A dissertation submitted in partial fulfilment of the requirements for M.Tech(Computer Science) degree of the Indian Statistical Institute

Human Face Recognition

M.Tech(Computer Science) Dissertation Report

By

Sourav Das

Roll Number : CS0904

Under the supervision of

Prof. Bhabatosh Chanda

Electronics and Communication Sciences Unit

Indian Statistical Institute

203, B.T. Road

Kolkata 700 108

Indian Statistical Institute

Kolkata 700 108

CERTIFICATE

This is to certify that the thesis entitled "Human Face Recognition" is submitted in the partial fulfilment of the degree of M. Tech. in Computer Science at Indian Statistical Institute, Kolkata. It is fully adequate in scope and quality as a dissertation for the required degree.

The thesis is a faithfully record of bonafide research work carried out by Sourav Das under my supervision and guidance. It is further certified that no part of this thesis has been submitted to any other university or institute for the award of any degree or diploma.

Prof. Bhabatosh Chanda
(Supervisor)
ECSU Unit

Countersigned
(External Examiner)
Date:

Acknowledgement

I also take this opportunity to express my sincere gratitude to the supervisor of this study, Prof. Bhabatosh Chanda. His command over this matter has been a great help for my analysis. He helped me in all aspects including the preparation of this manuscript. This work has been possible only because of his continuous suggestions, inspiration, motivation and full freedom given to me to incorporate my ideas. I also take this opportunity to thank all my teachers who have taught me in my M. Tech. course and last but not the least I thank all my family members and friends for their endless support.

Place : Kolkata
Date :

Sourav Das

Abstract

In today's world identification of a person is very important and necessary mostly in security purpose. Now a days many organisation use different type of identification techniques, one of them is Face Recognition. But in different conditions they may not have the capability to recognise efficiently with respect to both success rate and time complexity of their algorithm. There are different as well as important issues in face recognition like illumination, pose, expression, ageing, occlusion that may make trouble to any recognition system.

In this report we have described our own approach of recognizing human face which is very much robust under inconsistently illumination, pose, expression. The idea behind this approach is to fuse different local features, extracted from the face images. For classification, we use Nearest Neighbour and Support Vector Machine. To verify the algorithm we use ORL, Extended Yale B, AR database, MATLAB and LIBSVM tool in windows system.

Contents

Chapter 1

Introduction

Face recognition is a computer application for automatically identifying or verifying a person from a digital image or a snapshot from a video source. This is a branch of a most popular and advanced topic **Biometrics** [1]. Facial scan is an effective biometric attribute or indicator. Different biometric indicators are suited for different kinds of identification applications due to their variations in intrusiveness, accuracy, cost and ease of sensing. There are six main biometric indicators namely face, finger, hand, voice, eye and signature. Among those facial features scored the highest compatibility, in a **machine readable travel documents (MRTD)** system based on a number of evaluation factors [2].

Much of the work in face recognition by computers has focused on detecting individual features such as the eyes, nose, mouth and head outline, and defining a face model by the position, size, and relationships among these features. Such approaches have proven to depend on the precise features.

1.1 History of face recognition

The most famous of early example of a face recognition system is due to Kohonen[3], who demonstrated that a simple neural net could perform face recognition for aligned and normalized face images. The type of network he employed computed a face description by approximating the eigenvectors of the face image's auto-correlation matrix; these eigenvectors are now known as eigenfaces. Kohonen's system was not a practical success, however, because of the need for precise alignment and normalization. Kirby and Sirovich (1989)[4] later introduced an algebraic manipulation (PCA) which made it easy to directly calculate the eigenfaces, and showed that fewer than 100 were required to accurately code carefully aligned and normalized face images. Turk and Pentland (1991)[5] then demonstrated that the residual error when coding using the eigenfaces could be used both to detect faces in cluttered natural imagery, and to determine the precise location and scale of faces in an image. They then demonstrated that by coupling this method for detecting and localizing faces with the eigenface recognition method, one could achieve reliable, real-time recognition of faces in a minimally constrained environment.

Face recognition scenarios can be classified into two types :

- **Face Verification or Authentication** is a one-to-one match that compares a query face image against a template face image whose identity is being claimed.
- **Face Identification or Recognition** is a one-to-many matching process that compares a query face image against all the template images in a face database to determine the identity of the query face.

Issues in Face Recognition

- 3D face projected into 2D image
- Illumination variation because of different lighting condition
- Facial expression
- Occlusion due to other objects or accessories(e.g. sunglasses, scarf etc.)

- Ageing is another great issue as the face changes over time in a non linear fashion over long periods.

1.2 Different Approaches of Face Recognition

Finding efficient facial features to represent the face appearance is the most critical aspect in face recognition. Facial features fall into two classes : **global feature** and **local feature**. In global feature extraction process, the whole image is taken into account, but local feature considers only the local region within the image. **Principal Component Analysis (PCA)**, **Linear Discriminant Analysis (LDA)** etc., generates global features that have been widely used. Many systems based on PCA or LDA have been developed for face recognition and verification, and comparative advantages of such methods have been studied in detail for face recognition. Both methods are sensible to scale variations. Although global features based face-recognition and verification systems have proven to be reliable in ideal environments, they can be very sensitive to real environmental conditions. As an example, the effectiveness of PCA-based face recognition and verification strongly depends on lighting conditions and on variations in the subject's pose in front of the camera.

Hence local features are gaining more attention for their robustness in uncontrolled environment. Many researchers have used the **gray-level co-occurrence matrix**[6] for the extraction of texture features to be used in texture classification. Gelzinis et al. [7] presented a new approach to exploiting information available from the co-occurrence matrices computed for different distance parameter values. This co-occurrence matrix and its extracted features can be used in face recognition. Ahonen et al.[8] proposed **Local Binary Pattern** that provides an illumination-invariant description of face image. Very recently, a **Local Directional Pattern**[9] is being used which overcomes the draw-backs of LBP and which is more robust in recognizing faces.

1.3 Face Recognition System

A schematic block diagram of a face recognition system is shown in Figure 1.1.
Important modules of the system include:

1. Face detection module,
2. Face alignment module,
3. Feature extraction module,
4. Similarity measure and matching module

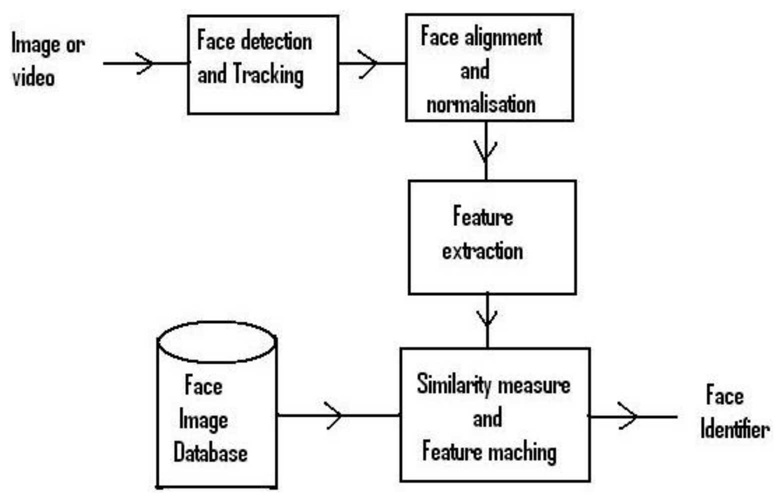

Figure 1.1: Schematic diagram of a face recognition system

A brief description of each module is given below. Once the training is over, the system takes in the query image, processes it, extracts relevant features and finally, based on the images in the face image database, the system gives out its decision automatically.

