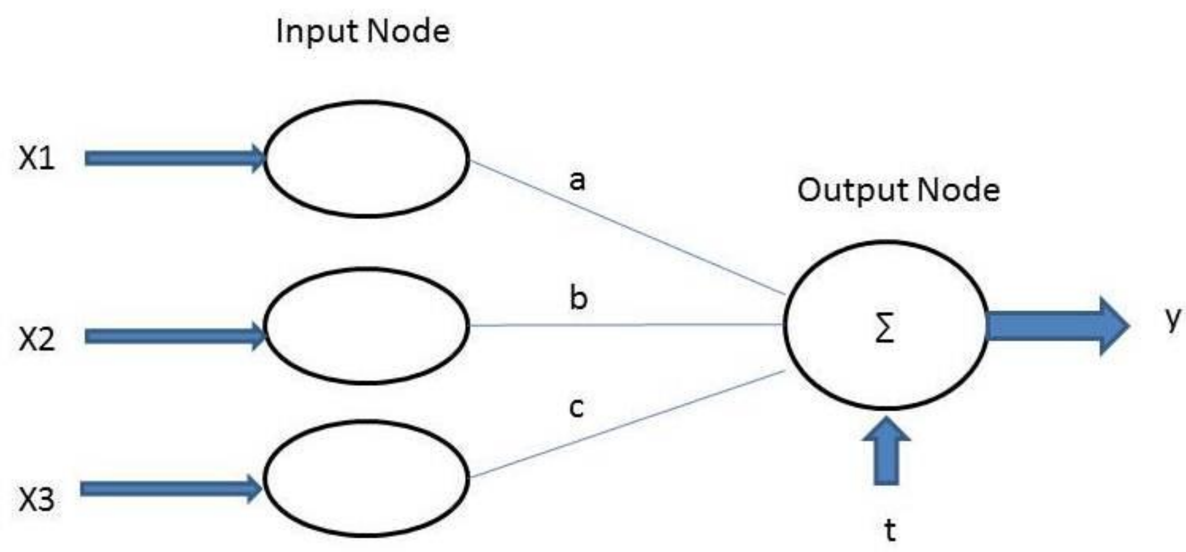


Figure 3.6 Model of a perceptron

The perceptron consists of two kinds of nodes, the input nodes and the output node. Input nodes are used to represent the input attributes and the output node represents the model output. The nodes in the neural network are called neurons are units. In a perceptron, each of the input nodes is connected to the output node via a weighted link. The weight of the link is used to emulate the strength of the synaptic connection between neurons. The perceptron computes the output value by calculating a weighted sum of the inputs, subtracting a bias factor ‘t’ from the sum and then examine the sign of the result. Training of the perceptron network involves changing and adapting the weight of the links until they fit the input output relationship of the training data i.e. the outputs of the perceptron become consistent with the true outputs of training data.

In an ANN model, an input node simply transmits the value it receives to the outgoing link without any transformation. On the other hand, the output node is a mathematical device which computes the weighted sum of its inputs subtracts the bias term and produces an output that depends on the sign of the result. The sign function is an activation function for the output neuron, it value is +1 if positive and -1 if negative. Once the model is trained, it can now be used to test new examples and classify them. The perceptron learning algorithm converges to an optimum solution for linearly separable classification problems. If the problem is not linearly separable the algorithm fails to converge and in that case, multilayer artificial neural network is needed.

A multilayer neural network has a more complex structure than a perceptron. It contains several intermediate layers between its input and output layers. These intermediate layers are called hidden layers and the nodes in these layers are called hidden nodes. There are two types of networks, feed forward and recurrent networks. In feed forward networks, the nodes in one layer are connected only to the nodes in the next layer while in a recurrent network; the links may connect the nodes in the same layer or in the previous layers. The activation function may be other than sign function also like linear or sigmoid function. These activation functions allow the hidden and the output nodes to produce nonlinear outputs. This type of complex structure can classify problems which are not linearly separable and have nonlinear solutions. This model will converge to the right solution when sufficient training data is provided.

3.3.2.4 K Nearest neighbor (KNN) classifier

In pattern recognition and classification, the k-nearest neighbor algorithm (k-NN) [48], is an algorithm for classifying data objects based on closest training examples in the feature space. The KNN is a type of instance based or lazy learning, where the function is only approximated locally and all computation is postponed till classification. The k-nearest neighbor algorithm is simplest among all machine learning algorithms: an object is classified by a majority vote of its neighboring data, and the test object is assigned to the class which is most common amongst its k nearest neighbors (k is a small positive integer). If 'k' is chosen to be 1, then the test object is simply assigned to the class to which its nearest neighbor belongs.

Although there is no explicit need for training in this algorithm, the neighbors can be regarded as training examples and are chosen from a set of objects for which the correct classification is known. The k-nearest neighbor algorithm is affected by the local structure
of the data. The nearest neighbor rules effectively compute the decision boundary in an implicit way.

The training examples can be regarded as vectors in a multidimensional feature space, each belonging to a class. So, in the training phase of the algorithm, it is required to only store the feature vectors of the training samples along with their class.

In the classification phase, k is a constant decided by the user and a test vector (can also be called query or a test point) is classified by assigning it the label which is most common among the k training data nearest to that particular test point.

Euclidean distance is most commonly used as the distance measure; however this only applies to the continuous variables. For other types of classification like text classification, other measures such as the overlap metric (or

Hamming distance) can also be used. The accuracy of the k nearest neighbor algorithm can be improved significantly by using special algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

The concept of ‘majority voting’ for classification has one drawback. If more examples in the training sample belong to one class, it tends to dominate the prediction of test sample, since their chances of being present in the k nearest neighbors is high owing to their large numbers. So the training examples have to be chosen very carefully. This problem can also be overcome by taking into consideration the distance from the test vector to each of the k nearest neighbors.

