BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Robust Face Recognition Using SPIDER Descriptor

By

Farhan Bashar (084425)

Asif Khan (084439)

Supervised By

Dr. Md. Hasanul Kabir

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)
Organization of the Islamic Cooperation (OIC)
Dhaka, Bangladesh

October, 2012

Robust Face Recognition Using SPIDER Descriptor

By

Farhan Bashar

Student No: 084425

Asif Khan

Student No: 084439

Supervised By

Dr. Md. Hasanul Kabir

A Thesis Submitted to the Department of Computer Science and Engineering (CSE) in Partial Fulfilment of the Requirements for the Award of the Degree of Bachelor of Science in Computer Science and Engineering (B.Sc. in CSE).

Department of Computer Science and Engineering (CSE)

Islamic University of Technology (IUT)

Dhaka, Bangladesh

October, 2012

Abstract

Real world face recognition systems require balancing of three concerns : computational cost, robustness in changing environment and discriminative power. Designing a face feature having small size would reduce computation cost but would also have minimal discriminative power and may not be stable in uncontrolled environment and including more facial information to mitigate this challenge would increase the feature size and thus the computation cost. So, achieving a balance is a key aspect for every successful system.

In our thesis we introduce an effective appearance-based facial feature descriptor constructed with the new local texture pattern namely the Similarity Pattern of Image Directional Edge Response (SPIDER) for face representation and recognition. The proposed method encodes texture information of a center cell pixel by accumulating the directional edge response of all the pixels in the cell and then computing the dissimilarity measure of the local histogram of the center cell with its 8 neighbor cells using the Chi-square method. The dissimilarity values are then thresholded against a local average dissimilarity to generate a 8 bit-binary code which is assigned to the center pixel of the center cell. The distribution of the resulting SPIDER codes are then used as a face representation. To make the method compatible with real world systems a further step of dimension reduction is done to achieve fast classification time at the cost of a slight decrease in the accuracy. The effectiveness of the proposed method has been evaluated using the FERET face image database using template matching and experimental results shows better performance of the SPIDER feature descriptor in comparison to other well-known appearance based feature extraction methods.

Acknowledgment

First of all, we would like to thank the Almighty Allah for all the blessings he has bestowed upon us throughout our entire life, without the mercy of Allah, we couldn't make everything. "All thanks and praises be to Allah".

I would like to thank my thesis supervisor, Dr. Md. Hasanul Kabir, for his support and guidance on this thesis. Dr. Kabir has been an instrumental to this work and my career. He taught me how to do research, think critically and find out useful solutions. He provided me with a strong vision for what this should accomplish, without his insights the "Robust Face Recognition using SPIDER Descriptor" would not have reached completion. Sir may the Al-Mighty Allah reward you abundantly.

Again, it was our pleasure to get the cooperation and coordination from Head of The Department, Prof. Dr. M.A. Mottalib during various phases of the work. We are grateful to him for his constant and energetic supervision, constructive criticism and valuable advice.

Special thanks goes to the Faisal Ahmed, Lecturer, CSE Dept., Islamic University of Technology, for his kind suggestion and cooperation regarding implementation of our algorithm.

Finally, we like to offer thanks to all who contributed us in any way during the project.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Face Recognition

A facial recognition system is a computer application for automatically identifying or verifying a person from a digital image or a video frame from a video source. One of the ways to do this is by comparing selected facial features from the image and a facial database. It is typically used in security systems and can be compared to other biometrics such as fingerprint or eye iris recognition systems.

1.1.1 Application

In the modern era the advancement in hardware and software technology have increased the demand for personalized consumer products. This includes identifying users through face recognition and enabling the appropriate services such as personalized TV programs [1, 2], smart phones, smart homes and many others. Face recognition has also seen an increased use in the field of security and surveillance. The introduction of high resolution surveillance cameras have increased the demand for accurate face recognition systems. One of the reasons why face recognition has seen an increased use is that other bio metric methods such as retina scan, fingerprint scan requires the cooperation of the individuals while face recognition does not require the cooperation of the individual [1]. All that is required is an image which can be taken using a camera, sometimes even without the individual noticing.

1.1.2 Generic Steps of Face Recognition

Figure 1.1 shows the generic technique for face recognition.

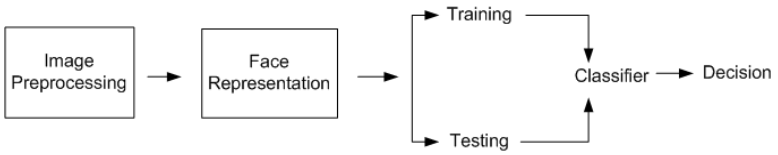

Figure 1.1: Generic Steps of Face Recognition

The steps common to all face recognition systems involves taking an image with a camera or some other input device and then cropping the image to keep only the face and discarding the unwanted background. Next, a filter or filter bank is applied on the image to generate a pattern which encodes the facial information associated with each image. Using the pattern a histogram is generated for the image which is used as the representation of each face. The system is then trained with a set of known (labeled) image and then some unknown face is classified to one of the known faces.

1.2 Problem Statement

From the generic steps of Face Recognition system shown in Figure 1.1, the most critical aspect is to find out the effective face descriptor. This face descriptor should be robust in uncontrolled environment like noise, pose variation, illumination variation, ageing, occlusion etc [3]. So, designing a face descriptor is a very critical task. Our objective is to design a face descriptor which overcomes all the stated problems. Also computational complexity is a great issue when recognizing a face from a huge database. So, our another objective is to make the size of the feature representation smaller for computational feasibility without significant deterioration of the recognition rate. Figure 1.2 shows the variation of same persons image. A good feature representation should have the ability to recognize the person in all those environment.

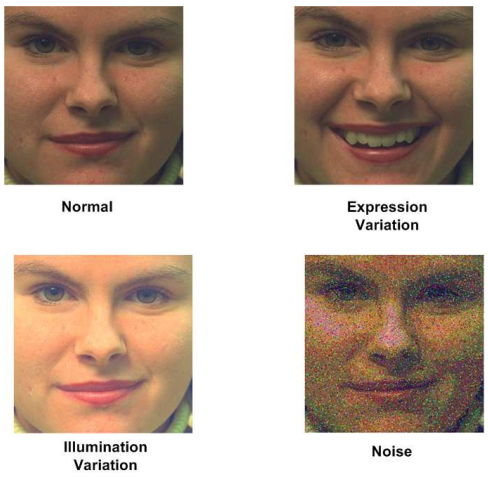

Figure 1.2: Same Person in Different Environment

1.3 Research Challenges

The challenge faced today by the researchers in the field of face recognition is to design a system which can classify facial images in uncontrolled environment with a high accuracy. In real world, facial image tend to be non-uniform because of factors such as lighting variation, pose variation, expression variation and occlusion. Moreover, the hardware used to capture the images may add some noise to the image. An ideal feature representation should be stable in all these environment and have high discrimination information which minimized the within class variations while maximizing the between class variation. In real world, the face recognition systems should not only be accurate but should also be able to classify images in quick time. A face representation which is robust in changing environment but requires a large time for classification is not suitable and is of little use in real time systems. On the other hand discarding discriminatory information in order to reduce the feature size would reduce the accuracy and reliability of the system. So, the major challenge is to extract a face representation which is robust in uncontrolled environment and at the same time having a small size while still holding the essential discriminatory information.

Many methods have been proposed in the last few decades and they can mainly be classified into two classes: global feature based approaches and local

feature based approaches. In global feature based approaches the entire image is used to extract facial information [26]. However, the entire face of a person is not the same and there are local regions which vary from region to region such as eyes, mouth and nose. The methods which use these different local features to extract information are called local feature based methods.Some common global appearance based feature extractors are Principal Component Analysis (PCA) [7], Linear Discriminant Analysis(LDA) [24] and Independent Component Analysis (ICA) [8].Features can also be extracted in the frequency domain by converting the image from spatial domain to frequency domain and using Gabor wavelets to extract the face features. Testing on different image databases has shown that the performance of global feature based methods degrade in changing environments with variation in illumination and pose. Hence, local features are getting popular for their robustness in such changing environment. Among the local feature based approaches some use geometric methods to generate the features based on the geometric relationships(distance, angle, positions) between different facial components(eyes, mouth and nose).The features are then used for classifying the images [5, 6]. However, the high computational cost and the pin-point accuracy required for selecting the components makes geometric approaches inappropriate for real time applications [3]. Other local based approaches use the local appearance for feature generation. One such method, known as the Local Binary Pattern (LBP) [14] , was proposed by *Ahonen et.al.* for recognizing faces. It generates binary codes to represent images. Although it is robust in monotonic illumination varia-tion, it suffers in environment having non-monotonic illumination variation and noise. Some extensions of LBP (Sobel-LBP) [19] also exist.To improve the performance of LBP, Local Ternary Pattern (LTP) was proposed by *Tan and Triggs* [15]. It adds a threshold with the intensity of the center pixel and generates the face features. It is still not robust in presence of noise. Motivated by the performance of LBP and LTP,researchers started utilizing edge and gradient information of each pixel instead of the intensity. This gives robustness in presence of noise. Local Directional Pattern (LDP) [3], Directional Ternary

Pattern (DTP) [20] and Patterns of Oriented Edge Magnitude (POEM) [21] uses edge information of each pixel and encodes them to use as the face feature representation. However, the size of the feature vector generated by the above methods is very large and is computationally very expensive.

1.4 Research Contribution

In this field of research, we have introduced a face descriptor namely, Similarity Pattern of Image Directional Edge Magnitude (SPIDER). It uses directional edge responses and dissimilarity method to generate an efficient face representation. Using SPIDER representation, the recognition rate increases and also it is robust in uncontrolled environment like noise, illumination variation, pose variation, ageing etc. Also for computation feasibility, we have used DCT to reduce the feature dimension. Though the performance degrades a little bit, we used the reduced feature vector as the final face representation by thinking about the complexity of recognizing among millions of images.