1.3.1 Face Detection

Face Detection module detects or, in other words, separates out human faces from the non-face objects present in an image. The image may be captured

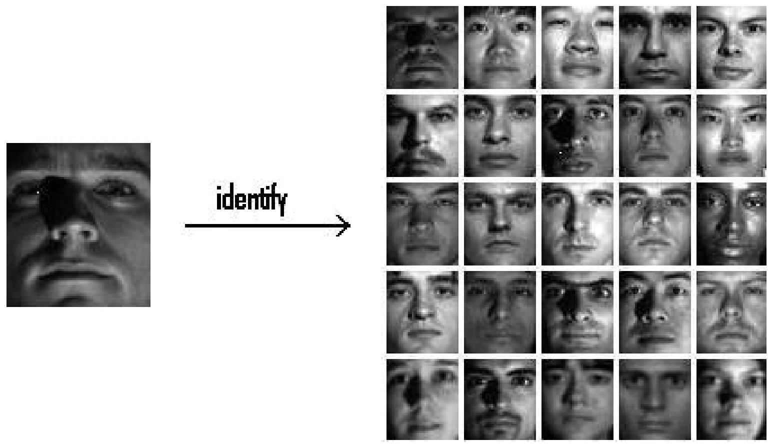

Figure 1.2: Images from the Extended Yale Face Database B

either in a controlled environment (probably in a studio or dedicated room with uniform lighting, against a constant background, consisting of a single object, at more or less predefined position etc.) or in an uncontrolled environment (probably in drawing room, office or even in crowd along with many other objects with various shapes and sizes at various positions with uncontrolled lighting and background). An example of face detection is shown in Figure 1.3.

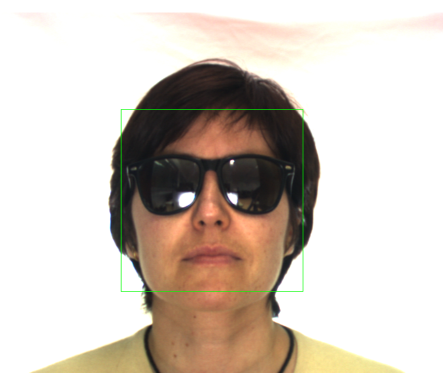

Figure 1.3: An example of face detection

1.3.2 Face Alignment and Normalisation

Face alignment and normalisation module rectifies each block containing a single face image in terms of geometric parameters and intensity distribution. As the name suggests, it employs two transformations :

1. **Geometric transformation :** It normalises orientation, translation and scale such that each face appear vertically upright, centrally located and of same size.

2. **Photometric transformation :** It normalises intensity distribution and skin colour so that face images of a single person appear similar to each other.

1.3.3 Feature Extraction

This module extracts relevant visual features (e.g., texture, shape) that would make the recognition system reliable and robust in terms of recognition and/or verification of query image against the face images available in the database.

1.3.4 Similarity Measure and Matching

Conceptually this module compares the feature vector extracted from the query image with that of individual face image present in the face image database. Similarity between two feature vectors may be measured as the reciprocal of some distance measure. various types of distance measures are being employed. Euclidean distance is one of the simplest ones. It may be noted that not all the features extracted from the image are equally important nor they have equal variability. A better similarity measure should assign appropriate weight to individual element of the feature vector. Mahalanobis distance is found to be very useful in most of the cases.

The query image is finally, recognised as the database image whose feature vector is closest to that of the query image and the distance between them is less than acceptable error. Otherwise, it is declared that no match is found.

1.4 Objective of the Work

Face Recognition is an important component of any security system used all around the globe. But in reality several issues, some of them mentioned already, arise a great problem for a face recognition system. We have performed experiments with different kinds of global (PCA, LDA) and local features (GLCM, LBP, LDP) for face description and used different classifiers (KNN, SVM). The training images contain variation of pose, illumination, and occlusion. **We have proposed a fusion of local feature descriptors, which outperforms the state-of-the-art global as well as local feature based face-recognition techniques and is robust under varying lighting conditions.**

1.5 Organization of the Report

Organization of rest of the report is as follows. We first discuss the different global and local feature extraction techniques for face recognition, we have used and then we give a brief overview of the classifiers, we have used in the second chapter. our proposed feature extraction technique and experimental results are given, in the third chapter along with conclusion of our work.

Chapter 2

Face Recognition Techniques

In this chapter we want to describe some of the state-of-the-art global and local feature extractors, we have used in our experiment. Also we will talk about some of the well-known classifiers, we have used.

2.1 Global Feature Extractor

Let X be a d-dimensional feature vector. In our case, d is equal to the number of pixel of each face. The high dimensionality of the related "image space" is a well-known problem for the design of a good recognition or verification algorithm. Therefore, methods for reducing the dimensionality of such image space are required. To this end, Principal Component Analysis (PCA) which seeks a projection that best represents the data in a least-squares sense, and Linear Discriminant Analysis (LDA) which seeks a projection that best separates the data in a least- squares sense, are widely used. We briefly describe these methods in the following section.

2.1.1 Principal Component Analysis

Principal Component Analysis[10] is defined by the transformation:

$$y_i = W^t x_i \tag{2.1}$$

where $x_i \in X \subseteq \Re^d$, $i = 1, \ldots, n$ (n samples). W is a d- dimensional transformation matrix whose columns are the eigenvectors related to the eigenvalues computed according to the formula:

$$\lambda e_i = S e_i \tag{2.2}$$

where S is the scatter matrix (i.e., the covariance matrix) and is given by:

$$S = \sum_{i=1}^{n} (x_i - m)(x_i - m)^t, \tag{2.3}$$
$$m = \frac{1}{n} \sum_{i=1}^{n} x_i$$

This transformation is called Karuhnen-Loeve transform. It defines the d - dimensional space in which the covariance among the components is zero. In this way, it is possible to consider a small number of "principal" components exhibiting the highest variance (the most expressive features). In the face space, the eigenvectors related to the most expressive features are called **"eigenfaces"**.

EIGENFACE ALGORITHM

Let a face image $I(x, y)$ be a two-dimensional N-by-N array of intensity values or a vector of dimension N^2. A typical image of size 256-by-256 describes a vector of dimension $65,536$, or equivalently, a point in $65,536$-dimensional space[5]. Consider our training set of images of 112-by-92 pixels. The main idea of principal component analysis is to find the vectors which best account for the distribution of the face images within the entire image space. Steps for Feature Extraction:

1. The first step is to obtain a 1-D vector from each of the 2-D facial images, in the training set by concatenating each row (or column) into a long thin vector. Let's suppose we have M vectors of size N (= rows of image $*$ columns of image) representing a set of sampled images.

p_j's represent the pixel values.

$$\Gamma_i = [p_1 \dots p_N]^T, i = 1 \dots N \tag{2.4}$$

The training set can be represented as :

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

2. The second step is to obtain the mean image and the training images are mean centered by subtracting the mean image from each image vector, Γ_i . Let m represent the mean image.

$$m = \frac{1}{M}\sum_{i=1}^{M}\Gamma_i \tag{2.5}$$

And let w_i be defined as mean centered image

$$w_i = \Gamma_i - m \tag{2.6}$$

3. Our goal is to find a set of M orthonormal vectors, e_i's which have the largest possible projection onto each of the w_i's. The k^{th} vector, e_k is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(e_k^T w_n)^2 \tag{2.7}$$

is maximised with the orthonormality constraint

$$e_l^T e_k = \delta_{lk} = \begin{cases} 1 & \text{if } l = k \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

It has been shown that the e_i's and λ_i's are given by the eigenvectors and eigenvalues of the covariance matrix, C.

