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

Human Iris Verification

M.Tech(Computer Science) Dissertation Report

By

Suvadip Mukherjee

Roll Number : CS0911

Under the supervision of

Prof. Bhabatosh Chanda

Electronics and Communication Sciences Unit

Indian Statistical Institute

203, B.T. Road

Kolkata 700 108

Abstract

This article proposes a method for personal identification based on iris recognition. The method consists of three major components: image preprocessing, feature extraction and classifier design. The UBIRIS database is used for obtaining iris images. The iris segmentation is obtained by using an integro-differential operation.The segmented iris is then normalised and a small portion of the normalised portion is used for feature extraction and verification.Three types of features are extracted from the normalised iris segments - GLCM based features, features based on number of runs of pixels in four directions (N,NE,E,NW) and features extracted using Local Directional Pattern or LDP.We present a comparison of the performances of the method using a combination of the above mentioned feature extraction techniques.It has also been shown experimentally that the iris patterns exhibit a symmetry about the vertical axis. The multiclass problem is reduced to a two class verification problem. Two types of feature vectors - interclass difference vectors and intraclass difference vectors, thus created, are trained on a Support Vector Machines for classification. Experimental results show that the proposed method has encouraging performance.

Keywords Iris Verification, LDP, GLCM, biometrics

Contents

1 Introduction **2**
- 1.1 Background Concepts . . . 3
  - 1.1.1 Iris Anatomy . . . 3
  - 1.1.2 Performance Measurement . . . 3
- 1.2 Commonly used datasets of Iris images . . . 4
- 1.3 Overview of the Iris Recognition/Verification algorithms . . . 4

2 Iris Verification System **7**
- 2.1 Iris Segmentaion . . . 7
  - 2.1.1 Symmetry of the iris pattern . . . 7
  - 2.1.2 Segmentation of the pupil . . . 8
  - 2.1.3 Segmentation of the Iris . . . 10
- 2.2 Iris Normalisation . . . 12
- 2.3 Feature Extraction . . . 14
  - 2.3.1 GLCM based features . . . 14
  - 2.3.2 Edge based features . . . 16
  - 2.3.3 LBP/LDP based feature . . . 16
- 2.4 Verification/Authentication . . . 19
  - 2.4.1 Reduction to two class problem . . . 19
  - 2.4.2 Training and Classification . . . 19
  - 2.4.3 Support Vector Machine . . . 20

3 Experimental Results **23**
- 3.1 Iris Verification Output . . . 24
  - 3.1.1 Scope of further work . . . 25
- 3.2 Conclusion . . . 26

Chapter 1

Introduction

Today, biometric recognition is a common and reliable way to authenticate the identity of a living person based on physiological or behavioral characteristics. A physiological characteristic is relatively stable physical characteristic, such as fingerprint, iris pattern, facial feature, hand silhouette, etc. This kind of measurement is basically unchanging and unalterable without significant duress. A behavioral characteristic is more a reflection of an individuals psychological makeup as signature,speech pattern, or how one types at a keyboard.The degree of intra-personal variation in a physical characteristic is smaller than a behavioral characteristic. For examples, a signature is influenced by both controllable actions and less psychological factors, and speech pattern is influenced by current emotional state, whereas fingerprint template is independent. Nevertheless all physiology-based biometrics dont offer satisfactory recognition rates The automated personal identity authentication systems based on iris recognition are reputed to be the most reliable among all biometric methods: it is claimed that the probability of finding two persons with the same iris pattern is almost zero! Hence it is no surprise that iris recognition techniques have become an important biometric technique.Compared to fingerprint, iris is protected from the external environment behind the cornea and the eyelid. Not subject to deleterious effects of aging, the small-scale radial features of the iris remain stable and fixed from about one year of age throughout life.

1.1 Background Concepts

Here we briefly review some basic concepts before we move on to the more specific aspects of iris verification.

1.1.1 Iris Anatomy

The iris is the colored ring of tissue around the pupil through which light enters the interior of the eye.Two muscles, the dilator and the sphincter muscles, control the size of the iris to adjust the amount of light entering the pupil. Figure 1.1 shows an image from the UBIRIS database. The sclera, a white region of connective tissue and blood vessels, surrounds the iris. A clear covering called the cornea covers the iris and the pupil. The pupil region generally appears darker than the iris. However, the pupil may have specular highlights, and cataracts can lighten the pupil. The iris typically has a rich pattern of furrows, ridges,and pigment spots.

