BIOMETRIC AUTHENTICATION FOR CRITICAL APPLICATIONS

By

S. DEVI

Reg. No: 0820108007

of

KUMARAGURU COLLEGE OF TECHNOLOGY
(AUTONOMOUS)

COIMBATORE – 641 006

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
for the award of the degree
Of

MASTER OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MAY 2010

BONAFIDE CERTIFICATE

Certified that this project report titled "**Biometric Authentication for Critical Applications**" is the bonafide work of **Ms.S.Devi (0820108007)** who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report of dissertation on the basis of which a degree or ward was conferred on an earlier occasion on this or any other candidate.

GUIDE

(Dr.S.THANGASAMY.PhD,)

HEAD OF THE DEPARTMENT

(Dr.S.THANGASAMY.PhD,)

CO-GUIDE

(Ms.L.LATHA.M.E,)

The candidate with **University Register No. 0820108007** was examined by us in Project Viva-Voce examination held on 17/5/10

17/5/10

Internal Examiner

External Examiner

ABSTRACT

Person authentication techniques based on the human iris pattern is well suited for any access control system requiring highest security, due to the desirable properties of iris such as uniqueness, stability of iris patterns and non-invasiveness of measurement. A new multi-unit iris recognition method deploying the most consistent bits (best bits) in an iris code is developed. With this reduced feature set size, good recognition rate in less time is obtained compared to other methods that use all bits in the iris code. Matching score level fusion using SVM classifier, weighted sum rule, min score rule and max score rule methods are applied to enhance the recognition rate. Fusion of left and right iris improves the recognition rate from 88% to 97.89% and reduces the equal error rate from 0.06 to 0.02. Iris authentication system is demonstrated using CASIA (Ver.3, “Interval”) database. Experimental results showed that the accuracy of the proposed method was comparatively superior to previous methods that used multi-unit iris recognition.

ஆய்வுச்சுருக்கம்

கரு விழியின் தனிதன்மையும் அதன் நிலைதன்மையும் அதிகமான பாதுகாப்பை தருகிறது ஆதலால், இந்த ஆய்வில், ஒரு நபரின் கரு விழியை ஆய்வு செய்து அந்த நபர் அங்கீகரிக்கபட்டவரா இல்லையா என்று கண்டறியப்படுகிறது. இரு கருவிழிகளையும் கண்டறியும் ஒரு புதுமையான முறை மேற்கொள்ளப்பட்டுள்ளது,மேலும் இம்முறையில் பெரும்பாலும் ஒரே மாதிரியான துணுக்களை கொண்டு ஆய்வு செய்யபட்டுள்ளது. அனைத்து துணுக்களை கொண்டு கண்டறியும் முறையைவிட ஒப்பிட்டு பார்க்கும்போது இம்முறையில் கால அளவு குறைவு. மேலும் ஆய்வின் சிறப்பை அதிகப்படுத்துவதற்க்காக இடது மற்றும் வலது கரு விழியின் துணுக்களை ஒருங்கிணைந்து அதிகபடுத்தபடுகிறது. இந்த ஒருங்கிணைப்பிற்காக எஸ்விம் கிலாசிஃபியெர், வெய்ட்டெட் சம் ரூல், மின் ஸ்கோர் ரூல் மற்றும் மேஃஸ் ஸ்கோர் ரூல் முறைகள் பயன்படுத்தபடுகிறது. ஒருங்கிணைத்தல் முறையினால் ஆய்வின் மதிப்பீட்டு அளவு 88% லிருந்து 97.89% ஆக உயர்ந்துள்ளது.மேலும் தவறான மதிப்பீட்டின் அளவு 0.06 லிருந்து 0.02 ஆக குறைந்த்துள்ளது. இதற்காக கேசியா சேமிப்புத்தளதிலிருந்து கரு விழி படங்கள் எடுக்கப்படுள்ளது.இந்த ஆய்வின் முடிவில் கிடைக்கபடும் உருவ வரைபடதிலிருந்து பழைய ஆய்வு முறைகளை விட இந்த கரு விழி ஒருங்கிணைப்பு ஆய்வு முறை சிறந்தது என கண்டறியப்படுகிறது.

ACKNOWLEDGEMENT

I express my profound gratitude to our Chairman **Padmabhusan Arutselver Dr. N. Mahalingam B.Sc, F.I.E** for giving this great opportunity to pursue this course.