1.5 Thesis Organization

This thesis book has four chapters. Chapter 1 introduced about the facial recognition system. Also it described the problem domain and research challenges of face recognition system. Chapter 2 gives us a overview of Related work on this face recognition system where we can get the idea of how far the research has gone in this field. Chapter 3 describes our proposed face descriptor named SPIDER. Chapter 4 shows the analysis of different algorithms for face recognition. In Chapter 5 we have referred to some future plan with this algorithm for better face recognition accuracy. References are added in the end of this book.

Chapter 2

Literature Review

2.1 Feature Extraction

For any Face Recognition system the most critical step is the extraction of features from each facial images. The challenge is to extract features which are robust to noise, illumination variation, pose variation, occlusion and aging. Currently researchers have focused on feature extraction using local appearance based micro-patterns.

2.1.1 Local Binary Pattern

Ojala et al. (2002) first proposed LBP as a gray-scale invariant method for texture analysis. For analysing the texture variations in a particular region, this method was used [18]. Finding the texture pattern helps to find the similarity between two images. This is one of the useful micro-pattern based operator in recent advances. It takes P local neighbor values around each pixel and generates a P-bit binary code by thresholding the intensity value of the neighbour pixels with respect to the intensity of the center pixel. The binary code of LBP is generated using equation (2.1)

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_c - i_p) \tag{2.1}$$

where i_c denotes the gray value of a center pixel (x_c, y_c) and i_p corresponds the gray value of the local neighborhood pixels. The function $s(x)$ is defined with Equation (2.2)

$$s(x) = \begin{cases} 1 & x > 0 \\ 0 & otherwise \end{cases} \tag{2.2}$$

In practise, Equation (2.1) means that the signs of the differences in a neighborhood are interpreted as a P-bit binary number, resulting in 2^P distinct values for binary pattern. The individual pattern value is capable of describing the texture information at the center pixel i_c. The process of generating P-bit binary number is shown is Figure 2.1 where $P = 8$. It takes a 3×3 neighborhood and finds the LBP code using Equation (2.1) and (2.2).

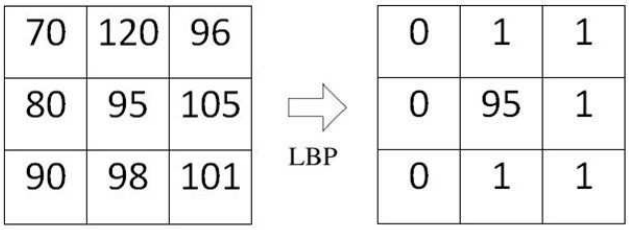

Figure 2.1: Illustration of the Basic LBP Operator

In Figure 2.1, applying LBP code generation process we get the LBP code = 00111110. This code tells us that there are 2 transition in the local neighborhood pixel. This transition is very important for finding the significant area of an image. One of the variations of this original LBP is uniform LBP where the LBP patterns are named as uniform as they contain very few transitions from 0 to 1 or 1 to 0 in circular bit sequence. For example the patterns 00000000 and 11111111 have zero transitions, 00011000 has two transitions and 10001101 has four transitions. *Ahonen et al.* [14] used this variant LBP patterns in FERET database for face recognition and found good recognition accuracy in monotonic illumination change. Also this variant is still sensitive for low amount of noise and non monotonic illumination variation. He applied the LBP operator in each pixel for generating a LBP code and assign the decimal value of this code to that pixel. Finally, a histogram of the encoded LBP image is created which is used as

a feature vector for face recognition. The general feature size obtained from LBP encoded image is 256. This feature vector generation process is shown in Figure 2.2.

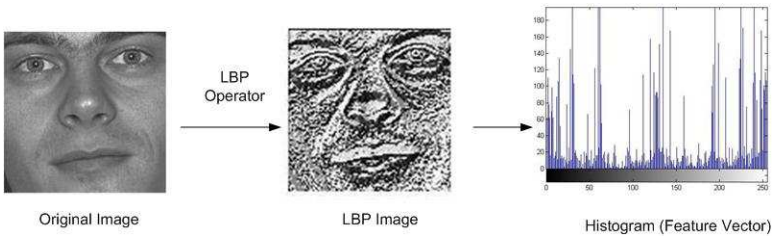

Figure 2.2: Illustration of LBP Feature Vector Generation Process

LBP is a intensity based operator. So due to some noise in a pixel will create great change in the LBP code. Thus noise will affect on the performance of LBP operation. Same thing will happen due to large change in illumination. Then we also have changes in intensity value and thus creates a different LBP code.

2.1.2 Sobel-LBP

Sobel operator with LBP can enhance the Local features, and thus more detailed information can be extracted from LBP operation. This new operator is namead as Sobel-LBP proposed by *Zhao et. al.* (2009) [19]. The sobel operator contains two 3×3 kernals (horizontal kernel S_x and vertical kernal S_y) which are convolved with the original image I to calculate gradient approximations.

$$I^x = S_x * I = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * I, I^y = S_y * I = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * I$$

Figure 2.3: Gradient Computation Along X and Y axis Using Sobel mask

In Figure 2.3, I^x and I^y represent the horizontally and vertically filtered results, respectivel. Normally I^x and I^y are combined to give gradient magnitude

$\sqrt{I^{x2} + I^{y2}}$. Here Sobel-LBP operator is defined as the concatenation of LBP operations on I^x and I^y

$$Sobel - LBP_{P,R} = \left\{Sobel - LBP^x_{P,R}, Sobel - LBP^y_{P,R}\right\} \tag{2.3}$$

where,

$$Sobel - LBP^x_{P,R} = \sum_{P=0}^{P-1} s(I^x_{P,R} - I^x_c)2^P \tag{2.4}$$

$$Sobel - LBP^y_{P,R} = \sum_{P=0}^{P-1} s(I^y_{P,R} - I^y_c)2^P \tag{2.5}$$

Though it is the first method used directional edge responses, it overcomes the problems of LBP slightly. But accuracy is not so convenient due to the increase in feature vector. So, new methods are proposed to overcome the problems using 8-directions instead of 2.

2.1.3 Local Ternary Pattern

LBPs have proven to be highly discriminative features for texture classification and they are resistant to lighting effects in the sense that they are invariant to monotonic gray-level transformations. However because they threshold at exactly the value of the central pixel i_c they tend to be sensitive to noise, particularly in near-uniform image regions, and to smooth weak illumination gradients. Many facial regions are relatively uniform and it is legitimate to investigate whether the robustness of the features can be improved in these regions. This section extends LBP to 3-valued codes, Local Ternary Patterns (LTP) proposed by , *Tan and Triggs* (2009) [15], in which gray-levels in a zone of width t around i_c are quantized to zero, ones above this are quantized to +1 and ones below it to -1, i.e. the indicator $s(u)$ is replaced with a 3-valued function shown in Equation (2.4)

$$S(u, i_c, t) = \begin{cases} 1, & u \geq i_c + t \\ 0, & |u - i_c| < t \\ -1, & u \leq i_c - t \end{cases} \tag{2.6}$$

When using LTP for visual matching we could use 3- valued codes, but the uniform pattern argument also applies in the ternary case. For simplicity, the experiments below use a coding scheme that splits each ternary pattern into its positive and negative halves as illustrated in Fig. 2.4, subsequently treating these as two separate channels of LBP descriptors for which separate histograms and similarity metrics are computed, combining the results only at the end of the computation.

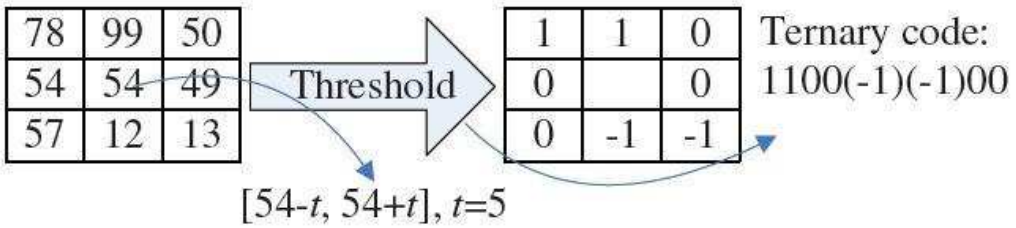

Figure 2.4: Illustration of the Basic LTP Operator

Figure 2.5 shows the LTP feature vector generation process. Applying LTP operator, we get the Positive and Negative LTP image. From this two images we can get two separate histograms. This two histograms are then concatenated to generate the full feature vector.

2.1.4 Local Directional Pattern

The Local Directional Pattern (LDP), proposed by *Jabid et.al* (2009), uses the 8 directional edge responses to generate a 8-bit binary code for each pixel. The use of directional edge responses makes LDP more robust to noise and illumination variations [3]. A 3×3 local window is centered on each pixel of the image and

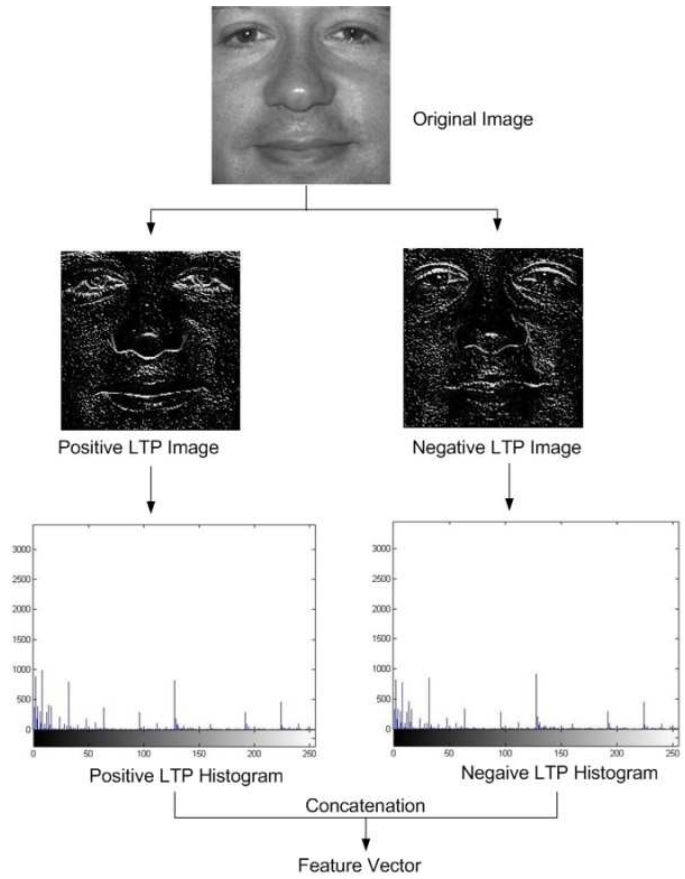

Figure 2.5: Illustration of LTP Feature Vector Generation Process

$$\begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}$$ East (M_0)

$$\begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix}$$ North East (M_1)

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$ North (M_2)

$$\begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$ North West (M_3)

$$\begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix}$$ West (M_4)

$$\begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix}$$ South West (M_5)

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix}$$ South (M_6)

$$\begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$ South East (M_7)

Figure 2.6: Krisch Eight Directional Edge Response Masks

Kirsch masks are used to calculate the 8 directional edge responses. The Kirsch masks are shown in Fig Fig 2.6.