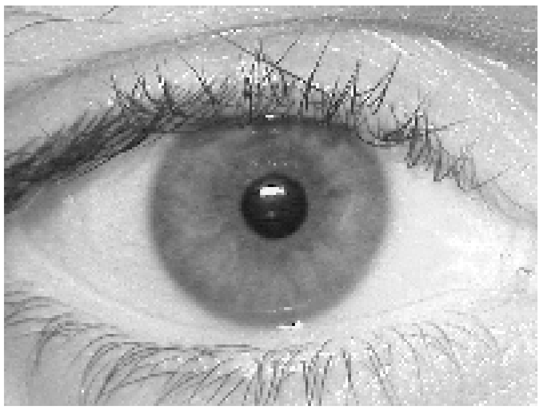


Figure 1.1: Human Iris

1.1.2 Performance Measurement

A biometric problem can be viewed from two points of views. First, where one is concerned with the matching of an unknown iris template with a template existing in the database. This kind of problem is often refered to as *Identification Problem*. Clearly, this is a N class problem, where N is the number of people, whose Iris images are present in the database.

This problem can also be converted to a verification problem. Here, a certain Mr.X inputs his iris template, and the algorithm needs to verify whether the input template corresponds to Mr.X or not. Verification is done by matching a biometric sample acquired at the time of the claim against the sample previously enrolled for the claimed identity. If the two samples match well enough, the identity claim is verified, and if the two samples do not match well enough, the claim is rejected. Thus there are four possible outcomes. A *true accept* occurs when the system accepts, or verifies, an identity claim, and the claim is true. A *false accept* occurs when the system accepts an identity claim, but the claim is not true. A *true reject* occurs when the system rejects an identity claim and the claim is false. A *false reject* occurs when the system rejects an identity claim,but the claim is true.
We have dealt with iris verification problem, due to some merits it enjoys over the identification model.

1.2 Commonly used datasets of Iris images

Experimental research on segmentation, texture encoding, and matching requires an iris image dataset.One issue deserves a brief mention at this point. The first iris image dataset to be widely used by the research community was the CASIA version 1 dataset. Unfortunately, this dataset had the feature that the pupil area in each image had been replaced with a circular region of constant intensity to mask out the specular reflections from the NIR(near-infrared) illuminators. This feature of the dataset naturally calls into question any results obtained using it, as the iris segmentation has been made artificially easy. In this project, we have used the UBIRIS $V1.0$ database [9].

1.3 Overview of the Iris Recognition/Verification algorithms

An Iris Verification algorithm can be broadly subdivided into four modules:

1. Iris Segmentation
2. Iris Normalisation
3. Feature Extraction

4. Matching or Verification

Iris Segmentation In the segmentation stage, an image of the human eye is input to the algorithm and the desired output is the segmented iris portion. This stage is supposedly the most difficult part of the algorithm. This is because, there exists a number of images in the database, where the iris is not totally visible. Sometimes the iris is partially or totally occluded by the uppper eyelid. Segmentation results are also affected due to the presence of eyelashes and presence of cataracts and contact lenses. There are a few

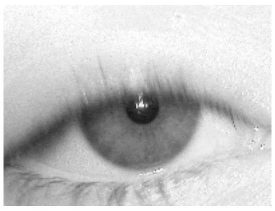


Figure 1.2: Occlusion by eyelid

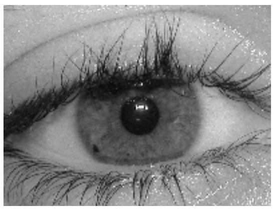


Figure 1.3: Occlusion by eyelash

methods that have been proposed to obtain good segmentation. In fact, segmentation in presence of eyelashes or contact lenses is still considered to be a major challenge.A properly segmented iris is shown in Figure 1.4

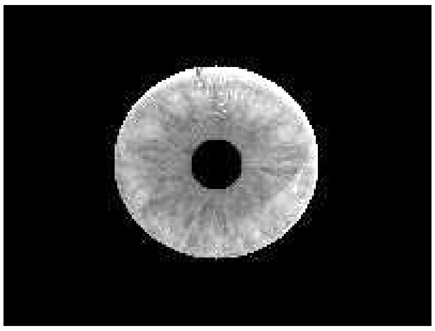


Figure 1.4: After Segmentation

Iris Normalization This step is optional, but normalisation of the iris often helps in computation. This is because, the size and shape of the annular iris region varies from one individual to the other. Hence, a normalised rectangular block is desirable for extracting features from the Iris. The annular iris region is converted to a rectangular region by a simple change of coordinate system. Figure 1.5 shows a normalised iris portion.

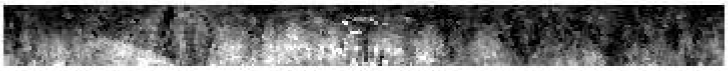


Figure 1.5: After Normalization

Feature Extraction Once the normalised iris block is obtained,some features need to be extracted, so that two iris blocks can be distinguished.Looking at different approaches to analysing the texture of the iris has perhaps been the most popular area of research in iris biometrics. John Daughman used Gabor filters to extract textural features from iris images[4]. One body of work effectively looks at using something other than a Gabor filter to produce a binary representation similar to Daugmans iris code. Another body of work looks at using different types of filters to represent the iris texture with a real-valued feature vector.