Choosing the value of ‘k’ also depends on the data. Larger values of ‘k’ help in reducing the effect of noise by averaging it out. But it makes the boundaries between the classes less distinct. There are various techniques for selecting the optimum value of ‘k’ like cross validation. The special case when the value of ‘k’ is ‘1’, i.e. the testing example is assigned the class to which its nearest neighbor belongs, is called the nearest neighbor algorithm.

Chapter 4

EXPERIMENTSAND RESULTS

Scale invariant Feature transform is used for the purpose of feature extraction. The standard ORL database is used for conducting all the experiments. It consists of images of 20 subjects of size 92 x 112 as shown in Figure 4.1

Figure 4.1 Subjects of the ORL database

Each of the 20 subjects has 10 orientations. Some orientations of sample image (subject 2) are shown in figure 4.2.

Figure 4.2 Different orientations of subject 2 in ORL database

There are variations in facial expression (open/closed eyes, smiling/non-smiling.), facial details (glasses/no glasses) and scale (variation of up to about 10 %). All the images were taken against a dark homogenous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees.

For all the methods implemented, 8 orientations (of each subject) were used for training and two orientations for testing. The entire programming is done using the software 'MATLAB'. An introduction to the programming in matlab has been provided in appendix A.

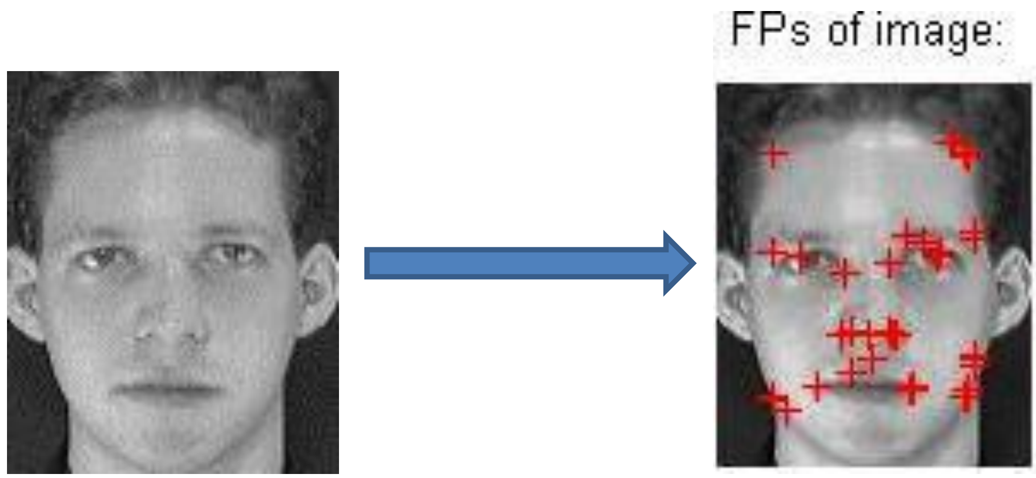
4.1 Image preprocessing

Image normalization is done in order to reduce computational overhead and improve the performance of the method. The images in the ORL database are gray scale images which contain intensity values ranging from 0 to 255. The pixel values of all the images are divided by the maximum pixel value of the image. The value of all the pixels lies between 0 and 1.

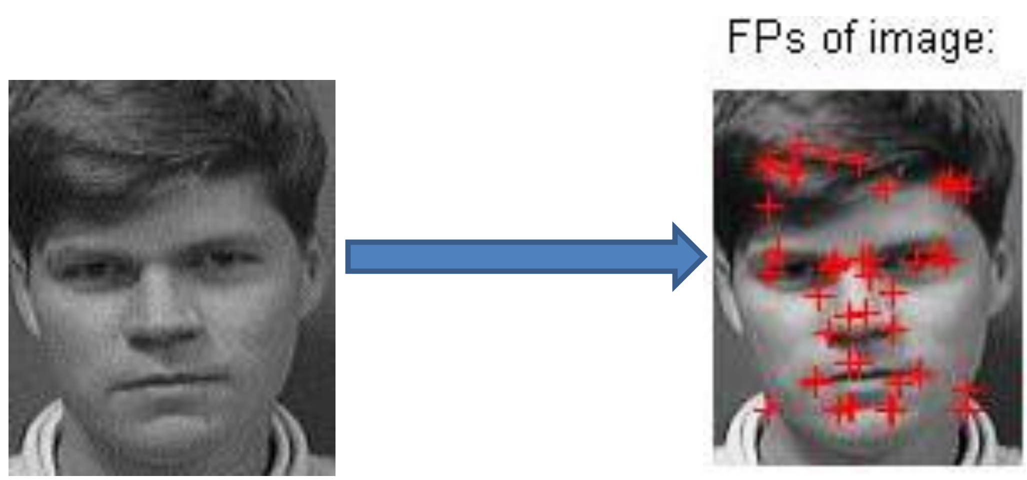
4.2 Feature extraction

Feature extraction of all the images is done using the SIFT. Some feature images are shown below:

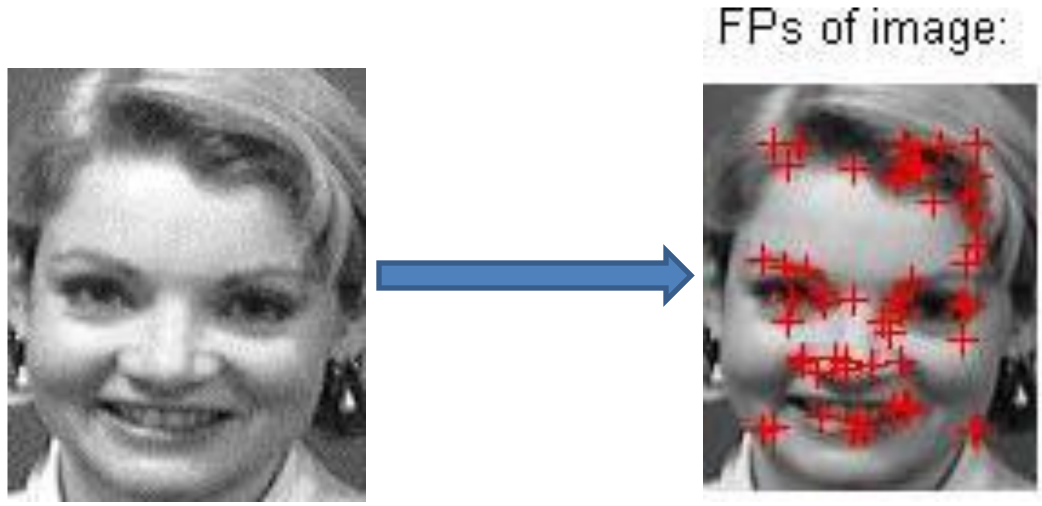
1) subject 1



2) Subject 3



3) Subject 5



4) subject 7

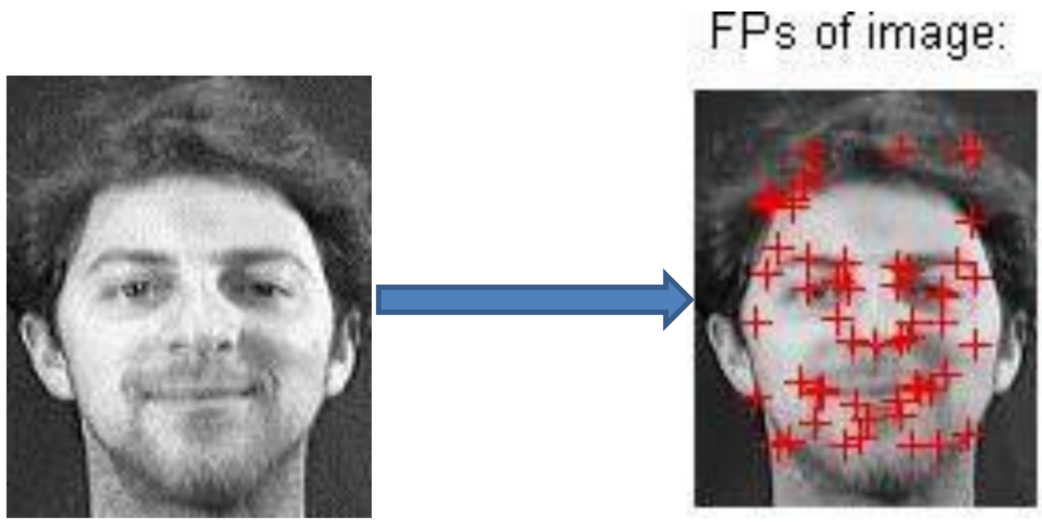
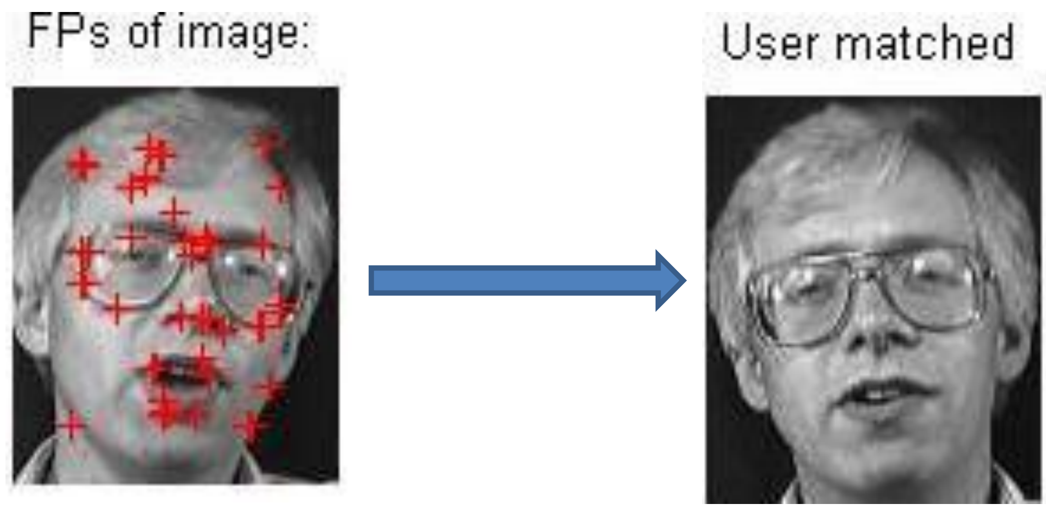