After finding the direction in the 8-direction, the K most significant bits are assigned a value of 1 and the rest are assigned 0. An 8-bit binary code is generated

and the corresponding integer value is assigned to the center pixel. Apart from using the Kirsch masks, the Preweitt masks and the Sobel masks can also be used to calculate the edge responses but Kirsch mask is known to detect edge responses more accurately. Each edge response is calculated based on the intensity values of the neighbor pixels. LDP generates stable codes in presence of noise because the weak responses are ignored, which are more prone to error due to noise. The LDP code is generated using Equation (2.7)

$$LDP_k = \sum_{i=0}^{7} s(m_i - m_k) \times 2^p \tag{2.7}$$

Figure 2.7 shows the LDP feature vector generation process. First after applying the Krisch mask, we get the 8-directional edge responses and then applying LDP operator, we get the LDP image and its histogram is used as the feature vector.

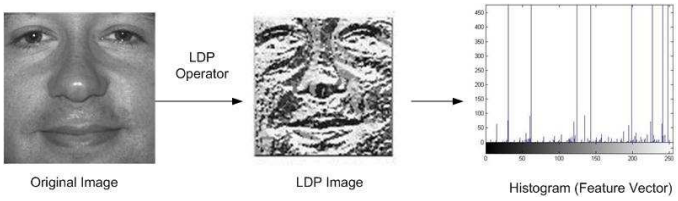

Figure 2.7: Illustration of LDP Feature Vector Generation Process

LDP is heavily dependent on the number of prominent edge responses and still generates unstable codes in smooth and uniform regions. Moreover, if there is more than one edge responses equal to the kth most significant value then no method is defined to select which one to assign the bit 1. Thus, the LDP code would vary in such cases.

2.1.5 Directional Ternary Pattern

To overcome the problems of LDP, a new local pattern based feature descriptor called Directional Ternary Pattern (DTP) was proposed by *Ahmed and Kabir* (2012) [20] . DTP calculates the edge responses of each pixel and then quantizes

them in 3 levels (-1,0,1). A 3×3 local window is centered on each pixel and the 8 directional edge responses are calculated using the Robinson mask . The Robinson mask is shown in Fig 2.8.

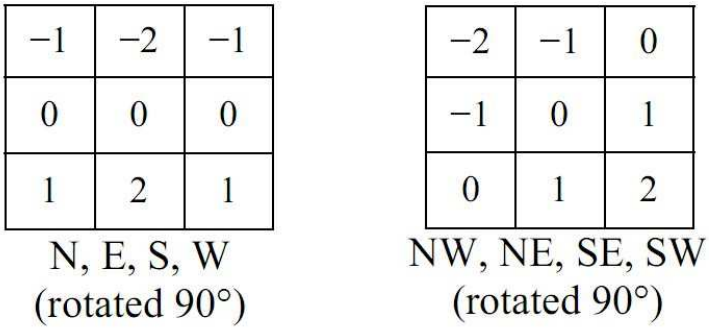

Figure 2.8: Robinson Eight Directional Edge Response Masks

The presence of edges or corners will produce high responses in the respective directions while a smooth region will produce similar responses in all directions. The average, avg, of the 8 responses is calculated and then the responses are quantized into one of 3 levels with respect to the average and some threshold, t. The values in range $avg-t$ to $avg+t$ are given a value 0. The ones above $avg+t$ are given 1 and the rest are given a value of -1. Usually a value of 10 is used as the threshold, i.e. $t = 10$. DTP produces stable codes in both smooth and high-textured regions even in the presence of noise. Quantization gives robustness against factors like illumination variation and aging. The general Equation of DTP is shown in (2.8)

$$S_P(v) = \begin{cases} 1, & v > \mu + 1 \\ 0, & \mu - 1 \leq v \leq \mu + 1 \\ -1, & v < \mu - 1 \end{cases} \tag{2.8}$$

where μ is the average edge response and t is a margin threshold. To reduce the length of the feature vector, each DTP code is further split into its corresponding positive and negative parts, and treated as two separate binary patterns, named as P_DTP and N_DTP by using the Equation (2.9) and (2.10)

To reduce the length of the

$$P_{DTP} = \sum_{i=0}^{7} S_P(S_{DTP}(r_i))2^i, S_P(v) = \begin{cases} 1, & v = 1 \\ 0, & otherwise \end{cases} \tag{2.9}$$

$$N_{DTP} = \sum_{i=0}^{7} S_N(S_{DTP}(r_i))2^i, S_N(v) = \begin{cases} 1, & v = -1 \\ 0, & otherwise \end{cases} \tag{2.10}$$

where r_i is the edge response value in the i-th direction. Figure 2.9 shows the DTP feature vector generation process. After applying Robinson mask, we will get 8-directional edge responses and then after applying DTP operator, we will get two images- one for positive and the other for negative. The histograms of this positive and negative DTP are concatenated together to form the full DTP feature vector.

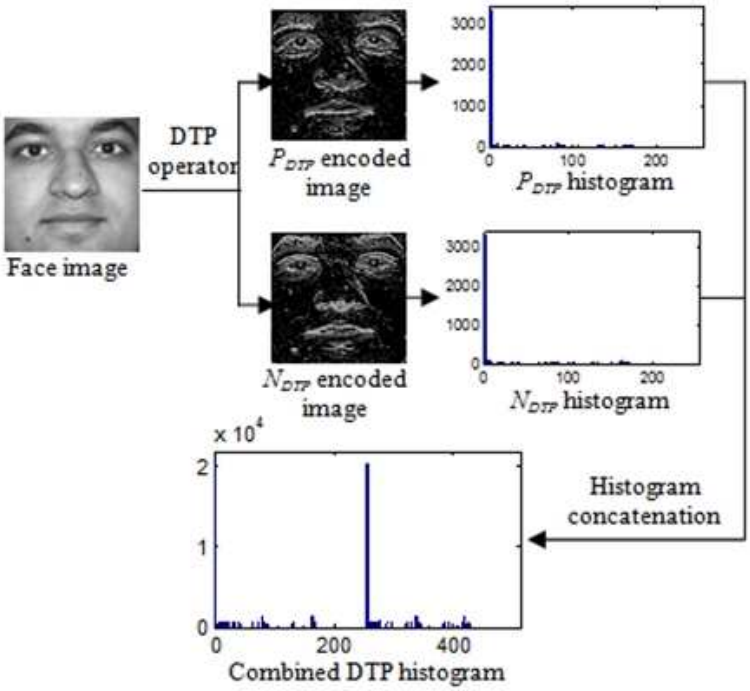

Figure 2.9: Illustration of DTP Feature Vector Generation Process

2.1.6 Patterns of Oriented Edge Magnitude

Pattern Oriented Edge Magnitude (POEM) is the LBP based structure applied on the oriented edge magnitudes developed by *Vu et. al.* (2010) [21]. Traditional LBP uses "cell" to find the feature vector but POEM uses a new term called "block". Block refer to more extended spatial regions where the LBP is applied. In this method, at first the x-directional and y-directional gradient is calculated using Sobel mask. Gradient magnitude and orientation of each pixel is then calculated using the gradient image. This orientation is then evenly discretised over $0 - 2\pi$. If the number of discretised orientation is m, then we referred this as the number of bins.