I would like to begin by thanking to **Dr.S.Ramachandran., PhD**, *principal* for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Dr. S.Thangasamy, Ph.D.,** Dean, *Head of the Department*, Computer Science and Engineering, my thesis advisor. I thank for his support, encouragement and ideas. I thank him for the countless hours he has spent with me, discussing everything from research to academic choices.

Special thanks to **Mrs.V.Vanitha M.E., (Ph.D)** *Assistant professor*, Department of Computer science and Engineering, for arranging the brain storming project review sessions. I thank all project committee members for their comments and advice during the reviews.

I register my hearty appreciation to **Mrs.L.Latha M.E., (Ph.D),** Assistant *Professor,* for her precious suggestions.

I would like to convey my honest thanks to all **teaching** staff members and **non teaching** staffs of the department for their support. I would like to thank all my classmates who gave me a proper light moments and study breaks apart from extending some technical support whenever I needed them most.

I dedicate this project work to my **parents** and my **friends** for no reasons but feeling from bottom of my heart, without their love this work wouldn't possible.

TABLE OF CONTENTS

CONTENTS PAGE NO

1. INTRODUCTION

2. LITERATURE REVIEW

3. METHODOLOGY

4. RESULTS AND DISCUSSIONS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
FAR	False Acceptance Rate
GAR	Genuine Acceptance Rate
FRR	False Rejection Rate
HD	Hamming Distance
RR	Recognition Rate
SVM	Support Vector Machine
DR	Detection Rate
FAR	False Alarm Rate
ROC	Region of Characteristics
CASIA	Institute of Automation,Chinese Academy of Sciences

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Biometrics

A biometric is a biological measurement of any human physiological or behaviour characteristics that can be used to verify the identity of an individual. Physiological characteristics are permanent without disturbance to the individual such as fingerprint, hand geometry, palm print, iris pattern, retina pattern, facial feature. Behavioural traits reproduce a person physiological state like human voice, gait features, dynamic signature analysis and keystroke dynamics. Biometric authentication (or identification) systems, which use physical characteristics to check a person's identity, ensure much greater security than password and number systems.

A biometric-based authentication system operates in two modes

Enrolment mode: In the enrolment mode a user's biometric data is acquired using a biometric reader and stored in a database. The stored template is labelled with a user identity (e.g., name, identification number, etc.) to facilitate authentication.

Authentication mode: In the authentication mode, a user's biometric data is once again acquired and the system uses this to either identify who the user is, or verify the claimed identity of the user. While identification involves comparing the acquired biometric information against templates corresponding to all users in the database, verification involves comparison with only those templates corresponding to the claimed identity. Thus, identification and verification are two distinct problems having their own inherent complexities.

1.1.1 Components of biometric system

A simple biometric system has four important components:

Sensor module acquires the biometric data of an individual. An example is an infrared camera that captures eye image of a user.

Feature extraction module in which the acquired data is processed to extract feature values. For example, the iris in a eye image would be extracted in the feature extraction module of a iris recognition system.

Matching module in which the feature values are compared against those in the template by generating a matching score. For example, in this module, the number of matching iris binary bit points between the query and the template will be computed and treated as a matching score.

Decision-making module in which the user's identity is established or a claimed identity is either accepted or rejected based on the matching score generated in the matching module.

The performance of a biometric system can be measured by reporting its false accept rate (FAR), false reject rate (FRR) and genuine acceptance rate(GAR) at various thresholds. These factors are brought together in a receiver operating characteristic (ROC) curve that plots the FRR against the FAR and the GAR against the FAR at different thresholds. The FAR, FRR and GAR are computed by generating all possible genuine and impostor matching scores and then setting a threshold for deciding whether to accept or reject a match. A genuine matching score is obtained when two feature vectors corresponding to the same individual are compared, and an impostor matching score is obtained when feature vectors from two different individuals are compared.

1.2 Iris recognition system

Iris recognition is a potential tool in secure personal identification and authentication due to its desirable properties such as uniqueness, non-invasiveness and stability of human iris patterns. The iris (Fig .1.1) is an annular part between the pupil and the white sclera in the human eye and it has an extraordinary texture which is unique to each human being as well as stable over an individual's lifetime.