Figure 4.3 Feature point detected from training image 1, 3, 5 and 7

4.3 Result:

4.3.1 Matching with pose image

a) Subject 2



b) Subject 9

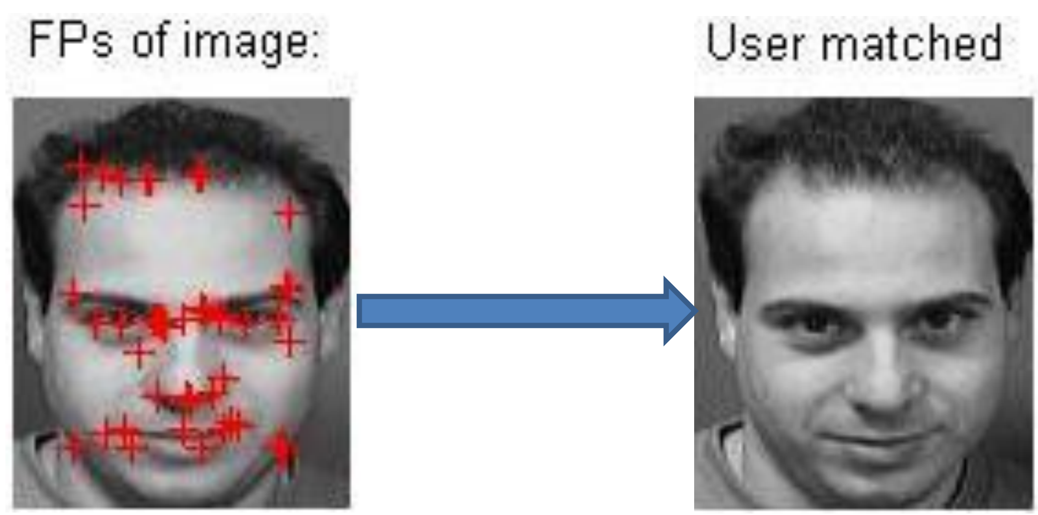
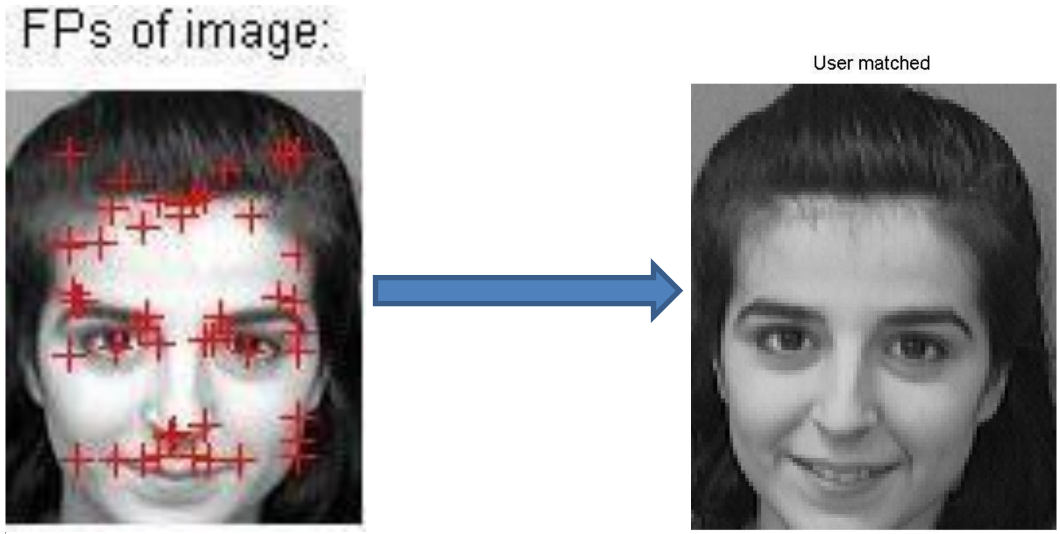


Figure 4.4 matching with pose image with subject 2 and 9

4.3.2 Matching result with illumination or intensity variation

a) Subject 8

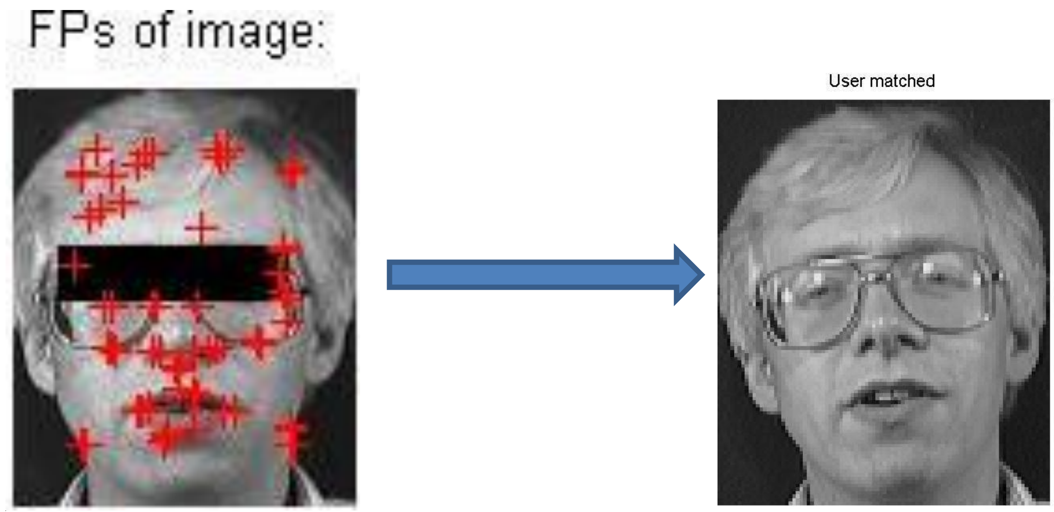


b) Subject 12

Figure 4.4 matching with pose image with subject 8 and 12

4.3.3 Matching result with Partial Occlusion

a) Subject 4



b) Subject 1

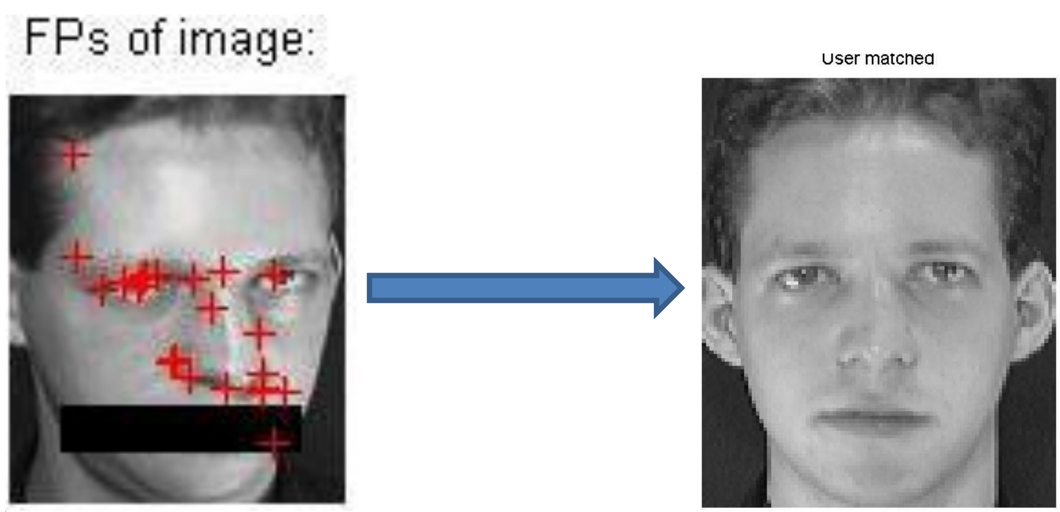


Figure 4.5 matching with pose image with subject 4 and 1

The confusion matrix for Test image of 20 subjects using SIFT and classified using k nearest neighbor’s classifier is presented below:

Test \ subject	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1/9	√																			
2/10		√																		
3/9			x																	
4/9				√																
5/10					√															
6/10						√														
7/10							√													
8/10								√												
9/9									√											
10/9										√										
11/10											√									
12/9												√								
13/9													√							
14/10														√						
15/10															√					
16/10																√				
17/9																	√			
18/9																		√		
19/10																			√	
20/9																				√

Table 1: confusion matrix of test image

4.3.4 Performance measures Recognition systems result in two types of errors: a False Acceptance (FA), which occurs when the system accepts an impostor face, or a False Rejection (FR), which occurs when the system refuses a true face. The performance is generally measured in terms of False Acceptance Rate (FAR) and False Rejection Rate (FRR), defined as:

$$\text{FAR} = \frac{\text{number of False acceptance}}{number\ of\ impstor\ face\ presentations}$$

$$\text{FRR} = \frac{number\ of\ False\ rejection}{number\ of\ true\ face\ presentations}$$

To aid the interpretation of performance, the two error measures are often combined using Equal Error Rate (EER)[49], when FAR=FRR on a particular database.

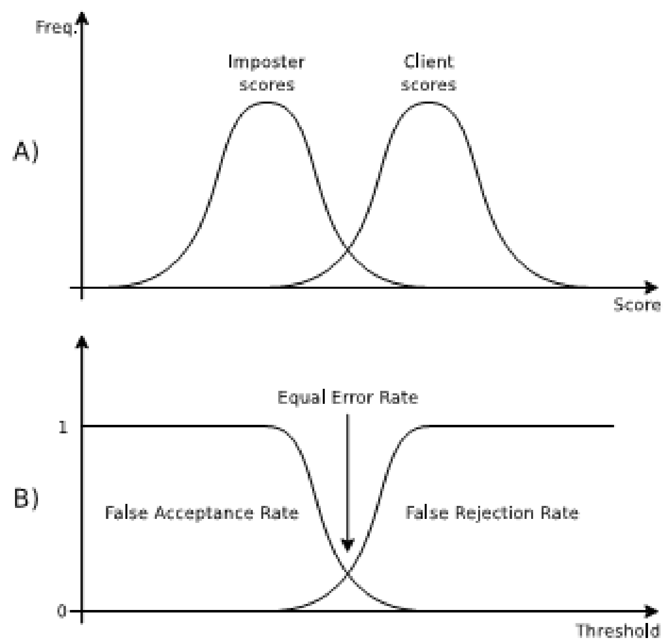


Figure 4.6 FAR vs FRR curve

Figure shows the Relation of False Acceptance Rate (FAR), False Rejection Rate (FRR) with the distribution of clients, imposters in a verification scheme. A) Shows the imposters and client populations in terms of the score (high score meaning high likelihood of belonging to the client population). B) The associated FAR and FRR, the Equal Error Rate (EER) is where the FAR and FRR curve meets and gives the threshold value for the best separability of the imposter and client classes.

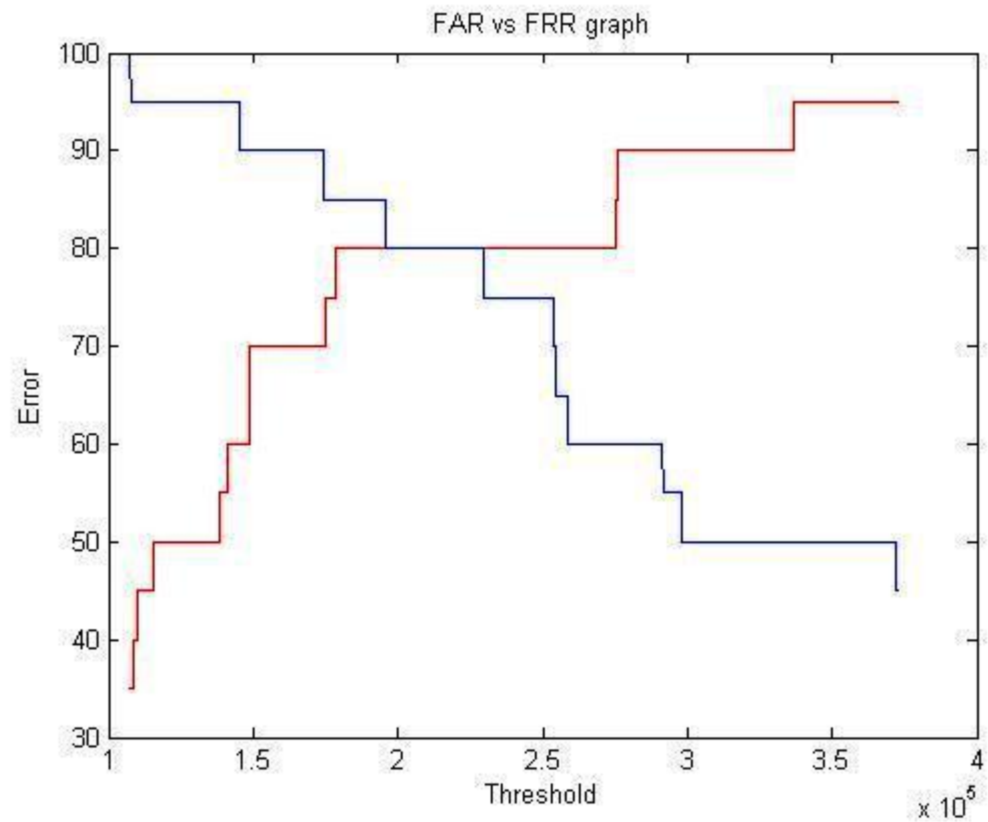


Figure 4.7 Experimental curve of FAR vs FRR

Figure 4.7 shows the FAR vs FRR curve. We can observe that the values of FAR and FRR intersects at point, defined as Equal error rate.