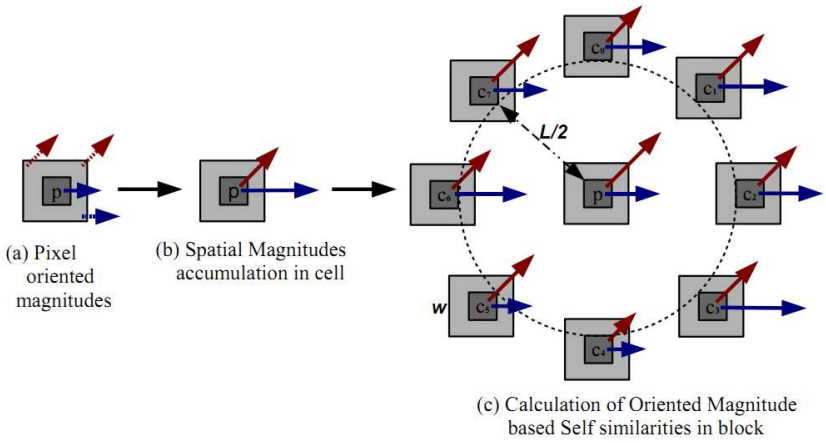

Figure 2.10: Illustration of POEM

The second step is to incorporate gradient information from neighbouring pixels by computing a local histogram of gradient orientation over all cell pixels. So, for each pixel a feature vector of m length is created. Finally, the accumulated magnitudes are encoded using the LBP operator within a block. The original LBP labels the 3×3 neighboring pixels by thresholding it with respect to the center pixel. In POEM this procedure is applied on the accumulated gradient magnitudes and across different directions. At the pixel p, a POEM descriptor is calculated for each discretion direction θ_i using Equation (2.11)

$$POEM^{\theta_i}_{L,w,n}(p) = \sum_{j=1}^{n} f(s(m^{\theta_i}_{p}, m^{\theta_i}_{c_j}))2^j \tag{2.11}$$

where m_p, m_{c_j} are the accumulated gradient magnitudes of the central and surrounding pixels p, c_j. $s(.,.)$ is the similarity function (e.g., the difference of two gradient magnitudes). L and w refer to the size of the blocks and cells respectively. n is the number of pixels in each cell which is set 8 by default. The function $f(x)$ is defined as:

$$s(x) = \begin{cases} 1 & x > \tau \\ 0 & otherwise \end{cases} \tag{2.12}$$

where the value τ is slightly larger than zero to provide some stability bin uniform regions. The final POEM descriptor at each pixel is the concatenation of these POEMs for each m-orientations:

$$POEM^{\theta_i}_{L,w,n}(p) = \left\{ POEM^{\theta_1}, ..., POEM^{\theta_m} \right\} \tag{2.13}$$

2.2 Feature Dimension Reduction

2.2.1 Principal Component Analysis

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [7]. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. It is a way of identifying

patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to nd in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once we have found these patterns in the data, and we can compress the data, i.e. by reducing the number of dimensions, without much loss of information.

There are a number of steps for computing PCA. First we have to take some data and form a matrix. Second, mean is subtracted from each row of the matrix. Third, Covariance matrix is calculated and by using that covariance matrix, eigenvectors and eigenvalue is calculated. Finally, we have to take the most significant eigenvalues. Significant eigenvectors are found by sorting their corresponding eigenvalues [10]. Normally, we take 98 percent of the total energy of the image and use that eigenvector for transformation matrix. This transformation matrix is for projecting the data to that dimension. After projection, we get our compressed feature vector. A projection of 2-dimensional feature vector is shown in Figure 2.11

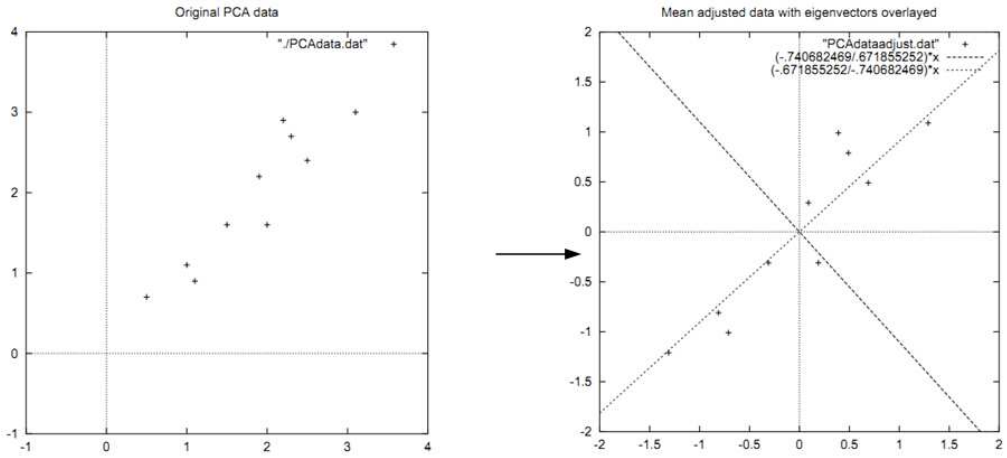

Figure 2.11: PCA Data Correlation Analysis

2.2.2 Discrete Cosine Transformation

A discrete cosine transform (DCT) expresses a sequence of finitely many data points in terms of a sum of cosine functions oscillating at different frequencies [23]. DCTs are important to numerous applications in science and engineering, from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical in these applications: for compression, it turns out that cosine functions are much more efficient, whereas for differential equations the cosines express a particular choice of boundary conditions. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. It uses the fourier domain for compressing data. Althoug, since the DCT is related to the DFT, it can be computed efficiently. There are four types of DCT , DCT I , DCT II , DCT III and DCT IV. Among them, DCT II performs well for data compression. DCT of an image can be calculated using Equation (2.14).

$$F(u,v) = \sqrt{\frac{2}{N}}\sqrt{\frac{2}{M}}\sum_{i=0}^{N-1} A(i)\times cos\left(\frac{u(2i+1)\pi}{2N}\right)\times\sum_{j=0}^{M-1} A(j)\times cos\left(\frac{v(2j+1)\pi}{2M}\right)\times f(i,j) \tag{2.14}$$

where,

$$A(i) = \begin{cases} \frac{1}{\sqrt{2}} & u = 0 \\ 1 & otherwise \end{cases} \tag{2.15}$$

$$A(j) = \begin{cases} \frac{1}{\sqrt{2}} & v = 0 \\ 1 & otherwise \end{cases} \tag{2.16}$$

2.2.3 Linear Discriminant Analysis

Linear discriminant analysis (LDA) is a method used in statistics, pattern recognition and machine learning to find a linear combination of features which char-

acterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier, or, more commonly, for dimensionality reduction before later classification. The objective of LDA is to perform dimensionality reduction. It preserves as much of the class discriminatory information as possible. The main idea of LDA is to maximize the between class distance and minimize the within class scatteredness [24].

LDA is also closely related to principal component analysis (PCA) in that they both look for linear combinations of variables which best explain the data. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique: a distinction between independent variables and dependent variables (also called criterion variables) must be made. The Fisher linear discriminant is defined as the linear function $w^T x$ that maximizes the criterion function:

$$J(w) = \frac{|\mu_1 - \mu_2|^2}{s_1^2 + s_2^2} \tag{2.17}$$

where

$$\mu_i = \frac{1}{N} \sum_{x \epsilon \omega_i} x \tag{2.18}$$

$$s_i^2 = \sum_{y \epsilon \omega_i} (y - \mu_i)^2 \tag{2.19}$$

2.3 Classification

There are several classification techniques available i.e. SVM, template matching. Among them we have focused on Template matching mainly.

2.3.1 Template Matching

2.3.1.1 Chi-Square Dissimilarity Measure

For generating better feature vector, the input image is first divided into sub regions (typically 3×3, 5×5 and 7×7). Feature extraction method is applied to each sub region and their respective histograms are generated. Histograms are the frequency of the grey values in each sub regions. All the histograms are then concatenated to form a combined histogram which is used as the feature representation of the input image. This feature vector size is different for each local subdivision. For 3×3 subdivision, the total length of LBP/LDP feature vector becomes: $3 \times 3 \times 512 = 4608$, and for 7×7 it becomes $7 \times 7 \times 256 = 12544$. On the other hand, For 3×3 subdivision, the total length of LTP/DTP/Sobel-LBP feature vector becomes: $3 \times 3 \times 256 = 2304$, and for 7×7 it becomes $7 \times 7 \times 256 = 25088$. Figure 2.12 shows the localization of images into several parts and concatenating their histograms together.

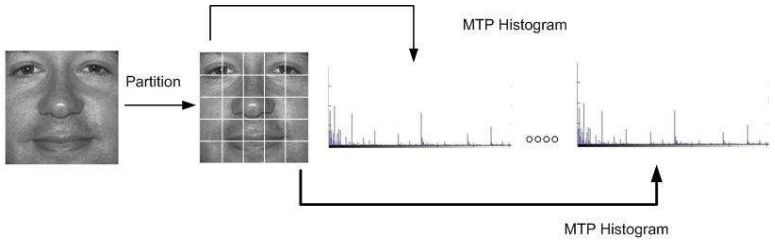

Figure 2.12: Localization of Images into Several Parts and Concatenating the Histograms Together to Generate Feature Vector

The histogram of the input image is compared with the histograms of the image gallery using the Chi square method and the input is matched with the galley image which gives the smallest Chi-Square difference. The Chi-Square difference is calculated using Equation (2.20). After calculating all classes' dissimilarity, The class which gives the lowest dissimilarity value is given to the input image.

$$\chi_w^2(H^1, H^2) = \sum_{i,j} w_i \frac{(H^1 - H^2)^2}{(H^1 + H^2)} \tag{2.20}$$

Where, H^1 nad H^2 are histograms of input and trained image respectively. w_i = weight assigned to the $i-th$ face region. Figure 2.13(a) shows an example of a facial image divided into 7×7 windows. Figure 2.13(b) shows the weight set for weighted χ^2 dissimilarity measure. Black squares indicate weight 0.0, dark grey 1.0, light grey 2.0 and white 4.0

(a) (b)

Figure 2.13: Localization of Images and Weight Adjustment of Corresponding Local Images

2.3.1.2 Euclidean Distance Measure

After compressing the feature vector with dimensionality reduction techniques, the feature length becomes comparatively smaller than the original ones. Here we have use Euclidean distance measurement for finding the dissimilarity between two classes. Euclidean distance can be measured using Equation (2.21)

$$d = \sqrt{(f_0 - v_0)^2 + (f_1 - v_1)^2 + (f_2 - v_2)^2 + \cdots + (f_{M-1} - v_{M-1})^2} \quad (2.21)$$

where

$$\begin{aligned} v &= [\ v_0 \quad v_1 \quad \cdots \quad v_{M-1}\ ]^T \\ f &= [\ f_0 \quad f_1 \quad \cdots \quad f_{M-1}\ ]^T \end{aligned} \quad (2.22)$$

Chapter 3

Proposed Method

3.1 Proposed Method

3.1.1 Overview

The recent challenge in face recognition systems is to extract facial features which are robust in changing environment. The feature size should also be small for fast classification time. In this section we propose a feature extraction method using directional edge responses and then generating a similarity code within each block. A dimension reduction method is then described to reduce the size of the feature vector and finally a classification method is described which is used for labeling an unknown image with one of the known gallery image.