Matching/Verification Once the feature vector has been obtained, a classifier needs to be designed for matching of an Iris template with one in the database. As explained before, the problem can be treated as a multiclass problem or an identification problem.It can also be reduced to a two class verification problem. One advantage that the later method enjoys over the former is that there is no need to train the model once a new set of images gets added to the database. This implies lesser time complexity, since training of a model is a time consuming affair.

Chapter 2

Iris Verification System

In this chapter, the implemented iris verification algorithm is discussed.

2.1 Iris Segmentaion

A close observation of the database images reveal some generic information about the shape of the iris and the pupil.The pupil can be approximately considered to be a spherical region.This is a safe approximation, given the fact that the pupilary area is quite small.However, one needs to be more careful to model the shape of the iris boundaries. Our experiments show that a few images result in improper segmentaion if the iris boundary is considered to be a circle. We have hence modelled the iris boundary as an elliptical surface and modified $Daughman's\ integro\ differential\ operator$ [4] to segment the iris portion accordingly. In order to segment the iris from the given eye image, one needs to determine two parameters initially: the centre coordinates of the pupil and the centre coordinates of the iris. But before we move on to the segmentation portion, one interesting observation regarding symmetry in iris pattern deserves a mention.

2.1.1 Symmetry of the iris pattern

In this subsection, our aim is to find a rotational symmetry in iris patterns. For this purpose, we have hand selected a few iris images (belonging to different individuals) where the entire iris portion is available for segmentation,i.e.,there is no occlusion. Here we have correlated the segmented iris image(Fig 2.1) with the same image, but rotated in the counter clockwise direction by an angle θ. The correlation coeffecient(ρ) (Y-axis) obtained is plotted against θ (X-axis) (Fig 2.2)

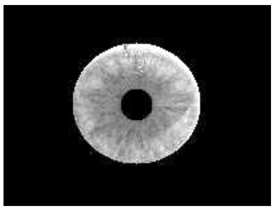


Figure 2.1: Original Image

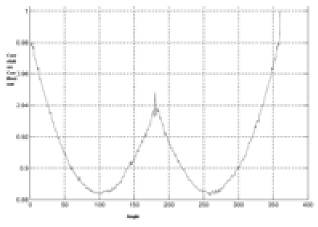


Figure 2.2: Correlation Vs Angle

We observe that the correlation coefficient reaches a peak around 180 degree, indicating that there exists a symmetry of the iris patterns about the vertical axis. Hence, during feature extraction, it would suffice to work with only one portion of the iris (either left or right).This is supported by the results obtained by using only the left portion of the iris template.This observation is significant due to the fact that using only one portion of the iris template reduces the computational time significantly,although the accurcay is not compromised. Also, in order to avoid occluded portions, we have discarded the top and bottom portion of the iris during segmentation(Fig 2.6).

2.1.2 Segmentation of the pupil

Our algorithm to segment the pupil is based on a particular assumption that the pupil is darker than any other region in the eye image.The algorithm to search for the pupil centre can be subdivided into two sub parts:

1. A coarse search for the pupil centre
2. A finer search for the pupil centre, using the output of the coarse search

Coarse Search The method for ascertaining the center and the radius is to search all pixels that belong to the pupil in the image and store the X and Y coordinates of each pixel[5]. According to the formula for area of a circle, $A = \pi r^2$ we derive the pupil's radius, where A denotes the number of pixels in the pupil, and we derive the pupil's center coordinates by computing the mean of all these pixels' X and Y coordinates. The method for determining those pixels belonging to the pupil is based on the fact that the pupil is blacker than other portions of the image. We can easily find that the pupil's white is smaller than other areas and the color is almost the same for each pixel in the pupil. The gray change between the pixels is very small. So we can judge whether a point is in the pupil according to its gray level. Set the threshold of gray to be

$$L = \frac{DC}{5 - 2 * (DC - 128)/128}$$

The above formula is emperical and has been verified experimentally. Let $f(I(x, y))$ denote the sum of absolute differences of a pixel $f(x, y)$ from its $8 - neighbours.$Matematically, it can be written as

$$f(I(x, y)) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} |f(x, y) - f(x - i, y - j)|$$

The average difference between the grey levels of two pixels is given by $\bar{f}(I)$

We claim that a pixel I(x,y) belongs to the pupil region if both the following conditions hold

1. $I(x, y) < L$
2. $f(I(x, y)) \; < \; \Delta L, \; where \; \Delta L = \bar{f}(I)/2$

We can now find the coarse pupil centre and radii in the following manner If (x_c, y_c) denote the pupil centre coordinates and r_p denote the pupil radius obtained from this coarse search,

$$x_c = mean(x_i) \; and \; y_c = mean(y_i) \; \forall i \; such \; that \; (x_i, y_i) \; is \; a \; pupil \; pixel$$

$$r_p = \sqrt{(}A/\pi) \; where \; A \; is \; the \; sum \; of \; greyvalues \; of \; all \; pupil \; pixels$$

In the UBIRIS database, there is a specular reflection in the pupilary region for most of the images. Due to this, the pupil no longer remains a homogeneously dark portion. Hence, we use a smoothing technique that darkens any patch of non homogeneous bright spots.