Methods	Fisher face	Hankan’s	SIFT+SVM	Proposed method
Equal error rate	4.58	2.61	2.58	0.80

Table 2. EER on ORL Database

Table 2 summarizes a comparative analysis on the ORL database our proposed method has minimum EER this show the robustness of the technique against face expression, illumination changes, pose changes and partial occlusion.

Table 3 shows the Recognition rate in different algorithm.

Methods	PCA	ICA	FISHER	2D_PCA	SIFT	PROPOSED METHOD
RT%	92.1	91.6	92.8	92.5	96.3	97.91

Table 3. The rate of recognition in different algorithm

Table 3 shows the RT% (Recognition rate) in [50] various algorithm. It is seen that this proposed method give better result than previous work.

Chapter 5

Conclusion and Future work

5.1 Conclusion

In the presented work, a novel method for face recognition using SIFT (Scale invariant Fourier transform) has been proposed. This method is invariant to pose, expression, illumination and partial occlusion. In our result we have observed the Equal error rate is 0.80 that should be below 0.89, obtained by using SIFT with SVM [49].the Recognition rate is found 97.91% that show Robustness of this method. It shows all the matching between training images and test images. We have done the matching technique by using k nearest neighbors' classifier. The method is applied on ORL Database. We found that Face Recognition using SIFT technique is robust and invariant to pose, expression, illumination and partial occlusion.

5.2 Future scope

As one of the most successful applications of image analysis and understanding, At present face recognition has gained so much attention. it has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding in last decade.

Face recognition is a technology just reaching sufficient maturity for it to experience a rapid growth in its practical applications. Much research effort around the world is being applied for expanding the accuracy and capabilities of this domain, with a consequent broadening of its application in the near future. Verification systems for physical and electronic access security are available today, but the future holds the promise and the threat of passive customization and automated surveillance systems enabled by face recognition.The largest example of face recognition systems in the world operates in the U.S. Department of State for visa processing. It consists of over 75 million photographs. There can be several other uses of face recognition in future. Some of them can be:

1. Face recognition can be used to prevent ATM frauds. A database of all ATM customers can be prepared with the banks. High resolution cameras will have to be deployed at all ATMs and face recognition software has to be used. Now, whenever a user will enter the ATM, his photograph will be taken and access to the ATM machine will be permitted only after the photo is matched with the database.

2. Duplicate votes are being reported in India. To prevent this, all voters database, all constituencies, will have to be prepared. While voting, a camera installed at the voting site will capture a photo of the person and match it with the database. The person will be allowed to vote only after his photo matches with the database. If it is found that the person has already voted once, then he can be prevented from voting twice.

3. Similarly, passport and visa verifications can also be done using face recognition technology. Driving license verification can also be performed using face recognition technology as mentioned earlier.

4. For identifying and verifying terrorists at airports, railway stations and shopping malls also, face recognition technology will be the best choice in India as compared with other biometric technologies since other technologies cannot be helpful in crowds.

Along with all the advantages, face recognition technology has its own problems. With the widespread installation of security cameras and the increasing financial and technological feasibility of automating this surveillance, public fears have also increased about the potential for invasion of privacy that this technology can bring about.

REFERENCES

1) X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang. Face recognition using Laplacian faces. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3):328–340, 2005.
2) J. Daugman. How iris recognition works. Proceedings of 2002 International Conf. on Image Processing, 1, 2002.
3) B. Moghaddam and M. H. Yang, "Learning Gender with Support Faces," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, pp.707-711, 2002.
4) Y. Shinohara and N. Otsu, "Facial Expression Recognition Using Fisher Weight Maps," in Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Vol.100, 2004, pp.499-504.
5) Ce Wang and Michael S. Brandstein, "A HYBRID REAL-TIME FACE TRACKING SYSTEM", Division of Engineering and Applied Sciences Harvard University Cambridge, MA 02138
6) Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection", IEEE Transactions On Pattern Analysis And Machine Intelligence, Vol. 19, No. 7, July 1997
7) M. Turk, A. Pentland. ”Eigenfaces for Recognition”. Journal of Cognitive Neuroscience. Vol 3, No. 1. 71-86, 1991.
8) M. H. Yang. Kernel Eigenfaces vs. Kernel Fisher faces: Face recognition using kernel methods. Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition , pages 205–211, 2002.
9) L. Zhi-fang, Y. Zhi-sheng, A.K.Jain and W. Yun-qiong, 2003, “Face Detection and Facial Feature Extraction in Color Image”, *Proc. The Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA’03),* pp.126-130, Xi’an, China.

10) K. Seo, W. Kim, C. Oh and J. Lee, 2002, “Face Detection And Facial Feature Extraction Using Color Snake”, *Proc. ISIE 2002 - 2002 IEEE International Symposium on Industrial Electronics*, pp.457-462, L 'Aquila, Italy.

11) J. Ruan and J. Yin, 2009, “Face Detection Based on Facial Features and Linear Support Vector Machines”, *Proc. 2009 International Conference on Communication Software and Networks*, pp.371-375, Macau, China.

12) L. Zhao, X. Sun and X. Xu, 2006, “Face Detection Based On Facial Features”, *Proc. ICSP2006*, Guilin, China.

13) C. Lin, 2005, “Face Detection by Color and Multilayer Feed forward Neural Network”, *Proc. 2005 IEEE International Conference on Information Acquisition*, pp.518-523, Hong Kong and Macau, China.

14) M. A. Berbar, H. M. Kelash and A. A. Kandeel, 2006, “Faces and Facial Features Detection in Color Images”, *Proc. Geometric— Modeling and Imaging New Trends (GMAI'06)*, pp.209-214, London, UK.