4. The covariance matrix, C has been obtained in the following manner:

$$\begin{aligned} C &= \frac{1}{M}\sum_{n=1}^{M} w_n w_n^T \\ &= WW^T \end{aligned} \tag{2.9}$$

where W is a matrix composed of the column vectors w_i's placed side by side, i.e., $W = \{w_1, w_2, w_3, \ldots, w_M\}$.

5. To find eigenvectors from the covariance matrix, C is a huge computational task. The size of C is $N \times N$ which could be enormous. For example, images of size 64×64 create the covariance matrix of size 4096×4096. It is not practical to solve for the eigenvectors of C directly. A common theorem in linear algebra states that the vectors e_i and scalars λ_i can be obtained by solving for the eigenvectors and eigenvalues of the $M \times M$ matrix W^TW. Let d_i and μ_i be the eigenvectors and eigenvalues of W^TW respectively. Then

$$W^TWd_i = \mu_i d_i \tag{2.10}$$

By multiplying left to both sides by W

$$WW^T(Wd_i) = \mu_i(Wd_i) \tag{2.11}$$

which means that the first $(M-1)$ eigenvectors e_i and eigenvalues λ_i of WW^T are given by Wd_i and μ_i, respectively. Wd_i needs to be normalized in order to be equal to e_i. Since we only sum up a finite number of image vectors, the rank of the covariance matrix cannot exceed $M - 1$(The -1 come from the subtraction of the mean vector m).

6. The eigenvectors corresponding to nonzero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace within which most image data can be represented with a small amount of error. The eigenvectors are sorted from high to low according to their

corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is one that reflects the greatest variance in the image. That is, the smallest eigenvalue is associated with the eigenvector that finds the least variance. They decrease in exponential fashion, meaning that the roughly 90% of the total variance is contained in the first 5% to 10% of the dimensions.

A facial image can be projected onto $M'(\lll M)$ dimensions by computing

$$\Omega = [v_1 v_2 v_3 \ldots v_{M'}]^T \tag{2.12}$$

where $v_i = {e_i}^T w_i$. v_i is the i^{th} coordinate of the facial image in the new space, which came to be the principal component. The vectors e_i are also images, so called, "eigenimages", or "eigenfaces" in our case. They can be viewed as images and indeed look like faces. So, Ω describes the contribution of each "eigenface" in representing the facial image by treating the "eigenfaces" as a basis set for facial images.

Figure 2.1: An example of "eigenfaces"

Figure 2.2: A face image, represented as a linear combination of "eigenfaces"

2.1.2 Linear Discriminant Analysis

For the $c(c \geqslant 2)$-class problem, the natural generalization of Fisher's linear discriminant involves $(c-1)$ discriminant functions. Thus, the projection is from a d- dimensional space to a $(c-1)$-dimensional space, and it is tacitly assumed that $d \geqslant c$. The steps for LDA for multiclass problem are as follows[10]:

1. The generalization for the **within-class scatter matrix** is obvious:

$$\mathbf{S_W} = \sum_{i=1}^{c} \mathbf{S_i} \tag{2.13}$$

   where, as in 2-class problem, x's are the feature vectors, extracted from the training images and

$$\mathbf{S_i} = \sum_{\mathbf{x} \in \mathfrak{D}_i} (\mathbf{x} - \mathbf{m_i})(\mathbf{x} - \mathbf{m_i})^T \tag{2.14}$$

   and

$$\mathbf{m_i} = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathfrak{D}_i} \mathbf{x} \tag{2.15}$$

   where n_i = number of samples in the i^{th} class.

2. The proper generalization for **between-class scatter matrix $\mathbf{S_B}$** is not quite so obvious. Suppose that we define a total mean vector $\mathbf{m}$ and a total scatter matrix $\mathbf{S_T}$ by

$$n = \sum_{i=1}^{c} n_i \tag{2.16}$$

$$\mathbf{m} = \frac{1}{n} \sum_{\mathbf{x} \in \mathfrak{D}} \mathbf{x}$$

$$= \sum_{i=1}^{c} n_i \mathbf{m_i} \tag{2.17}$$

and

$$\mathbf{S_T} = \sum_{\mathbf{x}\in\mathfrak{D}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T \tag{2.18}$$

Then it follows that

$$\begin{aligned}
\mathbf{S_T} &= \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathfrak{D}} (\mathbf{x} - \mathbf{m_i} + \mathbf{m_i} - \mathbf{m})(\mathbf{x} - \mathbf{m_i} + \mathbf{m_i} - \mathbf{m})^T \\
&= \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathfrak{D}_i} (\mathbf{x} - \mathbf{m_i})(\mathbf{x} - \mathbf{m_i})^T + \sum_{i=1}^{c} \sum_{\mathbf{x}\in\mathfrak{D}_i} (\mathbf{m_i} - \mathbf{m})(\mathbf{m_i} - \mathbf{m})^T \\
&= \mathbf{S_W} + \sum_{i=1}^{c} n_i (\mathbf{m_i} - \mathbf{m})(\mathbf{m_i} - \mathbf{m})^T
\end{aligned} \tag{2.19}$$

It is natural to define this second term as a general **between-class scatter matrix**, so that the **total scatter** is the sum of the **within-class scatter** and the **between-class scatter**:

$$\mathbf{S_B} = \sum_{i=1}^{c} n_i (\mathbf{m_i} - \mathbf{m})(\mathbf{m_i} - \mathbf{m})^T \tag{2.20}$$

and

$$\mathbf{S_T} = \mathbf{S_W} + \mathbf{S_B} \tag{2.21}$$

3. The projection from a d-dimensional space to a $(c-1)$-dimensional space is accomplished by $(c-1)$ discriminant functions:

$$\begin{aligned} y_i &= \mathbf{w_i}^{\mathbf{T}} \mathbf{x} \\ i &= 1, \ldots, (c-1) \end{aligned} \tag{2.22}$$