3.1.2 Proposed Face Descriptor: SPIDER

Here we propose a face descriptor, Similarity Pattern of Image Directional Edge Response (SPIDER), for generating feature representation of facial images which is robust in presence of noise and illumination and pose variation. In the pre-processing step the image is first cropped based on the position of the eyes, nose and mouth and the face descriptor is applied on the resulting cropped image to generate a face representation. In our proposed method, the first step is to generate the 8 directional edge responses of the image using the Robinson mask. Robinson mask used for the 8 directions are shown in Figure 3.1

Applying the 8 masks generates 8 responses each representing the edge significance in its respective direction. Figure 3.2 shows a 3×3 block of an image

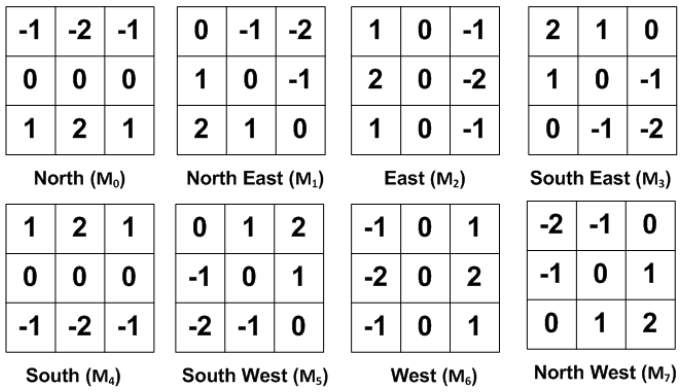

Figure 3.1: Eight Directional Robinson Mask

and the value of the edge responses as a result of placing the Robinson masks in the center pixel of the block. The initial value in the block is the intensity of the 9 pixels in the block. Likewise each pixel of the image generates 8 directional responses.Edge response magnitudes are more stable than intensity values in presence of noise and illumination variation [3] and the use of edge response in our method makes it more robust in presence of noise and lighting variation.

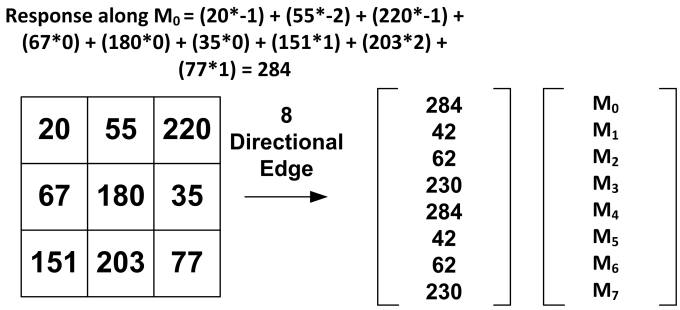

Figure 3.2: Edge Response Calculation Along 8 Directions

After calculating the edge responses a 3×3 neighborhood is defined as a cell around each pixel and the edge responses in a particular direction of all the pixels in the cell is accumulated and assigned as the cumulative response value to the respective direction of the center pixel. The accumulated responses are used to generate a local histogram for each cell. The histogram consists of 8 bins, one for each direction and the value of each bin is the accumulated edge response of the cell in each of the 8 directions. The local histogram of the 3×3 neighborhood of

each pixel is calculated and at each pixel now the feature is a vector of 8 values (the value of the 8 bins). Accumulation of the edge responses are shown in Figure 3.3

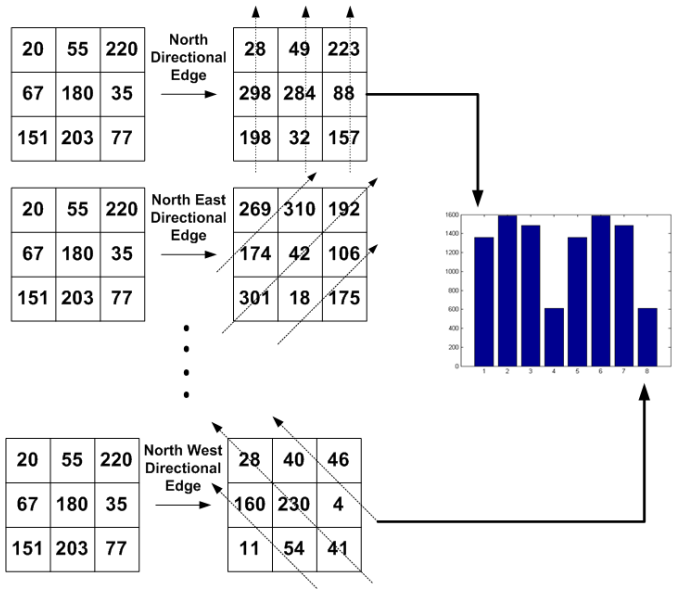

Figure 3.3: Accumulation of the Edge Responses of 8 Neighborhood Cell

A block is defined as the 3x3 neighboring cells around a pixel. The local histogram of the center cell is compared with the local histogram of the neighbor cells using Chi-square dissimilarity method. This method gives the dissimilarity between two histograms and is calculated using the following formula:

$$\chi_w^2(H_1, H_2) = \sum_i \frac{(H_1 - H_2)^2}{(H_1 + H_2)} \tag{3.1}$$

Calculation of the dissimilarity around 3×3 cell is shown in Figure 3.4

After computing the dissimilarity of the center cell with its 8 neighbor cells the average dissimilarity is used as a threshold. The cells whose dissimilarity with the center cell is greater than the average dissimilarity are assigned a value of 0 and the rest are assigned a value of 1. This generates a 8-bit code which is assigned to the center pixel of the center cell. The histogram matching is done for the cells centered on each pixel and now each pixel contains a single value between 0

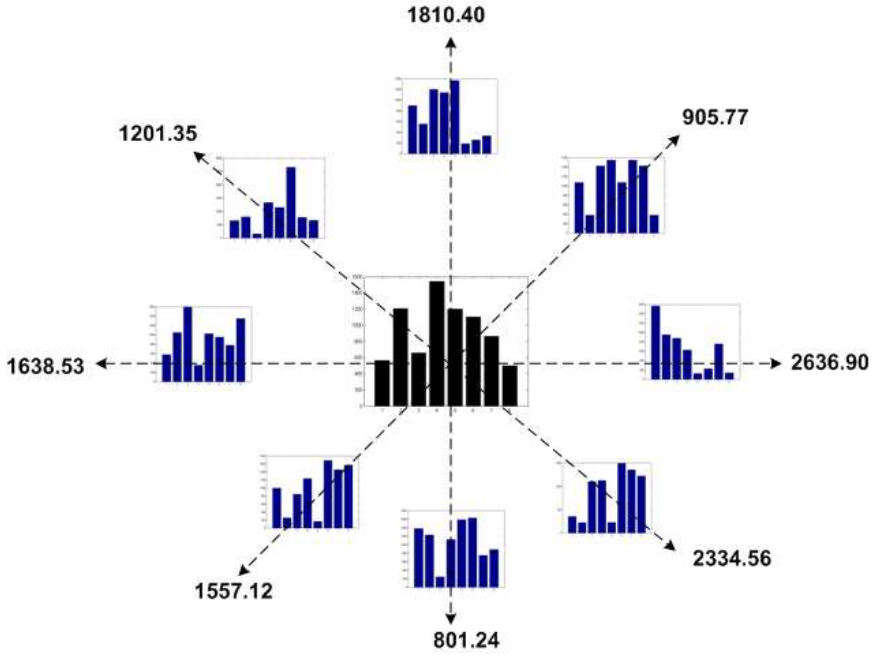

Figure 3.4: Similarity Measurement in a Block

and 255. Figure 3.5 shows the generation of SPIDER code. The resulting image is the coded image and a histogram is generated from this image.

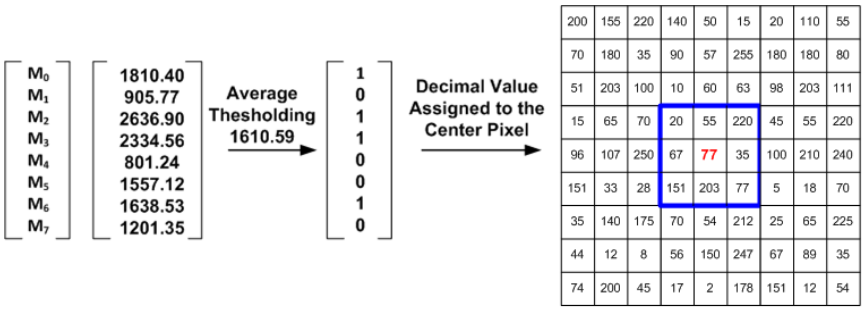

Figure 3.5: Generation of a SPIDER code

The resulting image contains fine details of the image such as edges, corners and information about other high textured regions but histogram computed over the entire image contains only the occurrence of the micro-patterns without any knowledge about their locations. To extract the local information of the image it is divided into $n \times n$ sub-regions and the local histogram of each sub-region is generated. Here the local histogram consists of the frequency of values coded at

each of the pixels in each region and consists of 256 bins. The local histograms of all the sub-regions are concatenated to form a combined histogram and this histogram is used as the feature representation of each image. Figure 3.6 shows the histogram generation process.