15) S. Kherchaoui and A. Houacine, 2010, “Face Detection Based On A Model Of The Skin Color With Constraints And Template Matching”, *Proc. 2010 International Conference on Machine and Web Intelligence*, pp. 469 - 472, Algiers, Algeria.

16) D. Huang, T. Lin, C. Ho and W. Hu, 2010, “Face Detection Based On Feature Analysis And Edge Detection Against Skin Color-like Backgrounds”, *Proc. 2010 Fourth International Conference on Genetic and Evolutionary Computing*, pp.687-690, Shenzen, China.

17) J. Qiang-rong and L. Hua-lan, 2010, “Robust Human Face Detection in Complicated Color Images”, *Proc. 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, pp.218 – 221, Chengdu,

18) Z. Li, L. Xue and F. Tan, 2010, “Face Detection In Complex Background Based On Skin Color Features And Improved Adaboost Algorithms”, *Proc. 2010 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp.723 – 727, Shanghai, China.

19) C. Aiping, P. Lian, T. Yaobin and N. Ning, 2010, “Face Detection Technology Based On Skin Color Segmentation And Template Matching”, *Proc. 2010 Second International Workshop on Education Technology and Computer Science*, pp.708-711, Wuhan, China.

20) P. Peer, J. Kovac and F. Solina, 2003, “Robust Human Face Detection in Complicated Color Images”, *Proc. 2010 The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, pp. 218 – 221, Chengdu, China.

21) W. Chen, T. Sun, X. Yang and L. Wang, 2009, “Face Detection Based On Half Face-Template”, *Proc. The Ninth International Conference on Electronic Measurement & Instruments ICEMI’2009*, pp. 4-54-4-59, Beijing, China.

22) Istanbul, Turkey. J. Wang and H. Yang, 2008, “Face Detection Based On Template Matching And 2DPCA Algorithm”, *Proc. 2008 Congress on Image and Signal Processing*, pp.575-579, Hainan, China.

23) A. R. Mohan and N. Sudha, 2009, “Fast Face Detection Using Boosted Eigenfaces”, *Proc. 2009 IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009)*, pp.102-1006, Kuala Lumpur, Malaysia.

24) D. Anijiantis, E. Dernzatas and G. Kukkinakis, 1999, “A Neural Network Method For Accurate Face Detection On Arbitrary Images”, *Proc. The 6th IEEE International Conference on Electronics, Circuits and Systems*, pp. 109 - 112 vol.1, Pafos, Cyprus.

25) C. Anagnostopoulos, I. Anagnostopoulos, D. Vergados, I. Papaleonidopoulos, E. Kayafas, V. Loumos and G. Stasinopoulos, 2002, “A Probabilistic Neural Network For Face Detection On Segmented Skin Areas Based On Fuzzy Rules”, *Proc. IEEE MELECON 2002*, pp.493-497, Cairo, Egypt.

26) X. Liu, G. Geng and X. Wang, 2010, “Automatically Face Detection Based On BP Neural Network And Bayesian Decision”, *Proc. 2010 Sixth International Conference on Natural Computation (ICNC 2010)*, pp.1590-1594, Shandong, China.

27) M. Tayyab and M. F. Zafar, 2009, "Face Detection Using 2D-Discrete Cosine Transform And Back Propagation Neural Network", *Proc. 2009 International Conference on Emerging Technologies,* pp.35-39, Islamabad, Pakistan.

28) K. A. A. Aziz and R. A. Ramlee, 2009, "Face Detection Using Radial Basis Function Neural Networks With Variance Spread Value", *Proc. 2009 International Conference of Soft Computing and Pattern Recognition*, pp.399-403, Malacca, Malaysia.

29) C. Shavers, R. Li and G. Lebby, 2006, "An SVM-Based Approach To Face Detection", *Proc. 38th Southeastern Symposium on System Theory*, pp.362-366, Tennessee Technological University, Cookeville, TN, USA.

30) H. Jee, K. Lee and S. Pan, 2004, "Eye And Face Detection Using SVM", *Proc. ISSNIP 2004,* pp.577-580, Melbourne, Australia.

31) G. Wang and Z. Ou, 2006, "Face Recognition Based on Image Enhancement and Gabor Features", *Proc. 6th World Congress on Intelligent Control and Automation,* pp.9761-9764, Dalian, China.

32) Y. Song, Y. Kim, U. Chang and H. B. Kwon, 2006, "Face Recognition Robust To Left-Right Shadows Facial Symmetry", *Pattern Recognition Vol.39 (2006),* pp.1542–1545.

33) M. I. Razzak, M. K. Khan, K. Alghathbar and R. Yousaf, 2010, "Face Recognition Using Layered Linear Discriminant Analysis And Small Subspace", *Proc. 2010 10th IEEE International Conference on Computer and Information Technology (CIT 2010),* pp.1407-1412, West Yorkshire, UK.

34) F. Wang, J. Wang, C. Zhang and J, Kwok, 2007, "Face Recognition Using Spectral Features", *Pattern Recognition Vol.40 (2007),* pp.2786–2797.

35) J. Youyi and L. Xiao, 2010, "A Method for Face Recognition Based on Wavelet Neural Network", *Proc. 2010 Second WRI Global Congress on Intelligent Systems*, pp.133-136, Wuhan, China.

36) K. Youssef and P. Woo, 2007, "A New Method for Face Recognition

Based on Color Information and a Neural Network", *Proc. Third International Conference on Natural Computation (ICNC 2007)*, pp.585 – 589, Hainan, China.

37) A. Rida and Dr BoukelifAoued, 2004, "Artificial Neural Network-Based Face Recognition", *Proc. First International Symposium on Control, Communications and Signal Processing*, pp.439 – 442, Hammamet, Tunisia.

38) D.N Pritha, L. Savitha and S.S. Shylaja, 2010, "Face Recognition By Feedforward Neural Network Using Laplacian Of Gaussian Filter And Singular Value Decomposition", *Proc. 2010 First International Conference on Integrated Intelligent Computing*, pp.56-61, Bangalore, India.