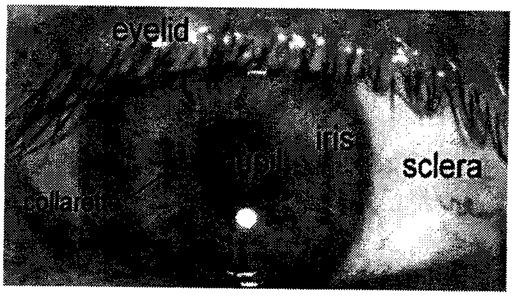


Fig.1.1 Front view of the human eye.

The average outer diameter of the iris is 12 mm, and the pupil size can vary from 10% to 80% of the iris diameter. The probability of finding two people with identical iris pattern is almost zero. Compared to fingerprint, iris is protected from the external environment behind the cornea and the eyelid. No 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. The false acceptance rate (FAR) of iris recognition system is 1 in 1.2 million statistically. Iris recognition system can able to handle very large population at high speed. It is becoming an important biometric solution for people identification in border control application such as e-passport and many secured and privacy applications.

The iris recognition system includes capturing an eye image, segmenting the image by locating the iris in an eye image, normalizing the iris region by converting it into a rectangular pattern, enhancing the normalized pattern so that the contrast of the image pixels is enhanced, encoding the image to generate the iris code which is the binary representation of the iris texture and finally matching process is performed to compute the matching ratio between the iris template and that saved in the database. After finding the matching ratio the decision has to be made whether the user is accepted or rejected based on matching threshold.

1.3 Intra modal fusion in biometrics

One of the main requirements of biometric systems is the ability of producing very low false acceptance rate (FAR), which very often can be achieved only by combining different biometric traits. The combination of different biometric traits is said to be fusion. The fusion may be of two types, intra modal and inter modal fusion. In the intra modal fusion multiple samples of same modality is taken from an individual. For example consider the left and right irises from a single individual are to be fused and another example is ten fingers from an individual is to fused. By performing these type of intra modal fusion better recognition rate (RR) and reduced false acceptance rate (FAR) is achieved. Inter modal fusion is done by combining different biometric traits. For example consider combining iris with fingerprint or iris with fingerprint and face, etc.

The three possible levels of fusion are

1.3.1 Fusion at the feature extraction level.
1.3.2 Fusion at the matching scores level.
1.3.3 Fusion at the decision level.

1.3.1 Fusion at the feature extraction level

The data obtained from each sensor is used to compute a feature vector. As the features extracted from one biometric trait are independent of those extracted from the other, it is reasonable to concatenate the two vectors into a single new vector. The new feature vector now has a higher dimensionality and represents a person's identity in a different (and hopefully more discriminating) hyperspace. Feature reduction techniques may be employed to extract useful features from the larger set of features.

1.3.2 Fusion at the matching score level

Each system provides a matching score indicating the proximity of the feature vector with the template vector. These scores can be combined together for deciding the claimed identity. The fusion at this level attempt to minimize the FRR for a given FAR.

1.3.3 Fusion at the decision level

Each sensor can capture multiple biometric data and the resulting feature vectors individually classified into the two classes-accept or reject. A majority vote scheme, can be used to make the final decision.

1.4 Benefits of Fusion technique:

- Multimodal biometric systems are expected to be more reliable due to the presence of multiple pieces of evidence.
- Multimodal systems address the problem of non-universality: it is possible for a subset of users to not possess a particular biometric.
- Multimodal systems also provide anti-spoofing measures by making it difficult for an intruder to spoof multiple biometric traits simultaneously.

1.5 Applications of biometrics:

Biometric authentication methodology can be used in various applications such as

- Border control applications such as e-passport
- Universal identity card application
- Banking and financial applications

CHAPTER 2

LITERATURE REVIEW

2.1Iris recognition system

Fingerprint recognition is the most widely used and least expensive biometrics technology, but it can be counterfeited easily by a “rubber finger”. The face may change much when an individual gains weight or ages, and may be modified by hair, beard or accessories as well. Iris, instead, is the only visible internal organ of human; it is stable and very hard to be altered throughout the whole life of a human being. Moreover its physiological response to light (pupil vibration and dilation) provides a natural test against fake irises.

Several methods have been proposed for iris recognition. In the field of the iris recognition, there are many efforts that have been done. All these efforts mainly aim to convert an acquired iris image into a suitable code that can be easily manipulated. Thus, a brief looks at the process of iris recognition from some remarkable works have been discussed.