If the y_i's are viewed as components of a vector $\mathbf{y}$ and the weight vectors $\mathbf{w_i}$'s are viewed as the columns of a d-by-$(c-1)$ matrix $\mathbf{W}$, then the projection can be written as a single matrix equation :

$$y = \mathbf{W^T x} \tag{2.23}$$

The samples $x_1, \ldots, x_n$ project to a corresponding set of samples $\mathbf{y_1}, \ldots, \mathbf{y_n}$, which can be described by their own mean vectors and scatter matrices. Thus, if we define

$$\tilde{\mathbf{m}}_\mathbf{i} = \frac{1}{n_i} \sum_{\mathbf{y} \in \mathrm{y_i}} \mathbf{y} \tag{2.24}$$

$$\tilde{\mathbf{m}} = \frac{1}{n} \sum_{i=1}^{c} n_i \tilde{\mathbf{m}}_\mathbf{i} \tag{2.25}$$

$$\tilde{\mathbf{S}}_\mathbf{W} = \sum_{i=1}^{c} \sum_{\mathbf{y} \in \mathrm{y_i}} (\mathbf{y} - \tilde{\mathbf{m}}_\mathbf{i})(\mathbf{y} - \tilde{\mathbf{m}}_\mathbf{i})^T \tag{2.26}$$

and

$$\tilde{\mathbf{S}}_\mathbf{B} = \sum_{i=1}^{c} n_i (\tilde{\mathbf{m}}_\mathbf{i} - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_\mathbf{i} - \tilde{\mathbf{m}})^\mathbf{T} \tag{2.27}$$

It is a straight-forward matter to show that

$$\tilde{\mathbf{S}}_\mathbf{W} = \mathbf{W^T S_W W} \tag{2.28}$$

$$\tilde{\mathbf{S}}_\mathbf{B} = \mathbf{W^T S_B W} \tag{2.29}$$

4. These equations show how the within-class and between-class scatter matrices are transformed by the projection to the lower dimensional space (Fig. 2.3). What we seek is a transformation matrix $\mathbf{W}$ that in some sense maximizes the ratio of the between-class scatter to the

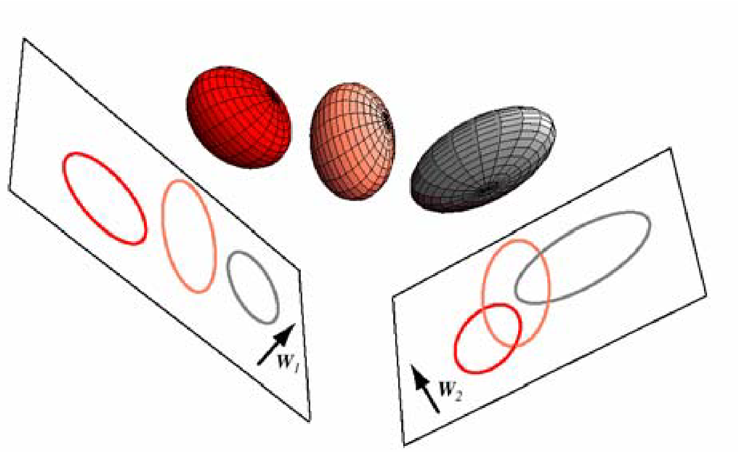

Figure 2.3: Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $\mathbf{W_1}$ and $\mathbf{W_2}$. Informally, multiple discriminant methods seek the optimum such subspace, that is, the one with the greatest separation of the projected distributions for a given total within- scatter matrix, here as associated with $\mathbf{W_1}$.

within-class scatter. So, we obtain the criterion function

$$J(\mathbf{W}) = \frac{|\tilde{\mathbf{S}}_\mathbf{B}|}{|\tilde{\mathbf{S}}_\mathbf{W}|} = \frac{|\mathbf{W^T S_B W}|}{|\mathbf{W^T S_W W}|} \tag{2.30}$$

The columns of an optimal $\mathbf{W}$ are the generalized eigenvectors that correspond to the largest eigenvalues in

$$\mathbf{S_B W_i} = \lambda_\mathbf{i} \mathbf{S_W W_i} \tag{2.31}$$

A few observations about this solution are in order.

- If $\mathbf{S_W^{-1}}$ is non-singular, we can obtain a conventional eigenvalue problem by writing:

$$(\mathbf{S_W^{-1} S_B})\mathbf{W_i} = \lambda_\mathbf{i}(\mathbf{S_W^{-1} S_W})\mathbf{W_i} \tag{2.32}$$

$$\begin{aligned}(\mathbf{S_W^{-1} S_B})\mathbf{W_i} &= \lambda_\mathbf{i}(\mathbf{I})\mathbf{W_i} \\ &= \lambda_\mathbf{i}\mathbf{W_i}\end{aligned} \tag{2.33}$$

- In practice, $\mathbf{S_W}$ is often singular since the data are image vectors with large dimensionality while the size of the data set is much smaller $(M << N)$
  To alleviate this problem, we can perform two projections:
  (a) PCA is first applied to the data set to reduce its dimensionality.
  (b) LDA is then applied to further reduce the dimensionality.

These global feature descriptors do not work well under variation of illumination. The experimental results would be given in the next chapter. Instead of the original image, we take **Intensity gradients of each image**, which is supposed to change a little under illumination variation, for Principal Component Analysis. A little bit of improvement in success rate is found and reported in the next chapter.

2.2 Local Feature Extractor

2.2.1 Gray-level Co-occurrence Matrix

One of the simplest approaches for describing texture is to use statistical moments of the intensity histogram of an image or region[11]. Using only histograms in calculation, will result in measures of texture that only carry information about distribution of intensities, but not about the relative position of pixels with respect to each other in that texture. Using a statistical approach such as **co-occurrence matrix** will help to provide valuable information about the relative position of the neighbouring pixels in an image. Given an image I, of size $N \times N$, the co-occurrence, matrix P can be defined as:

$$P(i,j) = \sum_{x=1}^{N}\sum_{y=1}^{N} \begin{cases} 1 & \text{if } I(x,y) = i \text{ and } I(x+\Delta_x, y+\Delta_y) = j \\ 0 & \text{Otherwise} \end{cases} \tag{2.34}$$

where the offset (Δ_x, Δ_y), is specifying the distance between the pixel-of-interest and its neighbour. Note that the offset (Δ_x, Δ_y) parametrization makes the co- occurrence matrix sensitive to rotation. Choosing the offset vector, so a rotation of the image not equal to 180 degrees, will result in a different co-occurrence matrix for the same (rotated) image. This can be avoided by forming the co-occurrence matrix using a set of offsets sweeping through 180 degrees at the same distance parameter Δ to achieve a degree of rotational invariance (i.e. $[0\ \Delta]$ for $0°$: P horizontal, $[-\Delta\ \Delta]$ for $45°$: P right diagonal, $[-\Delta\ 0]$ for $90°$: P vertical, and $[-\Delta\ \Delta]$ for $135°$: P left diagonal). The averaged GLCM matrix can be calculated by:

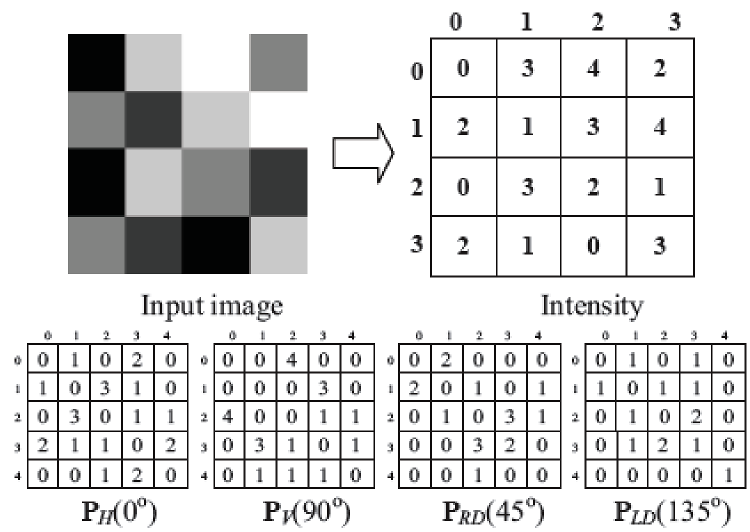

Figure 2.4: Co-occurrence matrix generation for $N_g = 5$ levels and four different offsets: $P_H(0°)$, $P_V(90°)$, $P_{RD}(45°)$, and $P_{LD}(135°)$.

$$P = \frac{P_H + P_V + P_{RD} + P_{LD}}{4} \tag{2.35}$$

For normalization, we then divide each entry in the GLCM matrix P by normalization constant R, that represents the number of neighboring pixel pairs used in calculating GLCM, to form a new normalized matrix p which will be used later on in the classification process.