Finer Search Once r_p and (x_p, y_p) is obtained from the above mentioned coarse search mechanism, we proceed to find the actual pupil centre and actual pupil radii. To achieve this, we consider a $M \times N$ window (W)centred around (x_p, y_p) such that each pixel in that window is a probable pupil centre. For each pixel $(x_i, y_i) \, \epsilon \, W$, we apply the following integro-differential operator.

$$(r, x_c, y_c) = \max_{(r,x_i,y_i)} |G_\sigma(r) * \frac{\partial}{\partial r} \oint \frac{I(x,y)}{2\pi r} ds| \tag{2.1}$$

The above equation can be interpreted in the following manner: centered at a possible pupil centre (x_0, y_0) , we can draw a circle of radii r. Another concentric circle of radii $r + dr$ can also be drawn. Now, the change in intensity of greylevel is maximum at the boundary of the pupil and the iris. Hence, in the above equation, we look for that particular (r, x_0, y_0) such that the right hand side is maximised. The convolution of the image with the Gaussian produces a smoothing effect. It is to be noted that during implementation, a minimum and maximum value of the iris radii has to be informed for better performance.It has been observed that $r_{min} = 10$ and $r_{max} = 20$ accurately segments the pupilary region of a $(200 * 150)$ image from our database.Figure 2.3 and Figure 2.4 show the segmentation results obtained by using the above method.

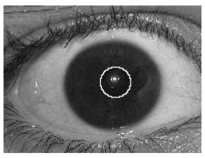


Figure 2.3: Segmented Pupil

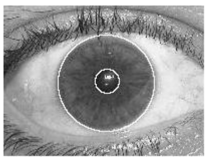


Figure 2.4: Segmented Pupil and Iris

2.1.3 Segmentation of the Iris

From the segmented pupil, we obtain two information: pupil radius:r and pupil centre: (x_p, y_p). In order to obtain the iris parameters from the above data, we need to find

1. Minimum and maximum iris radii

2. Iris centre (which is not always the same as the pupil centre)

Minimum and Maximum Iris Radius if r be the pupil radius, and R_{min} and R_{max} be the minimum and maximum iris radius respectively, then we obtain a certain result through experiments and it is observed that choosing $R_{min} = 2r$ and $R_{max} = 4.2r$ provides accurate results in most of the images.

Iris Centre Although the iris and the pupil appears to be concentric, in practice there exists a slight shift between the two centres. Thus, to detect the iris centre, we use a 9×9 window centred around (x_p, y_p) and apply Daughman's integro-differential operation at each pixel of the window . The segmentation procedure is similar to that of pupil segmentation, with the only exception that the equation is modified slightly to model the iris boundary as an elliptic curve. Also, in order to avoid eyelid and eyelash occlusion, the integration is not conducted from 0 to 2π radian. Rather, we only consider the portion between $45\deg - 135\deg$ and $225\deg - 315\deg$. Figure 2.5 and Figure 2.6 show the segmented iris obtained by the above algorithm.

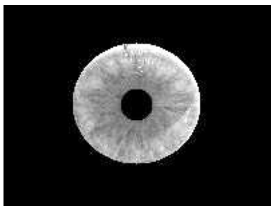


Figure 2.5: Segmented Iris without removal of any portion

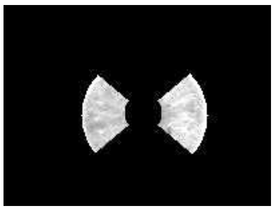


Figure 2.6: Removing portions to avoid occlusion

2.2 Iris Normalisation

After segmentation is complete, we get an image as shown in Figure 2.6.It is important to note that we have discarded certain portions of the circular iris, in order to avoid occlusion.This step is often necessary, since almost in all the images we encounter an occluded iris. We could have extracted features from the segmented iris image(Figure 2.6). However, there are a few disadvantages with this approach:

- The size of the non-black(iris) portion is different for different images. Hence,there arises a need for some sort of normalisation.
- Even for irises from the same eye, the size may change due to illumination variations and other factors.

Such elastic deformation in iris texture will affect the results of iris matching. For the purpose of achieving more accurate recognition results, it is necessary to compensate for the iris deformation. Hence, we have included this step of iris normalisation.This problem can be solved by projecting the original iris in a cartesian coordinate system into a doubly dimensionless pseudopolar coordinate. [4],[8] . In our algorithm, the annular iris portion obtained after segmentation is transformed into a 30×360 rectangular block. The $\theta(\theta \in [0; 2\pi])$ parameter and dimensionless ρ $(\rho \in [0; 1])$ parameter describe the polar coordinate system. Thus the following equations implement $I(x(\rho, \theta), y(\rho, \theta)) \rightarrow I(\rho, \theta)$:

$$x(\rho, \theta) = (1 - \rho)x_p(\theta) + \rho x_i(\theta) \tag{2.2}$$

$$y(\rho, \theta) = (1 - \rho)y_p(\theta) + \rho y_i(\theta) \tag{2.3}$$

Where

$$x_p(\theta) - x_{p0}(\theta) = r_p \cos(\theta) \tag{2.4}$$

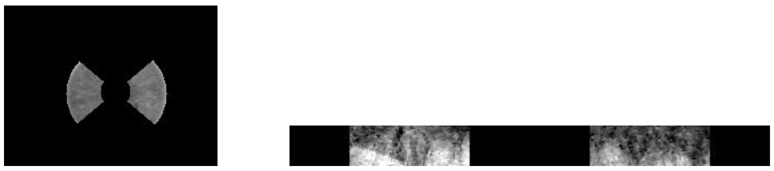
$$y_p(\theta) - y_{p0}(\theta) = r_p \sin(\theta) \tag{2.5}$$

and

$$x_i(\theta) - x_{i0}(\theta) = r_i \cos(\theta) \tag{2.6}$$

$$y_i(\theta) - y_{i0}(\theta) = r_i \sin(\theta) \tag{2.7}$$

Where r_p and r_i are respectively the radius of the pupil and the iris, while $(x_p(\theta), y_p(\theta))$ and $(x_i(\theta), y_i(\theta))$ are the coordinates of the pupillary and limbic boundaries in the direction $\theta.(x_{i0}, y_{i0})$ and $((x_{p0}, y_{p0}))$ are the Iris and Pupil centres respectively. The frontier zones (iris/pupil and iris/sclera) are truncated to avoid noising the iris rectangular representation by other patterns not included in the iris texture. The resulting image is shown in Fig 2.7. All the iris images are now converted to this 30×360 size block which is normalized in terms of size.



Before Normalization *After Normalization*

Figure 2.7: Normalization Results

The normalized image is contrast stretched for better visibility of the iris texture.

2.3 Feature Extraction

After normalisation of the annular iris portion, all the operations(feature selection etc) are performed on the normalised block itself.Owing to the texture rich characteristics, the features selected are mostly textural. Our method includes the following types of features that have been extracted from the image. The proformance of our algorithm on these features are illustrated in the next chapter. The various features used in this method are as follows

- GLCM or Gray Level Cooccurance Matrix based features
- Feature based on Edge Direction
- Local Binary Pattern(LBP) based features
- Local Directional Pattern (LDP) based features

In our algorithm we have analysed the performance using a combination of the above features. In this section, we describe the methods of feature extraction.

2.3.1 GLCM based features

The GLCM or Gray Level Cooccurance Matrix is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image.It was developed by Haralick in the 1970's. Mathematically, a co-occurrence matrix C is defined over an $n \times m$ image I, parameterized by an offset $(\delta x, \delta y)$, as:

$$\begin{aligned} C_{\delta x,\delta y}(i,j) &= \textstyle\sum_{p=1}^{n}\sum_{q=1}^{m} 1 \; if \; I(p,q) = i \; and \; I(p+\delta p, q+\delta q) = j \\ &= 0 \; otherwise \end{aligned}$$

Note that the $(\delta x, \delta y)$ parameterization makes the co-occurrence matrix sensitive to rotation. We choose one offset vector, so a rotation of the image not equal to 180 degrees will result in a different co-occurrence distribution for the same (rotated) image. This is rarely desirable in the applications co-occurrence matrices are used in, so the co-occurrence matrix is often formed using a set of offsets sweeping through 180 degrees (i.e. 0, 45, 90, and 135 degrees) at the same distance to achieve a degree of rotational invariance. From this co-occurance matrix, we calculate the following features

1. Energy = $\sum_{i,j} C(i,j)^2$

2. Contrast $= \sum_{i,j} |i-j|^k C(i,j)^l$
3. Correlation $= \sum_{i,j} \frac{(i-\mu)(j-\mu)C(i,j)}{\sigma^2}$
4. Homogeneity $= \sum_{i,j} \frac{C(i,j)}{1+|i-j|}$

where $\mu = \sum_{i,j} iC(i,j)$. Refer to [10] for further discussion on GLCM.

Steps to find GLCM based features

1. Divide the image into 30×30 overlapping sub-blocks $B_1, ...B_n$.(Fig 2.8)
2. For each such sub-block $B_i, i = 1, \ldots, n$ find the four features $f_i = (f_{i1}, \ldots, f_{i4})^T, i = 1, \ldots, n$
3. The final feature vector F $= (f_i, \ldots, f_n)^T$. Hence we obtain a feature vector of size $4 \times n$

where n is the number of sub-blocks . Note that we can also do the following

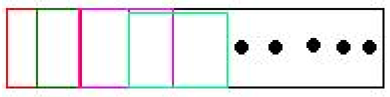


Figure 2.8: Dividing into Overlapping Sub-Blocks

for further precision :

- Find four GLCM for a block, one for each direction (0,45,90 and 135 degree) and the resulting GLCM is the average of these four matrices.
- We also consider GLCM s for distance = 1,2 and 3 units. This makes the feature vector size as $3 \times (4n)$ or $12 \times n$

Despite this slight increase in size of the feature vector,the classifier performance is largely improved by using the GLCMs for three distance values.