Daugman [11] developed an actual algorithm for iris recognition. This algorithm forms the ground of all current iris recognition systems. It finds the iris in a live video image of a person’s eye, defines a circular pupillary boundary between the iris and the pupil portions of the eye, and defines another circular boundary between the iris and the sclera portions of the eye. The algorithm fits the circular contours via Integrodifferential operator, and normalizes the iris ring to a rectangular block of a fixed size. After that it finds a 2,048-bit iris code according to the real and imaginary parts of 2D Gabor filters outputs. By using the hamming distance, the algorithm compares the code with stored iris codes.

Wildes et al.[12] developed iris recognition system that uses two cameras. The first one is with low resolution and the second one with high resolution. The obtained image from the second camera is processed to extract the iris region by using a binary edge map and Hough transform. In the next step, the iris image is filtered by Laplacian of Gaussian filters to get a Laplacian pyramid constructed with four different resolution levels to represent the iris texture. Then the normalized correlation between the acquired and database representation are used for pattern matching.

Boles and Boashash [15] proposed an algorithm for iris feature extraction using zero-crossing representation of 1D wavelet transform at various resolution levels of a concentric circle on an iris image. Two dissimilarity functions were used to perform iris matching.

Zhu et al.[16] adopted multi-channel Gabor filters and 2D wavelet transform to capture both global and local details in an iris. Weighted Euclidean distance was used for matching process.

Lim et al.[13] decomposed an iris image into four levels using 2D Haar wavelet transform and quantized the fourth-level high frequency information to form an 87-bit code. A modified competitive learning neural network (LVQ) was used for classification.

Tisse et al.[17] implemented a combination of the gradient decomposed Hough transform/Integrodifferential operators for iris localization and the analytic image (a combination of the original image and its 2D Hilbert transform) to extract pertinent information from iris texture. Similar to the algorithm by Daugman, they sampled binary emergent frequency images to form a feature vector and used Hamming distance for matching.

Ma et al. [14]adopted multi-channel Gabor filters for feature extraction and weighted Euclidean distance for matching. Ma et al. **[18]** adopted circular symmetric filters for feature extraction and a modified nearest feature line method for matching. In another work by Ma et al. **[19],** they constructed a bank of spatial filters for efficient feature extraction. Fisher linear discriminant was first used to reduce the dimensionality of the feature vector and then the nearest centre classifier was adopted for classification.

In all iris recognition system as discussed above, they considered all the bits in an iris code which is obtained after encoding. Matching is performed by comparing all the bits between the two templates. The recognition rate (RR) was good.

Karen et al [1] proposed that not all of the bits in an iris code are equally useful .They document that some bits are more consistent than others. Different regions of the iris are compared to evaluate their relative consistency and, contrary to some previous research, they find that the middle bands of the iris are more consistent than the inner bands. The inconsistent-bit phenomenon is evident across genders and different filter types. Possible causes of inconsistencies such as segmentation, alignment issues, and different filters, are investigated. The inconsistencies are largely due to the coarse quantization of the phase response. Masking iris code bits corresponding to complex filter responses near the axes of the complex plane improves the separation between the match and non-match Hamming distance distributions.

2.2 Fusion in Biometrics:

Multi-biometric fusion is typically applied in one of three specific steps in the classification process: fusion of the input feature vectors, fusion of the match scores output by the individual classifiers, or fusion at the decision level. Fusion of the input feature vectors is not always feasible, as these input features may not be directly accessible via professional biometric collection systems. Additionally, the specific input features that are used by the different modalities may not be compatible. Fusion at the match score level is applicable to general multi-biometric systems, and has been shown to be more informative than decision-level fusion.

Supervised parametric learning techniques, such as SVMs and Bayesian networks, have been shown empirically to produce more accurate fusion results than either voting or non-parametric learning techniques such as sum, fisher, and product. Of these parametric learning-based fusion techniques, the SVM appears to be the most popular choice in the literature. In short, the SVM learns to map the vector of individual biometric match scores into a joint (i.e., fused) match score or classification.