HARALICK FEATURES

In 1973, Haralick[6] introduced 14 statistical features. These features are generated by calculating the features for each one of the co-occurrence matrices obtained by using the directions 0°, 45°, 90°, and 135°, then averaging these four values. We have chosen Δ to be 1, 2, 3 and 4. A vector of these 7 statistical features is used for characterizing the co-occurrence matrix contents. Though we have used only first 4 features in the list given below.
Notations:
$p(i,j)$ is the $(i,j)^{th}$ entry in normalized GLCM.
Ng, dimension of GLCM (number of gray levels)
$p_x(i)$ and $p_y(j)$ are the marginal probabilities:

$$p_x(i) = \sum_{j=1}^{N_g} p(i,j) \tag{2.36}$$

$$p_y(j) = \sum_{i=1}^{N_g} p(i,j) \tag{2.37}$$

μ is the mean of μ_x and μ_y;

$$\mu_x = \sum_{i=1}^{N_g} i p_x(i) \tag{2.38}$$

$$\mu_y = \sum_{i=1}^{N_g} i p_y(i) \tag{2.39}$$

1. Energy:

$$f_1 = \sum_i \sum_j p(i,j)^2 \tag{2.40}$$

2. Contrast:

$$f_2 = \sum_{n=0}^{N_g-1} n^2 \{ \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j) : |i-j| = n \} \tag{2.41}$$

3. Correlation:

$$f_3 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{(i-\mu_x)(j-\mu_y)p(i,j)}{\sigma_x \sigma_y} \tag{2.42}$$

4. Homogeneity:

$$f_4 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} \frac{1}{1+(i-j)^2} p(i,j) \tag{2.43}$$

5. Autocorrelation:

$$f_5 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i*j)p(i,j) \tag{2.44}$$

6. Dissimilarity:

$$f_6 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} |i-j| p(i,j) \tag{2.45}$$

7. Inertia:

$$f_7 = \sum_{i=1}^{N_g} \sum_{j=1}^{N_g} (i-j)^2 p(i,j) \tag{2.46}$$

2.2.2 Local Binary Pattern

Ojala et al.[12] introduced the Local Binary Pattern operator in 1996 as a means of summarizing local gray-level structure. The operator takes a local neighborhood around each pixel, thresholds the pixels of the neighborhood at the value of the central pixel and uses the resulting binary-valued image patch as a local image descriptor. It was originally defined for 3×3 neighborhoods, giving 8-bit codes based on the 8 pixels around the central one. Formally, the LBP operator takes the form

$$LBP(x_c, y_c) = \sum_{n=0}^{7} 2^n s(i_n - i_c) \tag{2.47}$$

where in this case n runs over the 8 neighbors of the central pixel c, i_c and i_n are the gray-level values at c and n, and $s(u)$ is 1 if $u \geq 0$ and 0 otherwise. The LBP encoding process is illustrated in fig. 2.5.

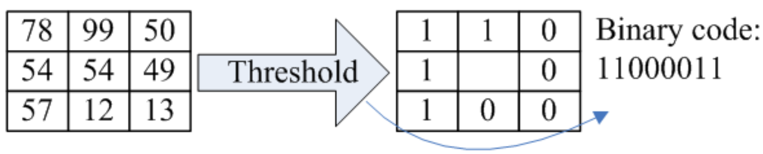

Figure 2.5: Illustration of the basic LBP operator.

2.2.3 Local Directional Pattern

The **Local Directional Pattern (LDP)** is an eight bit binary code assigned to each pixel of an input image[9]. This pattern is calculated by comparing the relative edge response value of a pixel in different directions. For this purpose, we calculate eight directional edge response value of a particular pixel using Kirsch masks in eight different orientations ($\mathbf{M_0} \sim \mathbf{M_7}$) centered on its own position. These masks are shown in the fig. 2.4.
Applying eight masks, we obtain eight edge response value $m_0, m_1, \ldots, m_7$, each representing the edge significance in its respective direction. The response values are not equally important in all directions. The presence of corner or edge show high response values in particular directions. We are interested to know the k most prominent directions in order to generate the **LDP**. Hence, we find the top k values $|m_j|$ and set them to 1. The other $(8-k)$ bit of 8-bit **LDP** pattern is set to 0.

$$\begin{array}{cccc}
\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} & \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} & \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \\
\text{East } (M_0) & \text{North East } (M_1) & \text{North } (M_2) & \text{North West } (M_3) \\
\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} & \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix} \\
\text{West } (M_4) & \text{South West } (M_5) & \text{South } (M_6) & \text{South East } (M_7)
\end{array}$$

Figure 2.6: Kirsch edge response masks in eight directions

85	32	26
53	50	10
60	38	45

Table 2.1: The Gray values

The LDP include similar information than LBP but produces more stable pattern in presence of noise and non monotonic illumination changes since gradients are more stable than gray level[9].

Histogram of LDP

After encoding an image with the LDP operator we get an encoded image I_L. We use $k = 3$ which generates 56 distinct values in our encoded image. So histogram H of this LDP labeled image $I_L(x, y)$ is a 56-bin histogram and can be defined as,

$$H_i = \sum_{x,y} P(I_L(x, y) = C_i), \tag{2.48}$$

where $C_i = i^{th}$ LDP Pattern $(0 < i \leq 56)$ and

$$P(A) = \begin{cases} 1 & \text{if } A = TRUE \\ 0 & \text{if } A = FALSE \end{cases}$$

Mask Index	m_7	m_6	m_5	m_4	m_3	m_2	m_1	m_0
Mask Value	161	97	161	537	313	97	-503	-393
Rank	6	7	5	1	4	8	2	3
Code Bit	0	0	0	1	0	0	1	1
LDP Code	19							

Table 2.2: LDP code generation

Face Representation using LDP

Each face is represented by a LDP histogram. LDP histogram contains fine detail information of an image, such as, edges, spot, corner and other local texture features. But histogram computed over the whole face image encodes only the occurrences of the micro-patterns without any knowledge about their locations. In order to incorporate some degree of location information, we divide face images into small regions $R_0; R_1; \ldots; R_n$ and extracted the LDP histograms H_{R_i} from each region R_i. These n LDP histograms are concatenated to get a spatially combined LDP histogram which plays the role of a global face feature for the given face image. The process can be visualized with fig. 2.5.