39) X. Wang, Q. Ruan and Y. Ming, 2010, "3D Face Recognition Using Corresponding Point Direction Measure And Depth Local Features", *Proc. ICSP 2010*, pp.86-89, Beijing, China.

40) J. Harguess, S Gupta and J. K. Aggarwal, 2008, "3D Face Recognition With The Average-Half-Face", *Proc. 19th International Conference on Pattern Recognition ICPR 2008*, pp.1-4, Florida, USA.

41) Y. Song, W. Wang and Y. Chen, 2009, "Research On 3D Face Recognition Algorithm", *Proc. 2009 First International Workshop on Education Technology and Computer Science*, pp.47-50, Wuhan, China.

42) D. Lee and J. Liang, 2010, "A Face Detection and Recognition System Based on Rectangular Feature Orientation", *Proc. 2010 International Conference on System Science and Engineering*, pp.495-499, Taipei, Taiwan.

43) P. Zhang, 2007, "A Video-based Face Detection and Recognition System using Cascade Face Verification Modules", *Proc. 37th IEEE Applied Imagery Pattern Recognition Workshop AIPR '08*, pp.1-8, DC, USA.

44) Han Yanbin, Yin Jianqin, Li Jinping, "Human Face Feature Extraction and Recognition Base on SIFT", 2008 International Symposium on Computer Science and Computational Technology.

45) Xiao-Lei(Celina) Xia and Kang Li, “A Sparse Multi-class Least-Squares Support Vector Machine’’, IEEE 2008.
46) H. Lin and A. N. Venetsanopoulos, “A Weighted Minimum Distance Classifier for Pattern Recognition’’, IEEE 1993.
47) Dar-Shyang Lee and Sargur N. Srihari, “A Theory of Classifier Combination: The Neural Network Approach’’,IEEE 1995.
48) Padraig Cunningham and Sarah Jane De lany, “ k -Nearest Neighbor’s Classifier’’, Technical Report UCD-CSI-2007-4 March 27, 2007
49) Lichun Zhang, Junwei Chen, Yue Lu, and Patrick Wang, “ Face Recognition Using Scale Invariant Feature Transform and Support Vector Machine’’, Department of Computer Science & Technology East China Normal University, College of Computer and Information Science, Northeastern University Boston, MA 02115, USA
50) Han Yanbin, Yin Jianqin, Li Jinping, “Human Face Feature Extraction and Recognition Base on SIFT’’2008 International Symposium on Computer Science and Computational Technology.

Appendix A

Abbreviations

1) ANN - Artificial Neural Network
2) BLD - Bayesian Logistic Discriminant
3) HMM - Hidden Markov Model
4) KNN - K nearest Neighbor
5) LDA - Linear Discriminant Analysis
6) ORL - Olivetti Research Laboratories
7) RGB - Red Green Blue
8) SVM - Support Vector Machine

Appendix B

An Introduction to Image Processing in Matlab

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expresses in familiar mathematical notation. It is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations in a fraction of the time it would take to write a program in a scalar non interactive language such as C, C++ or JAVA.

Image formats supported by Matlab

The following image formats are supported by Matlab:

- BMP
- HDF
- JPEG
- PCX
- TIFF
- XWB

Working formats in Matlab

If an image is stored as a JPEG-image on your disc we first read it into Matlab. However, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format.

Intensity image (gray scale image)

It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel.

Binary image

This image format also stores an image as a matrix but can only color a pixel black on white (and nothing in between). It assigns a 0 for black and a 1 for white.

Indexed image

This is a practical way of representing color images. An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

RGB image

This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use.

Reading Image Files

The command to read an image from file filename and store it in matrix variable p is:

p = imread('filename');

The filename may be an absolute pathname or a relative pathname from the current working directory. Omitting the ';' at the end of the command causes the value to be printed to the command window.

Writing Image Files

The command to write an image from variable p and store it in file filename is:

imwrite (p, 'filename');

The filename may be an absolute pathname or a relative pathname from the current working directory.

Displaying Image Files

The command to display the image from variable p to a figure window is:

Imshow (p);

An additional parameter may be used to set the number of display levels or set the range of display levels. Various controls, such as dynamic display of index and value for the cursor position, are available in the image display tool.

The figure command can be used to create a new current figure for the display:

figure, imshow (p);

Standard Arrays

MATLAB has standard arrays for creating arrays of a defined size with zeros, ones, true, false, or random values. For example: p = zeros(M,N);

Major built in Functions (used in this work) are:

1) Classify

Its format is:

class = classify (sample, training, group)

It classifies each row of the data in sample into one of the groups in training. Group is a grouping variable for training. ‘Class’ indicates which group each row of SAMPLE has been assigned to, and is of the same type as ‘group’.

2) Factorial Its format is: factorial (n)

This command calculates the factorial on ‘n’ which is the product of all the integers from 1 to n.

3) Bispeci

It calculates the bi spectrum using the indirect method. This command is in ‘HOSA’ toolbox.

Its format is:

[Bspec,waxis] = bispeci (y,nlag,segsamp,overlap,flag,nfft, wind)

Here, ‘y’ is the data vector or time-series, ‘nlag’ is the number of lags to compute, ‘segsamp’ is the number of samples per segment, ‘overlap’ is the percentage of overlap, ‘flag’ stands for ‘biased' or 'unbiased' , ‘ nfft’ is the FFT length to use, ‘wind’ is the window function to apply.

The output ‘Bspec’ is the estimated bispectrum and it is an nfft x nfft array with origin at the center, and axes pointing down and to the right. ‘Waxis’ is the frequency-domain axis associated with the bispectrum.

4) Size

Its format is:

[M, N] = SIZE(X)

It calculates the size of matrix X and returns the number of rows and columns in X as separate output variables.

5) Sort

Its format is:

Y = SORT(X, DIM, MODE)

This command sorts in ascending or descending order. ‘Dim’ selects a dimension along which to sort. ‘Mode’ selects the direction of the sort, 'ascend' results in ascending order and 'descend' results in descending order. The result is in Y which has the same shape and type as X.