Score level fusion is commonly preferred in multimodal biometric systems because matching scores contain sufficient information to make genuine and impostor case distinguishable and they are relatively easy to obtain. Given a number of biometric systems, matching scores for a pre-specified number of users can be generated even with no knowledge of the underlying feature extraction and matching algorithms of each system. Therefore, combining information obtained from individual modalities using score level fusion seems both feasible and practical. Since the scores generated by a biometric system can be either similarity scores or distance scores, one needs to convert these scores into a same nature by normalization. It is not only need to enhance the performance but also it reduces the time consumed by the algorithm to produce results either on the training level or on the test level.

The following issues are the key behind the critical need for score normalization for biometric fusion:

- The matching scores at the output of the individual matchers may not be homogeneous. For example, one matcher may output a distance (dissimilarity) measure while another may output a proximity (similarity) measure.
- The outputs of the individual matchers need not be on the same numerical scale (range).

- The matching scores at the output of the matchers may follow different statistical distributions.

Due to these reasons, score normalization is essential to transform the scores of the individual matchers into a common domain prior to combining them.

Kittler et al [23] have experimented with several fusion techniques for face and voice biometrics, including sum, product, minimum, median, maximum rules and they have found that the sum rule outperformed others. Kittler et al. note that the sum rule is not significantly affected by the probability estimation errors and this explains its superiority.

Ben-Yacoub et al. [24] considered several fusion strategies, such as support vector machines, tree classifiers, and multilayer perceptrons, for face and voice biometrics. The Bayes classifier is found to be the best method.

Ross and Jain [8] combined face, fingerprint, and hand geometry biometrics with sum, decision tree, and linear discriminant-based methods. The authors report that the sum rule outperforms others.

2.3 Multi-unit iris recognition:

Multi unit iris recognition involves the fusion of both the left and right irises. Until now, there has been little research into multi-unit iris recognition. **Jang [21]**, multi-unit iris recognition based on quality assessment was proposed. However, that research only used a single good quality iris image (left or right iris) even though both iris images might have been good quality, so the recognition performance was not really improved. In detail, in paper **[21]**, they selected one good quality iris image by quality assessment measurement based on binary selection ("good quality" or "poor quality") and did not adopt the scheme of using (or rejecting) two iris images both of which were determined as good (or bad) ones. In paper **[21]**, the author proposed a classifier level fusion method. This was a dynamic selection method that used one good quality iris image among the left and right iris images based on a quality checking pre-processing step. However, the researchers only used simple AND or OR rules in the classified levels. So, these approaches were simple and easy to implement, but the levels of accuracy were not sufficient compared to those achieved by score level fusion methods.

Kang et al[2] proposed a new multi-unit iris authentication method based on quality assessment and SVM fusion on a mobile phone. First, they assessed the quality of the left and right iris images. If one of both iris images was of good quality, identification was performed with a single HD that was calculated from the good quality image. However, if both iris images were of good quality, they performed a multi-unit iris recognition method based on score level fusion using the SVM. For score level fusion, they used a typical HD produced by a Gabor filter, which was easily applied to a conventional iris recognition system.

CHAPTER 3

METHODOLOGY

3.1 Iris Recognition system

The system is composed of a number of sub-systems, which correspond to each stage of iris recognition. These stages are segmentation (localization) -locating the iris region in an eye image, normalization - creating a dimensionally consistent representation of the iris region, enhancement-improving the contrast of the normalized iris image, feature encoding – creating an iris code containing only the most discriminating features of the iris and finally matching of the best bits is performed between two iris templates to make a decision of inter class and intra class iris. Both the left and right irises are recognized and finally score level fusion is performed by sum rule, minimum rule, maximum rule and support vector machine classifier. With the output of fusion decision is made for authentication. The overall diagram of iris recognition system is shown in Fig.3.1