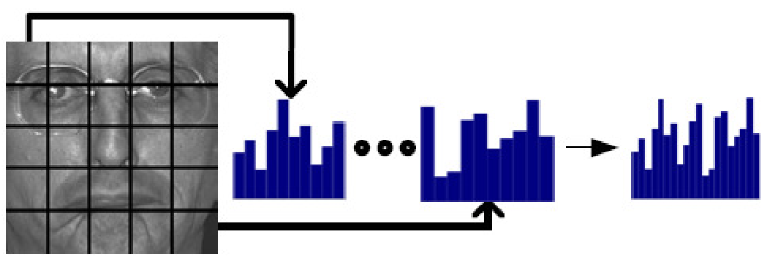

Figure 2.7: Facial Image Representation using Spatially Enhanced Histogram

2.3 Illumination Normalization

This section describes image preprocessing method, used prior finding the LBP histogram. It incorporates a series of steps,chosen to counter the effects of illumination variations, local shadowing and highlights, while still preserving the essential elements of visual appearance for use in recognition. Although not done by design, the final chain is reminiscent of certain preprocessing stages found in the mammalian visual cortex. In detail, the steps are as follows.

Gamma Correction :

This is a nonlinear gray-level transformation that replaces graylevel I with I^γ (for $\gamma > 0$) or $\log(I)$ (for $\gamma = 0$), where $\gamma \in [0;1]$ is a user-defined parameter. It has the effect of enhancing the local dynamic range of the image in dark or shadowed regions, while compressing it in bright regions and at highlights. The basic principle is that the intensity of the light reflected from an object is the product of the incoming illumination L (which is piecewise smooth for the most part) and the local surface reflectance R (which carries detailed object-level appearance information). We want to recover object-level information independent of illumination. Here we use $\gamma = 0.2$ as the default setting.

Difference of Gaussian (DoG) Filtering :

Gamma correction does not remove the influence of overall intensity gradients such as shading effects. High-pass filtering removes both the useful and the incidental information. Difference of Gaussians is a grayscale image enhancement algorithm that involves the subtraction of one blurred version of an original grayscale image from another, less blurred version of the original one. Difference of Gaussians can be utilized to increase the visibility of edges and other detail present in a digital image. We use $\sigma_0 = 1.0$ and $\sigma_1 = 2.0$ by default.

Masking :

If facial regions (hair style, beard etc) that are felt to be irrelevant or too variable need to be masked out, the mask should be applied at this point.

Contrast Equalization :

This stage rescales the image intensities to standardize a robust measure of overall contrast or intensity variation

$$I(x,y) \leftarrow \frac{I(x,y)}{(mean(|I(x\prime,y\prime)|^{a}))^{\frac{1}{a}}} \tag{2.49}$$

$$I(x,y) \leftarrow \frac{I(x,y)}{(mean(\min\ (\tau,|I(x\prime,y\prime)|)^{a}))^{\frac{1}{a}}} \tag{2.50}$$

By default we use $a = 0.1$ and $\tau = 10$.

2.4 Classifier

2.4.1 k-Nearest Neighbours Classifier

A classification problem for objects in a particular domain is the problem of separating these objects into smaller classes and giving a criteria whether a particular object in the domain is in a particular class or not. In pattern recognition, the k-nearest neighbours algorithm is a method for classifying point based on closest training samples in the feature space. An object is classified in class i if among its k nearest neighbours maximum no. of object belongs to class i. Here the nearest neighbour corresponds to the minimum Euclidean distance.

Let $(x_i;\theta_i)$; $i = 1;2;\ldots;m$ be given where $x_i \in \mathbb{R}_n$ and θ_i denote the class of x_i. Let there be C classes and $x \in \mathbb{R}_n$ be the point to be classified.

Let k be a positive integer.

Steps of k-nearest neighbours algorithm :

Step 1 Find k nearest neighbours of x among $(x_1;x_2;\ldots;x_m)$

Step 2 Let k_i of the nearest neighbours belongs to class i, $i = 1;2;\ldots C$.

Step 3 Classify x to class i if $k_i > k_j$, $\forall j \neq i$ and $\sum_{i=1}^{C} k_i = k$

2.4.2 Support Vector Machine

The basic thought of support vector machines (SVM) is to find a superior classified surface satisfying the classified request. It is making the distance from sample point to the classified surface as far as possible, in another word, to maximise the class interval[14].

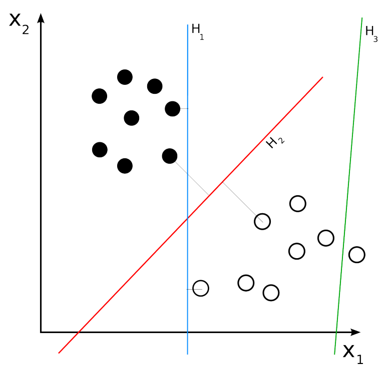

Figure 2.8: H3 (green) doesn't separate the two classes. H1 (blue) does, with a small margin and H2 (red) with the maximum margin.

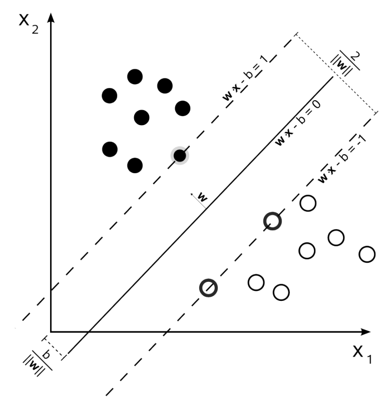

Figure 2.9: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes.

Supposes the training data are $(x_1, y_1), (x_2, y_2), \ldots, (x_L, y_L)$, $x \in \mathbb{R}^n$, $y \in \{1, +1\}$ .The general form of linear substitution function in d-dimensional space is:

$$g(x) = wx + b \tag{2.51}$$

All samples in the data set can be classified correctly by the classification surface $wx + b = 0$. This classified surface is the most superior hyper plane. The nearest vector to most superior hyper plane in different classes is called **support vector (SV)**.

The support vector distance is $\frac{1}{||w||}$, the classified surface distance is called the geometry boundary. The geometry boundary is bigger and the classified effect is better, with the wrong classification possibility smaller. Therefore, the question to seek the most superior classified surface is transformed to solve quadratic programming problems as following:

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(wx_i + b) \geq 1 \ \forall i = 1, \dots, L \end{aligned} \tag{2.52}$$

Regarding the situation non-linear, nuclear kernel function $K(x_i, x)$ may be introduced to find the linear classification surface in high dimensional space by the formula following:

$$f(x, \alpha^\star, b^\star) = \text{sgn}(\sum_{i=1}^{L} y_i {\alpha_i}^\star K(x_i, x) + b^\star) \tag{2.53}$$

The kernel functions used in general are linear, polynomial and radial basis functions (RBFs) defined as:

- Linear Kernel:

$$K(x_i, x_j) = x_i.x_j \tag{2.54}$$

- Polynomial Kernel:

$$K(x_i, x_j) = (x_i.x_j + 1)^d \tag{2.55}$$

- Radial Basis Kernel:

$$K(x_i, x_j) = \exp\left(-\gamma ||x_i - x_j||^2\right) \tag{2.56}$$

where x_i and x_j denote two samples. The user-controlled parameters are the degree d in the case of the polynomial and the γ value in the case of the RBF kernel.