2.3.2 Edge based features

Edges in the normalised image give us an information about some structural features of the image. We are interested to find the number of runs of edges in a particular direction(say 0 degree, 45 degree, 135 degree and 90 degree). Before we go into the algorithm, we discuss how to find out the number of runs of black pixel in a particular direction for a binary image.

$$M_0 = \begin{pmatrix} c & 1 \end{pmatrix}, M_{90} = \begin{pmatrix} c \\ 1 \end{pmatrix}, M_{45} = \begin{pmatrix} c & 0 \\ 0 & 1 \end{pmatrix}, M_{135} = \begin{pmatrix} 0 & 1 \\ c & 0 \end{pmatrix}$$

Here c denotes the centre of the structuring element. Now, *Morphological Erosion* of any binary image with M_0 would give the number of runs in the horizontal direction. Similarly, erosion with M_{90}, M_{45}, M_{135} gives the runs in the other three directions. Refer to [10],[7] for more details on Mathematical Morphology.

Steps to find edge based features

1. Divide the image into n overlapping sub-blocks as before
2. Convert the image into binary by applying Canny's edge detector [2].
3. Find the runs in the four directions by using *morphological erosion.* Let they be $R_i = (r_1, \ldots, r_4), i = 1, \ldots, n$.
4. Final Feature $F = (R_1, \ldots, R_n)^T$

2.3.3 LBP/LDP based feature

Local Binary Pattern or LBP has been used in texture analysis in the recent years[11].However, LBP suffers some drawbacks like it is sensitive to random noise.Hence we have used another approach for texture analysis called Local Directional Pattern(LDP).LDP has been used for Face Recognition purpose[6]. Our experiments show that LDP also performs impressively in the domain of iris recognition/verification. Here we touch upon the basics of LBP and then discuss our implementation of LDP.

Local Binary Pattern(LBP) The LBP operator, a gray-scale invariant texture primitive,has gained significant popularity for describing texture of an image [11].It labels each pixel of an image by thresholding its P-neighbor values with the center value and converts the result into a binary number by using (2.8).

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s * (g_p - g_c)^{2^p} \tag{2.8}$$

$$s(x) = 1 \; if \; x \geq 0 \; and \; 0 \; otherwise$$

where g_c denotes the gray value of the center pixel (x_c, y_c) and g_p corresponds to the gray values of **P** equally spaced pixels on the circumference of a circle with radius **R**.

Local Directional Pattern(LDP) Local Directional Pattern (LDP) is an eight bit binary code assigned to each pixel of an input image. 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 $(M0, \ldots, M7)$ centered on its own position. These masks are shown in Figure 2.9. Applying eight masks, we obtain eight edge response value

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

Figure 2.9: LDP Masks

$(m0, m1, \ldots, m7)$ 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 of $|m_j|$ and set them to 1. The other $(8 - k)$ bit of 8-bit LDP pattern is set to 0.

The LDP code produces more stable pattern in presence of noise. For instance, Fig. 2.10 shows an original image and the corresponding image after adding Gaussian white noise. After addition of noise, 5^{th} bit of LBP changed from 1 to 0, thus LBP pattern changed from uniform to a non-uniform code. Since gradients are more stable than gray value, LDP pattern provides the same pattern value even presence of that noise and non-monotonic illumination changes.

$\begin{pmatrix} 85 & 32 & 26 \\ 53 & 50 & 10 \\ 60 & 38 & 45 \end{pmatrix}$ LBP = 00111000 , LDP = 00010011	$\begin{pmatrix} 81 & 29 & 32 \\ 83 & 58 & 15 \\ 65 & 43 & 47 \end{pmatrix}$ LBP = 00101000 , LDP = 00010011
(a)Original Image	(b)After Addition of noise

Figure 2.10: Stability of LDP vs. LBP

LBP Histogram 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

$$H_i = \sum_{x,y} P(I_L(x, y) = C_i), \quad c_i = i^{th}\ LDP\ pattern(0 \leq i < 56) \qquad (2.9)$$

where $P(A) = 1\ if\ A\ is\ true,\ else\ 0.$

Steps to Compute LDP based features

1. Let I be the non occluded portion of the Iris.
2. Compute the LDP Histogram on I as per (2.9).This histogram acts as a feature vector of dimension 56.

We thus get a 56 dimensional feature vector using this LDP histogram.

2.4 Verification/Authentication

Once we have extracted the feature vectors, only thing that remains is to train a suitable model for proper classification. In this section we describe our method to generate the test set and train a Support Vector Machine for classification.