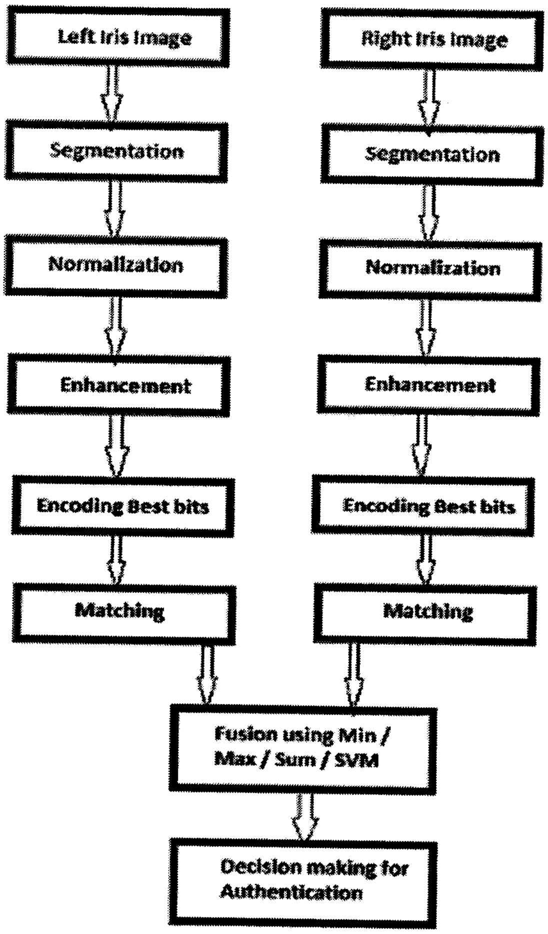


Fig.3.1 Iris Recognition system

3.1.1 Iris Pre-processing:

An iris image needs to be pre-processed before using it for the recognition purpose, as the unwanted data in the image such as eyelid, eyelashes, pupil and specular reflections should be excluded. Therefore, pre-processing is required to segment, normalize and exclude the artifacts [3]. The overview of the pre-processing flow is shown in (Fig. 3.2)

3.1.1.1 Segmentation:

Segmentation is the first stage in iris pre-processing to isolate the actual iris region from a captured eye image. **Canny edge** detection is performed to create an edge map using gradients information .The canny edge method finds edges by looking for local maxima of the gradient of the input image. The gradient is calculated using the derivative of a Gaussian filter. The method uses two thresholds, to detect strong and weak edges, and includes the weak edges in the output only if they are connected to strong edges. This method is therefore more robust to noise, and more likely to detect true weak edges.

The output of the canny edge detector is the edge strength image and the orientation image. The image intensity value has to be increased by **adjusting the gamma** correction factor. With the orientation image and the adjusted gamma image, as the input the **local maxima are suppressed.**

Thresholding with hysteresis requires two thresholds – high and low. Making the assumption that important edges should be along continuous curves in the image allows to follow a faint section of a given line and to discard a few noisy pixels that do not constitute a line but have produced large gradients. Therefore begin by applying a high threshold. These marks out the edges that is fairly genuine. Starting from these, using the directional information derived earlier, edges can be traced through the image. While tracing an edge, apply the lower threshold, allowing to trace faint sections of edges as long as to find a starting point. Therefore after the hysteresis thresholding all the edges are clearly identified.

The Hough transform is a standard computer vision algorithm that can be used to determine the parameters of simple geometric objects, such as lines and circles, present in an image. The circular Hough transform can be employed to deduce the radius and centre coordinates of the pupil and iris regions according to the equation

$$x_c + y_c - r^2 = 0 \qquad (1)$$

where x_c, y_c are the centre coordinates of the circle and r is the radius of the circle.

A maximum point in the Hough space will correspond to the radius and centre coordinates of the circle best defined by the edge points. In order to make the circle detection process more efficient and accurate, the Hough transform for the iris-sclera boundary was performed first, then the Hough transform for the iris-pupil boundary was performed within the iris region, instead of the whole eye region, since the pupil is always within the iris region. After this process was complete, six parameters are stored; the radius, and x, y centre coordinates for both circles. Eyelids were isolated by first fitting a line to the upper and lower eyelid using the linear Hough transform which is also called as Radon transform. The eyelashes are removed by thresholding. The overall method is very efficient and reliable as it managed to segment the iris region perfectly and isolate most occluding eyelashes occurring within the iris region.