Chapter 3

Our Proposed Face Identification & Verification Technique

In this chapter, we describe our face identification and verification algorithm based on fusion of various local features, described in the previous chapter. Comparison with the individual local features and the global feature based techniques show that identification as well as verification rate improves a lot in our case. We perform experiments on three well-known face datasets, ORL and Extended Yale-B and AR. ORL dataset has images of 40 persons, with 10 different poses whereas Extended Yale-B dataset has the frontal face images of 38 persons under 64 different illuminating conditions and AR database has pose, illumination, expression variation and partial occlusion. Our proposed algorithms exhibit high robustness under pose illumination, expression variation and partial occlusion.

Figure 3.1: Five sample images of one subject in the ORL face database.

Figure 3.2: Five sample images of one subject in the Extended Yale B face database.

Figure 3.3: Five sample images of one subject in the AR face database.

3.1 Our face identification algorithm

The face identification system, we have implemented, consists of the following major steps in order of execution.

3.1.1 Illumination Normalisation

If the images are captured under different illumination condition, illumination normalisation (including a. Gamma Correction, b. Difference of Gaussian (DoG) filtering, c. Masking (optional), d. Contrast Equalization) as mentioned in section 2.3 is essential. In case of uniform and consistent illumination condition, we do not need to go through these steps.

3.1.2 Feature Extraction

A. GLCM

To keep some local information, present in the image, we divide the whole image into 4-by-4 partition and find out the normalised gray-level co-occurrence matrix, p for Δ values : 1, 2, and 3, as mentioned in section 2.2.1 from each of the 16 blocks. From each of the GLCM p's, derived from each block, we calculate the first 4 Haralick features, namely a. Energy, b. Contrast, c.

Figure 3.4: Examples of images of one person from the Extended Yale-B frontal database.

Figure 3.5: The corresponding illumination normalized images from the preprocessing chain.

Correlation and d. Homogeneity. Hence we get a $3 \times 4 = 12$-dimensional feature vector from each block and concatenate each of the feature vectors to get a $12 \times 16 = 192$ dimensional feature vector, FV_{GLCM}.

B. LBP

We find out the LBP features from each image for 3 windows, e.g. 3-by-3, 5- by-5 and 7-by-7 as mentioned in section 2.2.2. For 3-by-3 window, we get a single 8-bit LBP value (ranging from 0 to 255) for each pixel. Hence we get a histogram of LBP values for the whole image.

For the 5-by-5 window, we have 16 cells in the periphery. Hence we get two 8-bit LBP values corresponding to the first and last 8 cells in the periphery (starting from the top left corner cell) and we get two histograms of LBP values for the whole image.

For the 7-by-7 window, we have 24 cells in the periphery. Hence we get three 8-bit LBP values corresponding to the first, middle and last 8 cells in the periphery (starting from the top left corner cell). Hence we get three histograms of LBP values for the whole image.

In total, we have $(1 + 2 + 3) = 6$ histograms. We take out the bin centered in 0 which correspond to pixels surrounded by pixels with the same value (pixels inside the strokes or inside the background) and concatenate each of the histograms and get a single feature vector FV_{LBP} of dimension $6 \times 255 = 1530$.

It is impractical to work with such a large dimensional feature vector. Hence we reduce its dimensionality, working out the discrete cosine transform (DCT) of FV_{LBP} and select as feature vector ${FV_{LBP}}^{DCT}$ the first 255 components.

C. LDP

We divide the whole image into 5-by-5 blocks and find out the Local Directional Pattern feature from each of the blocks as mentioned in section 2.2.3. Previously to calculate the histogram of each block, we take into account that the LDP is obtained coding the three predominant directions (there are just three ones of the 8 bits code), so there are just 56 possible values $\{7, 11, 13, 14, \ldots, 200, 208, 224\}$. Therefore, we calculate a 56 bin histogram of each block.
The histograms of each block are concatenated obtaining a $56 \times 25 = 1400$ dimensional feature vector called FV_{LDP}.
Again it becomes impractical to work with such a large dimensional feature vector. Hence we reduce its dimensionality, working out the discrete cosine transform (DCT) of FV_{LDP} and select as feature vector ${FV_{LDP}}^{DCT}$ the first $3 \times 56 = 168$ components.

D. Feature Vector

We concatenate FV_{GLCM}, ${FV_{LBP}}^{DCT}$ and ${FV_{LDP}}^{DCT}$ to get a feature vector FV of dimension $(192 + 255 + 168) = 615$.

3.1.3 Classification

We use Nearest Neighbour classifier in this case. Since first 192 components of the feature vector are of different variability than the next 255 components and the last 168 components, we use Mahalanobis distance as dissimilarity measure between two feature vectors.

3.2 Our face verification algorithm

Face verification is a one-to-one match that compares a query face image against a template face image whose identity is being claimed. In this case the problem can be degenerated to a two-class problem : face image pair belong to same person or not. The face verification system we have implemented has the following major steps.

1. In this case, suppose we want to train the system with images of M classes, each having N images. We calculate feature vectors, from each of the $M \times N$ images following steps mentioned in section 3.1.1 and 3.1.2.

2. Corresponding to a single class (say f^{th} class), we choose any two feature vectors $FV^{f}{}_{i}$, $FV^{f}{}_{j}$ $(i \neq j)$, in $\binom{N}{2}$ ways and find out the difference vector, $DV^{Sim}{}_{i,j} = |FV^{f}{}_{i} - FV^{f}{}_{j}|$.

3. We form a set $S_{Sim} = \{DV^{Sim}{}_{i,j}\}$.

4. We choose any two classes (say f^{th} and g^{th}class) and we choose any two feature vectors $FV^{f}{}_{i}$, $FV^{g}{}_{j}$, in $\binom{M \times N}{2} - \binom{N}{2}$ ways and find out the difference vector, $DV^{Dissim}{}_{i,j} = |FV^{f}{}_{i} - FV^{g}{}_{j}|$.

5. We form a set $S_{Dissim} = \{DV^{Dissim}{}_{i,j}\}$.

6. We train a binary SVM classifier with RBF kernel, with same number of samples collected from S_{Sim} and S_{Dissim}.

7. We form a test set in the same way and predict the class labels of the difference vectors in the test set using the model, we already generated.

Chapter 4

Experimental Results

4.1 Identification Results

4.1.1 ORL Database

ORL database has 400 images (40 persons, with 10 different poses). For each person we randomly choose 7 images for the training set and the remaining 3 images are kept in the test set. We rebuild the training set and the test set 30 times and perform the experiment.

4.1.2 Extended Yale B Database

Extended Yale B database has 2432 images (38 persons under 64 different illumination condition). We perform the experiments on a subset of this dataset containing 32 different images for each person. From this subset, for each person we randomly choose 27 images for the training set and the remaining 5 images are kept in the test set. We rebuild the training set and the test set 30 times and perform the experiment.