2.4.1 Reduction to two class problem

Suppose in our database we have eye images of N different individuals. Treating each individual as a different class, the iris recognition problem is a N-class problem, where one has to match an unknown iris template with the images in each of these N classes. However this methodology of matching has a drawback - Once a image of a new individual gets added to the database, the entire model needs to be re-trained.This is indeed a time consuming affair. Hence, another option is to convert this problem to a two-class problem: here the question to be addressed is whether the input iris template (of a certain Mr.X) matches with the claimed template present in the database.

Creating the Distance Vectors Our method of conversion to a 2-class problem is based on a simple intuition. In the UBIRIS database[9] there are 5 images for each individual. The feature vectors extracted from each of these 5 images are supposed to be more similar to each other in comparison to the similarity between the images of two different individuals. This leads us to define two types of distances: *intraclass distance and interclass distance* .
Let there be N classes and each classes contain n number of images.

- Intra Class Distance $(d_{intra}) = |V^i_{mean} - V^j_{mean}| \; \forall \, i, j \in [1, \ldots, N] \; such \; that \; i \neq j$
- Inter Class Distance $(d_{inter}) = |V^k_i - V^k_j| \; \forall \, i, j \in [1, \ldots, n] \; such \; that \; i \neq j$

Where, V^k_i denote the feature vector extracted from the i^{th} image of the k^{th} class, and $V^k_{mean} = \frac{1}{n} \sum_{i=1}^{n} V^k_i$

2.4.2 Training and Classification

The training dataset is consctruted by computing all possible interclass and intraclass distances. We assign a label $l = -1$ for all d_{inter} and $l = +1$ for all d_{intra} vectors.The distance vectors are now input to a two class Support Vector Machine[1]. In this section we present a very brief overview of SVM.

2.4.3 Support Vector Machine

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

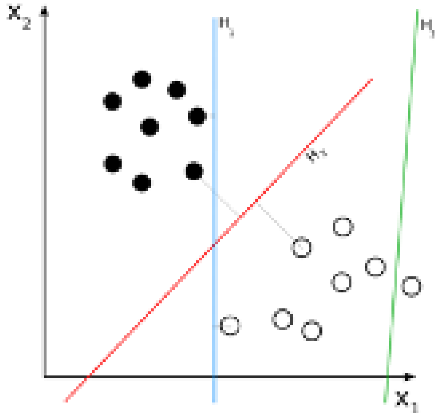


Figure 2.11: H3 doesn't separate the two classes. H1 does, with a small margin and H2 with the maximum margin.

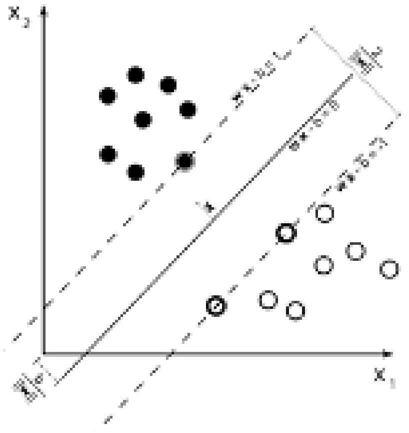


Figure 2.12: Maximum-margin hyperplane and margins for an SVM trained with samples from two classes.

Suppose 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:

$$g(x) = wx + b \tag{2.10}$$

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 vectors to the most superior hyper plane in different classes are called **support vectors (SV)**.

The support vector distance to the origin 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 the following quadratic programming problem :

$$\begin{aligned} \underset{w}{\text{minimize}} \quad & \frac{1}{2}||w||^2 \\ \text{subject to} \quad & y_i(wx_i + b) \geq 1 \; \forall i = 1, \ldots, L \end{aligned} \tag{2.11}$$

The classified function can be obtained through the Lagrange multiplier:

$$f(x, \alpha^\star, b^\star) = \text{sgn}(\sum_{i=1}^{L} y_i \alpha_i{}^\star (x_i x) + b^\star) \tag{2.12}$$

$$c^\star = \max_{y_i=1} w^\star x_i$$

$$d^\star = \min_{y_i=-1} w^\star x_i$$

$$b^\star = \frac{c^\star + d^\star}{2} \tag{2.13}$$

Here x_i is any support vector.
Regarding the situation as non-linear, nuclear kernel function $K(x_i, x)$ was introduced to find the linear classification surface in high dimensional space by the following formula:

$$f(x, \alpha^\star, b^\star) = \text{sgn}(\sum_{i=1}^{L} y_i \alpha_i{}^\star K(x_i, x) + b^\star) \tag{2.14}$$

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

- Linear Kernel:

$$K(x_i, x_j) = x_i.x_j \tag{2.15}$$

- Polynomial Kernel:

$$K(x_i, x_j) = (x_i.x_j + 1)^d \tag{2.16}$$

- Radial Basis Kernel:

$$K(x_i, x_j) = \exp\left(-\gamma ||x_i - x_j||^2\right) \tag{2.17}$$

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.
The distance vectors d_{intra} and d_{inter} are compiled into a Training Set and is input to the SVM with RBF kernel for training.