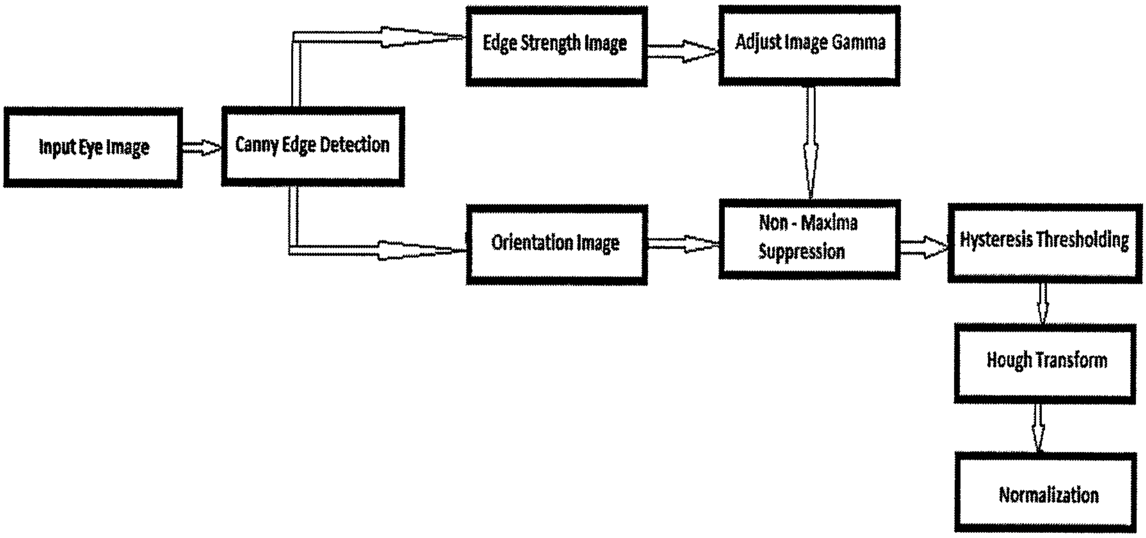


Fig.3.2 overview of iris pre-processing

3.1.1.2 Normalization

Normalization is a process of transforming the segmented iris region into fixed dimension. The dimensional inconsistencies between different eye images are mainly due to the stretching of the iris caused by pupil dilation from varying levels of illumination. The purpose of normalization is to compensate the iris deformation, which is caused by illumination variations. Such elastic deformation in iris texture will affect the result of iris matching. The iris regions with same dimensions will be used for comparisons. For this normalization process, Daugman's rubber sheet model [17] has been applied (Fig.3.3).

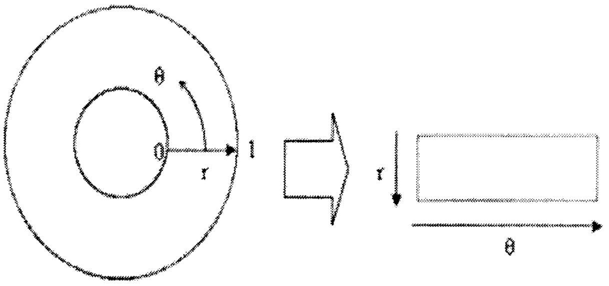


Fig.3.3 Daugman's rubber sheet model

Daugman's rubber sheet model remaps the annular iris image $I(x, y)$ from raw Cartesian coordinates (x, y) to a dimensionless pseudo-polar coordinate system (r, θ) [1].

$$I(x(r, \theta), y(r, \theta) \rightarrow I(r,\theta) \tag{2}$$

with

$$x(r, \theta) = (1-r)x_p(\theta) + r x_l(\theta) \tag{3}$$

$$y(r, \theta) = (1-r)y_p(\theta) + r y_l(\theta) \tag{4}$$

where r lies on [0, 1] and θ lies on $[0, 2\pi]$, while$(x_p(\theta), y_p(\theta))$ and $(x_l(\theta), y_l(\theta))$ are the coordinates of the pupillary (inner) and limbic (outer) boundaries in the direction θ, respectively.

The rubber sheet model takes into account pupil dilation and size inconsistencies in order to produce a normalized representation with constant dimensions. In this way the iris region is modelled as a flexible rubber sheet anchored at the iris boundary with the pupil centre as the reference point. The normalization produces a 2D array with horizontal dimensions of angular resolution and vertical dimensions of radial resolution with the fixed dimension of 20 X 240, where 20 is the radial resolution and 240 is the angular resolution. The normalization process proved to be successful and some results are shown in Fig.3.4