4.1.3 AR Database

We work on a subset of AR database having 1703 images (131 persons, each of whom has 13 different images). From this subset, for each person we

Table 4.1: AVERAGE IDENTIFICATION RATE ON ORL DATABASE

Features	Average Identification Rate (%)
GLCM+LBP+LDP (w/o DCT)	**98.89**
GLCM+LBP+LDP (with DCT)	**97.67**
(GLCM+LBP)	93.27
(LBP+LDP)	95.16
(GLCM+LDP)	95.8
GLCM	88.67
LBP (w/o DCT)	91.58
LDP (w/o DCT)	94.25
LBP (with DCT)	89.50
LDP (with DCT)	93.3
PCA (Gradient)	93.86
LDA	93.46
PCA	92.5

Table 4.2: AVERAGE IDENTIFICATION RATE ON EXTENDED YALE-B DATABASE

Features	Average Identification Rate (%)
GLCM+LBP+LDP (w/o DCT)	**95.39**
GLCM+LBP+LDP (with DCT)	**94.5**
(GLCM+LBP)	89
(LBP+LDP)	90
(GLCM+LDP)	92
GLCM	86
LBP (w/o DCT)	87.50
LDP (w/o DCT)	92
LBP (with DCT)	86.50
LDP (with DCT)	91.25
PCA (Gradient)	84.36
LDA	84
PCA	82.5

randomly choose 9 images for the training set and the remaining 4 images are kept in the test set. We rebuild the training set and the test set 30 times and perform the experiment.

Table 4.3: AVERAGE IDENTIFICATION RATE ON AR DATABASE

Features	Average Identification Rate (%)
GLCM+LBP+LDP (w/o DCT)	**95**
GLCM+LBP+LDP (with DCT)	**93.86**
(GLCM+LBP)	90.67
(LBP+LDP)	93.13
(GLCM+LDP)	93
GLCM	86
LBP (w/o DCT)	89.73
LDP (w/o DCT)	92.78
LBP (with DCT)	88.45
LDP (with DCT)	91.25
PCA (Gradient)	87.20
LDA	86.26
PCA	84

4.2 Verification Results

We have used the LIB-SVM[14] tool, in this face verification problem.

4.2.1 ORL Database

We first choose images of 20 classes of people. Each class has 10 images. Hence, $|S_{Sim}| = 20 \times \binom{10}{2} = 900$ and $|S_{Dissim}| = \binom{200}{2} - |S_{Sim}| = 19,900 - 900 = 19,000$. We randomly choose 900 difference vectors from the set S_{Dissim} 30 times and perform the experiment.

4.2.2 Extended Yale B Database

We first choose images of 20 classes of people. From each class, we choose 32 images. Hence, $|S_{Sim}| = 20 \times \binom{32}{2} = 9,920$ and $|S_{Dissim}| = \binom{20 \times 32}{2} - |S_{Sim}| = 204,480 - 9,920 = 194,560$. We randomly choose 9,920 difference vectors from the set S_{Dissim} 30 times and perform the experiment.

Table 4.4: AVERAGE VERIFICATION RATE ON ORL DATABASE

Features	Average Verification Rate (%)
GLCM+LBP+LDP (with DCT)	**99**
(GLCM+LBP)	95
(LBP+LDP)	96.33
(GLCM+LDP)	96
GLCM	91
LBP	92.50
LDP	94
PCA (Gradient)	94.73
LDA	95.4
PCA	94

Table 4.5: AVERAGE VERIFICATION RATE ON EXTENDED YALE-B DATABASE

Features	Average Verification Rate (%)
GLCM+LBP+LDP (with DCT)	**96**
(GLCM+LBP)	91.3
(LBP+LDP)	93.8
(GLCM+LDP)	94
GLCM	88
LBP	90
LDP	93
PCA (Gradient)	88.13
LDA	87
PCA	86

4.2.3 AR Database

We first choose images of 50 classes of people. From each class, we choose 13 images. Hence, $|S_{Sim}| = 50 \times \binom{13}{2} = 3,900$ and $|S_{Dissim}| = \binom{50 \times 13}{2} - |S_{Sim}| = 210,925 - 3,900 = 207025$. We randomly choose 3,900 difference vectors from the set S_{Dissim} 30 times and perform the experiment.

Table 4.6: AVERAGE VERIFICATION RATE ON AR DATABASE

Features	Average Verification Rate (%)
GLCM+LBP+LDP (with DCT)	**95.84**
(GLCM+LBP)	92.67
(LBP+LDP)	93.83
(GLCM+LDP)	93.77
GLCM	88
LBP	91.56
LDP	93.1
PCA (Gradient)	88.54
LDA	87.56
PCA	85.70

Chapter 5

Conclusion

We have presented our methods for face recognition and verification under uncontrolled environment based on fusion of three local feature descriptors, namely (a) LBP, (b) GLCM and (c) LDP. The combination of these features provides very promising performance on three well-known face datasets that contain pose variation, partial occlusion and widely varying lighting conditions. There are three main contributions: (*i*) LBP histogram is computed for 3 different sizes of window in an uncoventional approach and DCT is employed for reducing the size of the feature vector; (*ii*) LDP histogram is calculated after partitioning the training image and again DCT is applied on it and (*iii*) GLCM is calculated for distance values of 1, 2 & 3 and on 16 partitions of the input image.

Secondly, multi-class problem is mapped to two-class problem to achieve better accuracy as well as to avoid retraining the classifier whenever new subject is included in the database.

Bibliography

[1] Patrick Ross, Anil K. Jain, Flynn. *Handbook of Biometrics.* Springer-Verlag New York Inc.

[2] R. Hietmeyer. Biometric identification promises fast and secure processing of airline passengers.*The Intl Civil Aviation Organization Journal, 5555,*, no. 9, pp. 10-11, 2000.

[3] T. Kohonen. *Self-organization and Associative Memory.* Springer-Verlag, Berlin, 1989.

[4] M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Pattern Analysis and Machine Intelligence*, 12, no. 1, pp. 103-108, 1990.

[5] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cog. Neuroscience*, 3, no. 1, pp. 71-86, 1991.

[6] R.M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, *IEEE Trans. Syst. Man Cybern.* vol. 3, no. 6, pp. 610621, 1973.

[7] A. Gelzinisa, A. Verikasa, M. Bacauskienea, Increasing the discrimination power of the co- occurrence matrix-based features, *Pattern Recognition*, vol. 40, pp. 2367-2372, 2004.

[8] T. Ahonen, A. Hadid, M. Pietikainen, Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037 2041, 2006.

[9] Jabid, T., Kabir, M.H., Oksam Chae , Local Directional Pattern (LDP) for face recognition, *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on* , vol., no., pp.329-330, 9-13 Jan. 2010

[10] R.Duda, P.Hart, D.Stork, *Pattern Classification, 2nd Ed.*, John Wiley & Sons, 2001.

[11] R. C. Gonzalez, & R. E. Woods, *Digital Image Processing, 3rd Ed.* Prentice Hall, 2008.

[12] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29, 1996.

[13] Tan, X. Triggs, B., Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions, *LECTURE NOTES IN COMPUTER SCIENCE*, SPRINGER-VERLAG Germany 2007, No. 4778, pages 168-182,

[14] Chang, Chih-Chung and Lin, Chih-Jen,LIBSVM: A library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, vol 2, issue 3, 2011, Pages 27:1–27:27, software available at `http://www.csie.ntu.edu.tw/∼cjlin/libsvm`