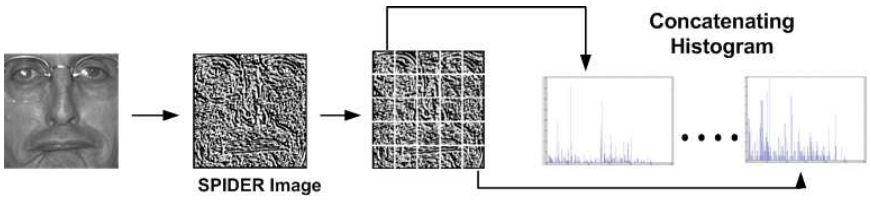

Figure 3.6: Illustration of SPIDER Feature Vector Generation Process

3.1.3 Feature Dimension Reduction

For a good recognition system it is critical that the feature representation vector contain essential information to make the classification task easier and accurate. Inadequate number of features would result in poor performance. However, using a feature vector having a very large size is also undesirable as it would increase the classification time without any significant increase in accuracy and also makes it unsuitable for use in real time systems. So, Dimensionality Reduction (DR) techniques are proposed to address the issue of large feature size [25]. Dimension reduction techniques transforms the original data into a lower dimension space without loosing significant discriminatory information. The DR functions can be divided into two groups:

1. Functions which transforms the features into a new reduced feature set.
2. Functions which select a subset of existing features.

In our proposed method we use a DR function, Discrete Cosine Transform (DCT), to reduce the size of the feature vector generated using the proposed face descriptor. DCT falls into the first category of the DR techniques and shows optimal performance in reducing feature size as well as keeping discriminatory information. DCT transforms the feature vector from spatial domain to frequency domain using Equation (2.14)

Fig.3.7 shows that most of the total energy lies in a subset of the total DCT co-efficients and this subset is retained while the other DCT coefficients are discarded. The DCT coefficients are chosen using a zigzag fashion as shown in Fig. 3.7. The retained subset is used as the final representation of each facial image. This reduces the size of the feature vector while still containing most of the essential information necessary for effective classification.

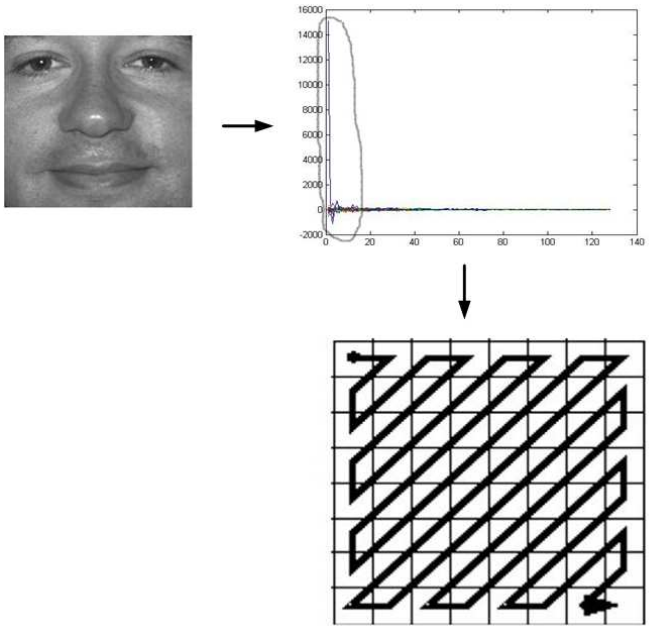

Figure 3.7: DCT Co-efficient Generation

3.1.4 Face Recognition Using The Extracted Feature Representation

Applying the proposed face descriptor on an image generates a transformed image which is then divided into nxn sub-regions and the local histogram of each region is concatenated to form the feature vector of the image. The feature vector of the known images (gallery image) is generated and stored beforehand. Upon receiving an unknown image the face descriptor is first used to extract the feature vector of the unknown sample and then the vector is compared with the feature vector of the known samples using template matching. Template matching makes use of

the Chi-square dissimilarity method for comparing images. The unknown sample is assigned a class label among the known images whose feature vector gives the minimum dissimilarity with the feature vector of the unknown image. Some regions of a face such as eyes, mouth and nose have more discriminatory properties than some other regions in the face and to provide more accurate matching a weighted Chi-square matching technique is used which assigns different weights to different regions of the face. The weighted Chi-square dissimilarity is measured using the formula:

$$\chi^2_w(H_1, H_2) = w_i \sum_i \frac{(H_1 - H_2)^2}{(H_1 + H_2)} \tag{3.2}$$

3.1.5 Properties of the Proposed Descriptor

Here we discuss the good properties of our proposed face descriptor for face recognition. For each pixel the proposed descriptor contains information about the local region as well as the information about the neighboring regions. Some properties of the method are:

Since the method uses directional responses it has the ability to capture face information in multiple directions.

Directional responses are more stable than pixel intensity [3] and using the directional response instead of pixel intensity make the method more robust to noise and illumination variation. As shown in [3] and [20] directional responses are insensitive to noise and light.

The use of cells captures local information around every pixel and calculating the dissimilarity between neighboring cells captures multi-scale texture information between image regions. Thus the pattern for every pixel generated by our method contains both local information as well as more global structure and this make it more robust to pose and expression variation and occlusion.

Reducing the feature dimension drastically reduces the classification time of the images and this makes it suitable for use in real time recognition systems.

Chapter 4

Experimental Result

4.1 Experimental Setup

We have used several tools to check the accuracy of the experimental data and analyzed it. For face database, we have used FERET database. The FERET database consists of 4 different set of Images [17]. Different set of images are taken under different conditions. Figure 4.1 shows the different set of FERET database.

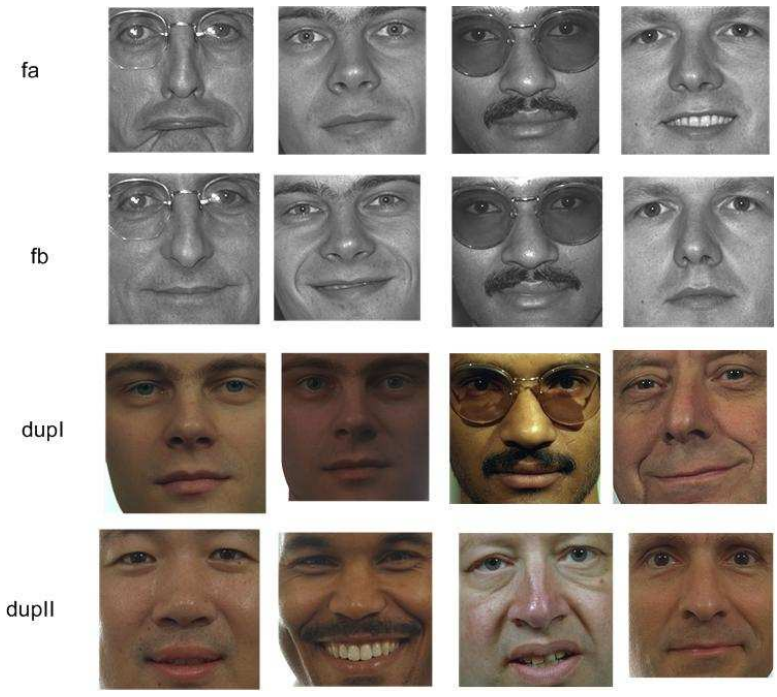

Figure 4.1: The Images of FERET Database

Threre are over 2200 images in FERET database [17]. Table 4.1 shows the

Table 4.1: FERET Database Gallery and Probe Set

Set	Property	No. of Images
fa	Training Set	766
fb	Experession Variation	766
dupI	Images taken later in time	500
dupII	subset of dupI taken a year later	180

number of images in different probe sets of FERET database.

4.2 Selecting Optimal Parameters

4.2.1 Determining the Value of Threshold, K

Table 4.2: Determining The Value Of Threshold, K

K	fb(%)	dupI(%)	dupII(%)
1	92.1	63.0	59.44
2	92.03	65.0	56.67
3	90.45	63.0	55.56
4	90.73	63.0	56.67
Average	95.2	72.4	71.11

Table 4.2 shows the accuracy while varying the threshold value. From, the table it can be deduced that using the average dissimilarity value in a block as the threshold gives the best performance. In a region where the dissimilarity values vary by a large margin, using a static threshold method, such as giving the lowest 3 dissimilarity value cells a value of 1 would generate the same pattern as in a region where the dissimilarity values are very close. However, the texture in the two regions are different which cannot be encoded in the pattern when using a static threshold method. Averaging gives a dynamic allocation of threshold value where the encoded pattern would differ for a high textured region when compared

with the one in a low textured region.

4.2.2 Determining the Number of Sub Region

Table 4.3: Determining the Number of Subregion

No. of Region	fb(%)	dupI(%)	dupII(%)
1×1	60.5	19.04	15.08
3×3	89.03	50.27	56.42
5×5	95.2	72.4	71.11
7×7	95.3	74.4	76.53

Table 4.3 gives the classification performance for different sub-region size. Generating the face representation without diving the image in sub-regions only gives the frequency of each code in the image and does not provide any information about the location of each code. Two completely different images having the same code frequency would be labeled as the same image which obviously would be a misclassification. Feature vector generated without dividing the image into sub-regions would not contain complete information of the face and this is exactly reflected by the experiment results. The accuracy for 1×1 division gives a very low accuracy and once some degree of subdivision in introduced the accuracy changes drastically because the local location of each pattern in also incorporated in the feature representation. Increasing the number of subdivision increases the local information and thus provides better accuracy but on the other hand increasing the number of subdivisions also increases the feature length and the classification time. A trade-off between time and accuracy have to be made and we have selected 5×5 sub division as the optimal choice. Figure 4.2 shows the graph of SPIDER accuracy in different sub-regions.