Chapter 3

Experimental Results

As mentioned earlier, we have used the UBIRIS database[9] for testing and training purpose. Here we have used 200 classes(1000 images) for training our SVM and 30 classes(150 images) for testing.We have used a combination of the different feature extraction techniques as mentioned earlier.The following section gives the accuracy obtained by our algorithm along with the computational time required for verification of an iris template.

The algorithm for Iris Verification has been written in *MATLAB 7.10* using *Microsoft Windows XP* operating System.The system has Intel Pentium-4 processor and 3GB RAM and 2.67 GHz of clock frequency.

For training and testing the Support Vector Machine, we have used the LIBSVM package[3].We have used *RBF-kernel* based Support Vector Machine for training and testing of our dataset.

3.1 Iris Verification Output

F/H	**Feature Combination**	**Similar Pair Accuracy**	**Dissimilar Pair Accuracy**	**Combined Accuracy**	**Verification Time(ms)**
F	GLCM+LDP+EDGE	100	99.2	99.6	3.231
F	GLCM	81.04	84.6	82.82	4.786
F	LDP	99.86	98.15	99.005	0.803
F	EDGE+LDP	99.96	98.2	99.08	1.009
F	GLCM+LDP	99.62	98.46	99.04	2.530
F/H	**Feature Combination**	**Similar Pair Accuracy**	**Dissimilar Pair Accuracy**	**Combined Accuracy**	**Verification Time(ms)**
H	GLCM+LDP+EDGE	100	98.86	99.43	1.197
H	GLCM	80.28	82.1	81.19	3.077
H	LDP	99.86	97.6	98.73	0.615
H	EDGE+LDP	99.96	97.1	98.53	0.718
F/H	**Feature Combination**	**Similar Pair Accuracy**	**Dissimilar Pair Accuracy**	**Combined Accuracy**	**Verification Time(ms)**
F	GLCM+LBP+EDGE	93.2	98.1	95.65	3.265
F	LBP	81.6	94.4	88	0.872
F	EDGE+LBP	86.1	95.3	90.7	1.556
F	GLCM+LBP	92.7	95.9	94.3	3.111
F/H	**Feature Combination**	**Similar Pair Accuracy**	**Dissimilar Pair Accuracy**	**Combined Accuracy**	**Verification Time(ms)**
H	GLCM+LBP+EDGE	91.8	97.6	94.7	2.000
H	LBP	77.9	93.8	85.85	0.803
H	EDGE+LBP	85.84	93.4	89.62	1.179
H	GLCM+LBP	90.91	93.86	92.385	1.915

F : Full Iris Template is used for testing and training
H : Half or Partial Iris Template is used for testing and training

The experimental results lead us to some interesting conclusions.

- LDP based feature extraction technique presents the best accuracy, when combined with GLCM based features and Edge based features.
- The computational time is significantly reduced by using only one por-

tion of the iris template, although the accuracy is not compromised.

3.1.1 Scope of further work

Scope of Further Work In this method for Iris Verification, we have successfully achieved a verification rate which is as good as any previously used methods.
However, there is still a lot of scope of work in the domain of proper segmentaion of the iris region. Specifically, segmentaion of iris is still a difficult job in certain scenarios where the subject is wearing a contact lens or the iris image is blurred. Occlusion by eyelids is still a problem that has not been totally resolved.
Regarding the computational time of the algorithm, our algorithm can be made to work faster by parallelising the different feature extraction processes and writing time effecient codes.

3.2 Conclusion

The implemented Iris Verification method provides excellent accuracy by using the combined features as shown in the previous section. We have also presented a comparison of the accuracies obtained by using differemt combinations of features. Another important observation is in detecting the symmetry of the iris pattern. Utilising this symmetry, we have been able to reduce the computational time significantly, without affecting the accuracy of verification. Hence we can claim to have presented an effecient method for iris verification.

Bibliography

[1] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[2] John Canny. A computational approach to edge detection.

[3] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology.*

[4] John Daugman. How iris recognition works. *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY,*, 2004.

[5] X. Ren et al. An improved method for daugmans iris localization algorithm. In *Computers in Biology and Medicine*, volume 38, pages 111–115. ELSEIVER, 2008.

[6] Taskeed Jabid etal. Local directional pattern (ldp) for face recognition, 2010.

[7] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* PEARSON, third edition, 2008.

[8] Christel loc TISSE etal. Person identification technique using human iris recognition.

[9] H. Proena and L.A. Alexandre. UBIRIS: A noisy iris image database. In *13th International Conference on Image Analysis and Processing - ICIAP 2005*, volume LNCS 3617, pages 970–977, Cagliari, Italy, September 2005. Springer.

[10] Linda G. Shapiro Robert M Haralick. *Computer and Robot Vision*, volume 1. ADDISON-WESLEY, 1992.

[11] M. Pietikainen T. Ojala. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell*, 24(7):971987, 2002.