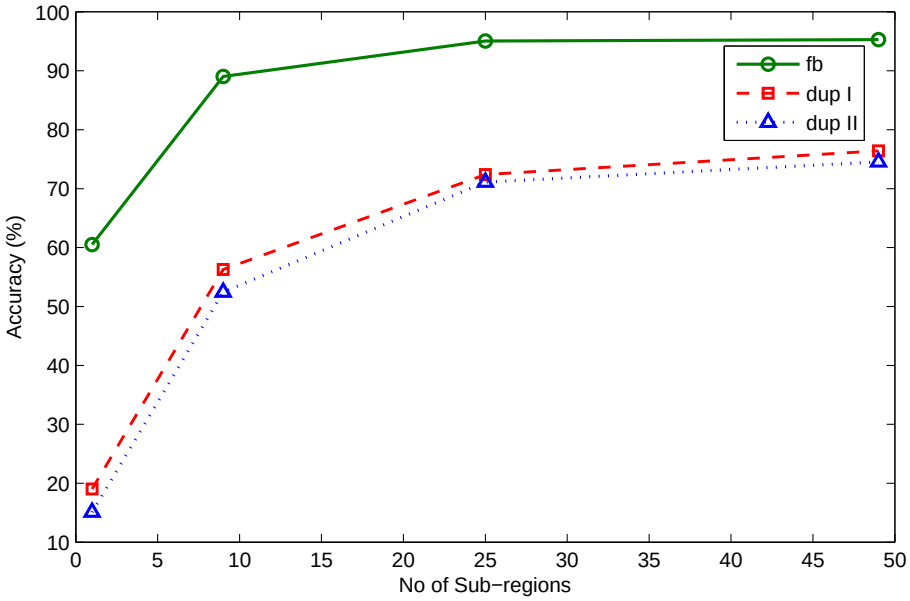

Figure 4.2: Accuracy Varying the Subregion

Table 4.4: Selection of Reduction Technique

Method	fb(%)	dupI(%)	dupII(%)	Feature Size
PCA	83.4	57.6	54.6	650
LDA	84.22	58.45	56.9	764
DCT	82.6	53.8	50.2	256

4.2.3 Selecting the Dimension Reduction Technique

Table 4.4 shows the classification accuracy after applying the commonly used dimension reduction techniques on the feature vector generated after using SPIDER. For LDA, the feature length of the transformed vector would always be c - 1, where c is the total number of classes in the training set. Applying PCA on a vector of dimension d produces d eigenvalues and d eigenvectors each having dimension d. The sum of the eigenvalues is the total energy and the eigenvectors which corresponds to the eigenvalues containing 98% of the energy were used to transform the feature vector to a lower dimension. From the table it can be seen that the accuracy after retaining the first 256 DCT coefficients is close to that

of LDA and PCA but the feature length of DCT is almost 3 times smaller than that of PCA and LDA. Considering, the negligible difference in accuracy but the significant difference in the feature length between the 3 methods, DCT is chosen as the dimension reduction method in our proposed method.

4.2.4 Determining the Number of DCT Co-efficient

Table 4.5: Accuracy Varying the Number of DCT Co-efficient

No of DCT Co-eff	fb(%)	dupI(%)	dupII(%)
64	66.0	23.2	18.7
128	78.4	42.2	40.2
256	82.6	53.8	50.2
512	83.4	57.6	55.6
1024	84.7	62.3	60.5
2048	85.3	64.4	62.2
4096	89.5	66.3	65.2
6400	95.2	72.4	71.1

Table 4.5 shows the classification accuracy while varying the number of DCT coefficients that are retained. Keeping too little coefficients would result in loss of important between-class and within-class discriminatory information and would reduce the accuracy of the system. As more DCT coefficients are retained the accuracy improves but from the classification time point-of-view, increasing the DCT coefficients makes the system computationally expensive. The number of DCT coefficients chosen should be one which reduces the computation time and at the same time gives a relatively accurate classification. From, the table it can be seen that using 256 coefficients satisfies both the conditions. Figure 4.3 shows the graph of accuracy of SPIDER with DCT by taking different number of co-efficients.

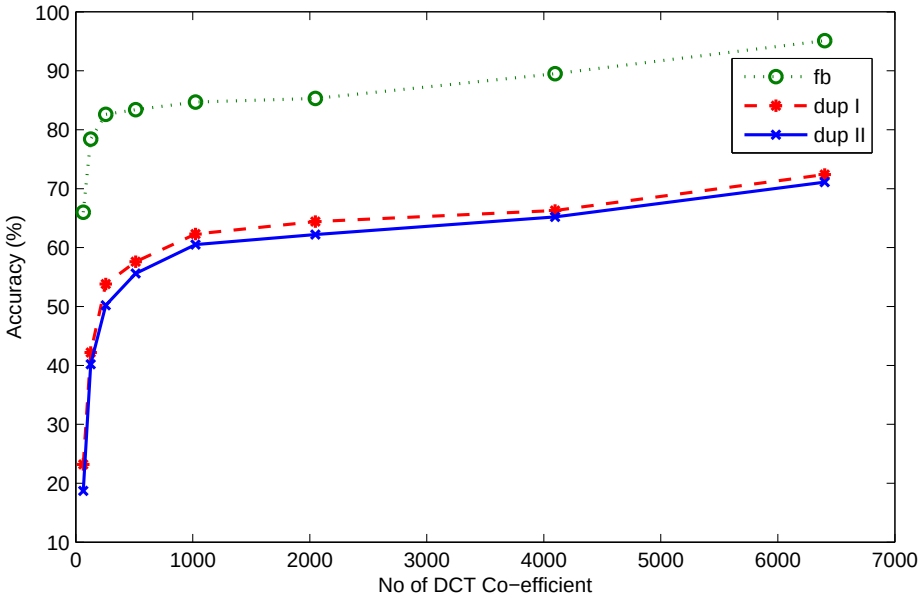

Figure 4.3: Accuracy Varying the Number of DCT Co-efficients

4.3 Performance Evaluation

4.3.1 Without Dimension Reduction

Table 4.6: Comparison With Original Feature Vector

Method	fb(%)	dupI(%)	dupII(%)
LBP	91.0	66.0	63.6
LDP	92.2	68.1	65.1
POEM	93.8	70.3	69.3
SPIDER	92.5	72.4	71.1

The comparison of SPIDER with other existing method is shown in Table 4.6. SPIDER outperforms LBP on all the probe sets, Fb, DupI and DupII because LBP uses the intensity values to generate the bit pattern and intensity values would give varying pattern in presence of noise and illumination variation. The use of directional edge response makes SPIDER more stable in noise and lighting changes. The performance of SPIDER is also comparatively better than LDP

and POEM. While LDP uses a static threshold value, SPIDER uses the dynamic averaging method for thresholding and thus generates stable codes. POEM uses the gradient information only in the x-axis and y-axis and misses any critical informations which might exists in some other direction. SPIDER uses edge response in 8 directions and thus encodes more directional information than POEM patterns. This gives better result when compared to the other methods specially in the challenging probe sets of the FERET database namely, DupI and DupII.

4.3.2 With Dimension Reduction

Table 4.7: Comparison With Reduced Feature Vector

Method	fb(%)	dupI(%)	dupII(%)
LBP	72.1	44.8	41.4
LDP	76.9	48.3	47.5
POEM	79.8	50.9	48.4
SPIDER	82.6	53.8	50.2

Since the pattern encoded by SPIDER has characteristically more information when compared to those generated by the existing methods, applying a dimension reduction method, DCT, on all the feature vector provides more discriminatory information than the other methods. This is shown, in Table 4.7 . After applying DCT on all the existing techniques and selecting the first 256 coefficient, the performance of SPIDER is still relatively better than the existing ones. This shows SPIDER is more compatible with dimension reduction and would perform well in real time applications.

4.3.3 Time Comparison

Table 4.8 shows the classification time and accuracy of SPIDER on Fb probe set while varying the number of DCT coefficients selected. Using fewer coefficients reduces time required for labeling all the images but the accuracy also falls. In-

Table 4.8: Time Comparison Varying The Number Of DCT Co-efficient

No of DCT Co-eff	Classification Time (seconds)	Accuracy (%)
64	3.196	66.0
128	4.887	78.4
256	7.323	82.36
512	12.183	83.4
1024	24.472	84.5
2048	45.558	85.3
4096	70.389	89.5
6400	93.328	95.0

creasing DCT coefficients increases accuracy but computation time also increases. From the table it can be deduced that selecting 256 DCT coefficients provides a good balance between classification accuracy and time.

Chapter 5

Conclusion and Future Plan

5.1 Research Summary

The thesis proposes a new face descriptor built using the SPIDER codes which is robust in presence of noise, aging, illumination variation and expression variation. The descriptor uses 8 directional edge response of the face image and defines a cell around each pixel and accumulates the directional edge along each direction in the cell and computes a local histogram. Next, the local histogram of each cell is compared with the local histogram of the 8 neighboring cells and computes a histogram dissimilarity value. The average dissimilarity is used as threshold and the neighbor cells are assigned a bit of 1 if the dissimilarity value is more than the threshold and a value of 0 otherwise. This produces a 8 bit binary code which is assigned to the center pixel of the center cell. The use of blocks encode the local texture as well as a more global structure present in each face regions.

As a further processing step an existing dimension feature reduction method, DCT, is used to reduce the feature length and make the proposed mechanism compatible for use in face recognition systems where quick classification is of utmost importance. In cases where high computational power is available and classification can be allowed to take a little time but accuracy must be very high, the dimension reduction step can be skipped and the entire feature representation, generated by the face descriptor, can be used for face recognition.

5.2 Future Works

In our proposed method a 8 bit binary pattern is generated for each pixel. The Robinson mask has a property that it generates responses having the same magnitude in pairwise opposite directions. Exploiting this property one of our future work will be to compress the number of bits used to generate SPIDER codes.

Till now we have only worked with still images and in future we plan to apply the proposed face descriptor for recognition of faces taken from video input (a stream of images).

Bibliography

[1] M. C. Hwang, L. T. Ha, N. H. Kim and C. S. Park, "Person identification system for future digital TV with intelligence", *IEEE Trans. on Consumer Electronic*, vol. 53, no. 1, pp. 218-226, 2007.

[2] F. Zuo and P. H. N. de With, "Real-time embedded face recognition for smart home", *IEEE Trans. on Consumer Electronics*, vol. 51, no. 1, pp. 183-190, 2005

[3] T. Jabid, M. H. Kabir, and O. Chae, "Local Directional Pattern (LDP) for Face Recognition," *International Journal of Innovative Computing, Information and Control*, vol. 8, pp. 2423-2437, 2012.

[4] X. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions," in *Proc. IEEE International Workshop on Analysis and Modeling of Faces andGestures (AMFG'07)*, 2007, pp. 168-182.

[5] A. Samal and P. A. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," *Pattern Recognition*, vol. 25, pp. 6577, 1992.

[6] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 10421052, 1993.

[7] M. A. Turk and A. P. Pentland, "Face Recognition using Eigenfaces," in *Proc International Conferencence on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[8] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, pp. 1450-1465, 2002.

[9] K. Etemad, and R. Chellappa, "Discriminant Analysis for recognition of human face images," *Journal of the optical Society of America*, vol. 14, pp. 1724-1733, 1997.

[10] J. Yang, D. Zhang, A. F. Frangi and J. Y. Yang, "Two-Dimensional PCA: A new approach to appearance-based face representation and recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004.

[11] L. Wiskott, J.M. Fellous, N. Krger, and C. von der Malsburg, "Face Recognition by Elastic Bunch Graph Matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, issue 7, pp. 775-779, 1997.

[12] P. Penav, and J. Attick, "Local feature analysis: A general statistical theory for object representation", *Network: Computation in Neural Systems*, vol. 7, pp. 477500, 1996.

[13] M. Lades, J.C. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. Wurtz, and W. Konen, "Distortion Invariant Object Recognition in the Dynamic Link Architecture," *IEEE Transactions on Computers*, vol. 42, issue 3, pp. 300-311, 1993.

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

[15] X.Tan and B.Triggs, "Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions," *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, LNCS 4778, pp.168-182, 2007.

[16] A. Hafiana, G. Seetharaman, and B. Zavidovique, "Median Binary Pattern for Texture Classification," *IEEE International Conference.*

[17] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET Database and Evaluation Procedure for Face Recognition Algorithms," *Image and Vision Computing*, vol. 16, no. 10, pp. 295-306, 1998.

[18] T. Ojala, and M. Pietikainen, "Multiresoulation Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[19] S. Zhao, Y. Gao, and B. Zhang, "Sobel-LBP", *International Conference on Image Processing*, 2008

[20] F. Ahmed and M. H. Kabir, Directional Ternary Pattern (DTP) for Facial Expression Recognition *International Conference on Image Processing*, 2011.

[21] Ngoc-Son Vu and Alice Caplier , "Face Recognition with Patterns of Oriented Edge Magnitudes", *Computer Vision ECCV* , Volume 6311, pg 313-326, 2010

[22] Al-Amin Bhuiyan, and Chang Hong Liu, "On Face Recognition using Gabor Filters," *World Academy of Science and Technology*, no. 28, pp. 51-56, 2007.

[23] Z. M. Hafed and M. D. Levine, "Face Recognition using Discrete Cosine Transform", *International Journal of Computer Vision*, vol. 43, no. 3, pp. 167-188, 2001.

[24] Hua Yu and Jie Yang, "A direct LDA algorithm for high-dimensional data - with application to face recognition", *The journal of pattern recognition society*, no. 34 pp. 2067 - 2070, october 2001.

[25] Kumar A., "Analysis of Unsupervised Dimensionality Reduction Techniques", *Computer Science and Information Systems*, vol. 6, no. 2, pp. 217-227, 2009.

[26] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfield, "Face Recognition: A literature survey", *ACM Computing Surveys*, vol. 35, no. 4, pp. 399-458, 2003.

[27] R. C. Gonzalez, and R. E. Woods, *"Digital Image Processing"*, 3rd Ed, Pearson Education, 2009.

Chapter 6

Appendix

6.1 SPIDER Descriptor Generation Code

```
fin = fopen('name_dupI.txt','r');
fid = fopen('spider_dupI.txt' , 'wt' );

for images = 1:500
    no = 0;
    maxd = -1;
    a = fscanf(fin, '%s',1);
    b = a(1:5);
    c = 0;
    for i=1:5
        c = c + ((b(i)-'0')*(10.^(5-i)));
    end
    fprintf(fid, '%d ',c);
    img = imread(a);
    img = imresize( img, [105 105 ]);
    [ r c k ] = size( img ) ;
    var2 = rgb2gray( img ) ;
    var2 = double(var2);
    [r c] = size(var2);
    dx = [ -1, -1, 0, 1, 1, 1, 0, -1, 0 ];
    dy = [ 0, -1, -1, -1, 0, 1, 1, 1, 0 ];
    rmask = zeros( 3,3,8 );

```

```
24      %% Robinson Mask
25      rmask( :,:,1 ) = [ -1 -2 -1; 0 0 0; 1 2 1 ];
26      rmask( :,:,2 ) = [ -2 -1 0; -1 0 1; 0 1 2 ];
27      rmask( :,:,3 ) = [ 1 0 -1; 2 0 -2; 1 0 -1 ];
28      rmask( :,:,4 ) = [ 0 -1 -2; 1 0 -1; 2 1 0 ];
29      rmask( :,:,5 ) = [ 1 2 1; 0 0 0; -1 -2 -1 ];
30      rmask( :,:,6 ) = [ 0 1 2; -1 0 1; -2 -1 0 ];
31      rmask( :,:,7 ) = [ -1 0 1; -2 0 2; -1 0 1 ];
32      rmask( :,:,8 ) = [ -2 -1 0; -1 0 1; 0 1 2 ];
33      edgeResponse = zeros( r,c,8 ) ;
34
35      %% Calculating Edge Response Of The Image
36      for i = 2:r - 1
37          for j = 2:c - 1
38              for k = 1:8
39                  responseSum = 0;
40                  for l = 1:8
41                     nr = i + dx( l );
42                     nc = j + dy( l );
43                     responseSum = responseSum + ( var2( nr, nc ) ...
                           * rmask( 2 + dx( l ) , 2 + dy( l ) , k ) ) ;
44                  end
45                  if( responseSum < 0 )
46                      responseSum = -responseSum ;
47                  end
48                  edgeResponse( i,j,k ) = responseSum;
49              end
50          end
51      end
52      bin = zeros( r, c, 8);
53      poemImg = zeros( r,c );
54
55      %% Accumulating The Edge Response In Each Cell
56      for i = 3: r - 2
57          for j = 3: c - 2
58              for l = 1:8
```

```
59                  for k = 1:9
60                     nr = i + dx( k );
61                     nc = j + dy( k ) ;
62                     if( nr >= 2 && nr <= r-1 && nc >= 2 && nc <= ...
                           c-1 )
63                        bin( i,j,l ) = bin( i,j,l ) + ...
                              edgeResponse( nr,nc,l);
64                     end
65                  end
66
67              end
68          end
69      end
70      dxx = [ -3, -3, 0, 3, 3, 3, 0, -3 ];
71      dyy = [ 0, -3, -3, -3, 0, 3, 3, 3 ];
72
73       cnt = 0;
74       hist = zeros(256,8);
75       a = 0.0;
76       cr = 0;
77       rd = zeros( 9025   , 1 );
78       tot = 0.0;
79
80       %% Computing Dissimilarity Measure
81
82       for i = 6: r - 5
83          for j = 6: c - 5
84              dis = zeros(8,1);
85              temp = zeros( 8,1 );
86              for k = 1:8
87                  disSimilarity = 0.0;
88                  nr = i + dxx( k );
89                  nc = j + dyy( k ) ;
90                  if( nr >= 3 && nr <= r-2 && nc >= 3 && nc <= c-2 )
91                      for p = 1:8
92                          num = bin( i,j,p ) - bin( nr, nc, p ) ;
```

```
93                          num = num * num;
94                          den = bin( i,j,p ) + bin( nr, nc, p ) ;
95                          if( den > 0 )
96                            disSimilarity = disSimilarity + ( num ...
                                  / den ) ;
97                          end
98                      end
99                  end
100                 dis (k,1) = disSimilarity;
101             end
102             mx = 0;
103             mn = 1000000000;
104             for p = 1:8
105                 if( dis(p,1) > mx ) mx = dis( p,1 );
106                 end
107                 if( dis(p,1) < mn ) mn = dis( p,1 );
108                 end
109             end
110             avg = 0;
111             %% Calculating Average Dissimilarity
112             for p = 1:8
113                 avg = avg + dis(p,1);
114             end
115             avg = avg / 8;
116             stemp = sort( dis );
117             s = 0;
118             temp = zeros( 8,1);
119
120             %% Generating Bit Pattern
121             for z = 1:8
122                 if( dis(z,1) > avg  )
123                    temp( z,1 ) = 1;
124                 end
125             end
126
```

```
127             %% Converting The 8-bit Code To Corresponding ...
                   Integer Value
128             s = 0;
129             for z = 1:8
130                 if( temp( z,1 ) == 1 )
131                     s = s + ( ( 2 .^ ( z - 1 ) )  * 1 ) ;
132                 end
133             end
134             poemImg(i,j) = s;
135
136             cnt = cnt + 1;
137         end
138      end
139
140     %% Subdividing The Image To Generate The Feature Vector
141     localWindow = 5;
142     ratio = r / localWindow;
143
144     for l = 1:localWindow
145         sr = ( ratio * ( l - 1 ) ) + 1;
146         er = ratio * l ;
147         for col = 1:localWindow
148             sc = ( ratio * ( col - 1 ) ) + 1;
149             ec = ratio * col ;
150             his = zeros( 256,1 );
151             cnt = 0;
152             hist = zeros(256,8);
153             hist2 = hist;
154             for i = sr:er
155                 for j = sc:ec
156                     hist(poemImg(i,j) + 1 ,1) = ...
                           hist(poemImg(i,j) + 1 ,1) + 1;
157                 end
158             end
159              for i=1:256
160                  no = no + 1;
```

```
161                     fprintf(fid,'%d ',hist(i,1));
162                 end
163             end
164         end
165         fprintf(fid, '\n');
166 end
```