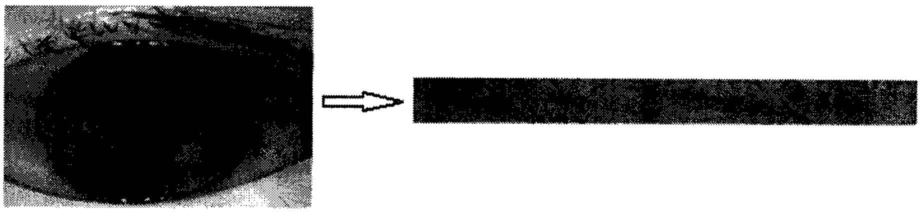


Fig.3.4 Normalized pattern of iris image

3.1.2 Normalized Iris Enhancement

The normalized pattern is enhanced for improving the contrast of the image. For this enhancement histogram equalization method is used. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. The normalized iris pattern after enhancement is shown in Fig.3.5

Fig. 3.5 Enhanced iris image

3.1.3 Feature encoding

Feature encoding is an important process in iris recognition. Its objective is to extract the underlying information in an iris pattern, to be encoded for the matching purposes. Convolving the normalized iris pattern with 1D Log-Gabor filters generates the iris feature set.

$$G(f) = \exp\left(\frac{-(\log(f/f_0))^2}{2(\log(\sigma/f_0))}\right) \tag{5}$$

where f_0 represents the centre frequency, and σ gives the bandwidth of the filter.

By applying 1D Log- Gabor Filters, 2D normalized pattern is divided into a number of 1D signals, and these 1D signals are convolved with 1D Gabor wavelets. The rows of the 2D normalized pattern are taken as the1D signal; each row corresponds to a circular ring on the iris region. The angular direction is taken rather than the radial one, which corresponds to columns of the normalized pattern, since maximum independence occurs in the angular direction. The resultant complex features are phased quantized and are then encoded into binary iris templates.

3.1.4 Matching

Matching is a process to determine whether two iris templates are from the same individual or not. For matching, the Hamming distance was chosen as a metric for recognition, since bit-wise comparisons were necessary. The Hamming distance algorithm employed also incorporates noise masking, so that only significant bits are used in calculating the Hamming distance between two iris templates [8].

$$HD = \frac{\|(\mathit{code}\ A \otimes \mathit{code}\ B) \cap (\mathit{mask}\ A \cap \mathit{mask}\ B)\|}{\|(\mathit{mask}\ A \cap \mathit{mask}\ B)\|} \qquad (6)$$

where HD is the Hamming distance, code A and code B are respectively the bit-code of A and B, mask A and mask B are respectively the mask of noise of A and B.

Now when taking the Hamming distance, only those bits in the iris pattern that corresponds to '0' bits in noise masks of both iris patterns will be used in the calculation. The Hamming distance will be calculated using only the bits generated from the true iris region. Although, in theory, two iris codes generated from the same iris will have a Hamming distance of 0.0, in practice this will not occur. Normalization may not be perfect, and also there will be some noise that goes undetected, so some variation will be present when comparing two intra class iris templates. The hamming distance calculation between two templates is shown in (Fig.3.6)

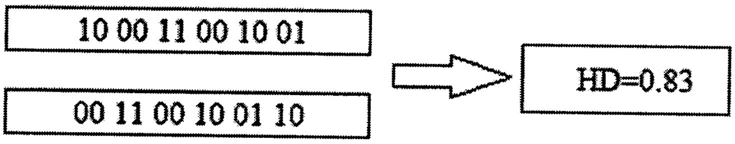


Fig. 3.6 Hamming distance calculation

3.2 Iris Recognition by Best Bit

The output of the feature encoding process is the binary iris code. In that iris code not all the bits are equally useful. Some bits are more consistent than other bits [2]. The bits in the middle rows of the iris were more consistent than outer and inner bands. Du et al. [14] suggested that "a more distinguishable and individually unique signal is found in the inner rings of the iris" and that "as one traverses to the limbic boundary of the iris, the pattern becomes less defined and ultimately less useful in determining identity." So towards both the pupil and the limbic boundary the bits are more fragile. Due to this the percentages of fragile bits in each row of the iris code, Rows in the middle of the iris code (rows 5 through 12) are the most consistent bits.

The inconsistencies in the outer and inner ring of the iris code are due to the specular reflections of light, eyelid and eyelashes occlusion. In this experiment the most consistent bits are taken and performed the feature encoding process. The output, which is the binary iris code, is used for matching between the two templates. The hamming distance between those templates, which have deployed the best bits, is reduced. Fig.3.7 shows the fraction of fragile bits in each row, and it is clear that from the rows 5 to 12 the percentage of fragile bits is lower when compared to other rows. So, the middle band is considered for processing.

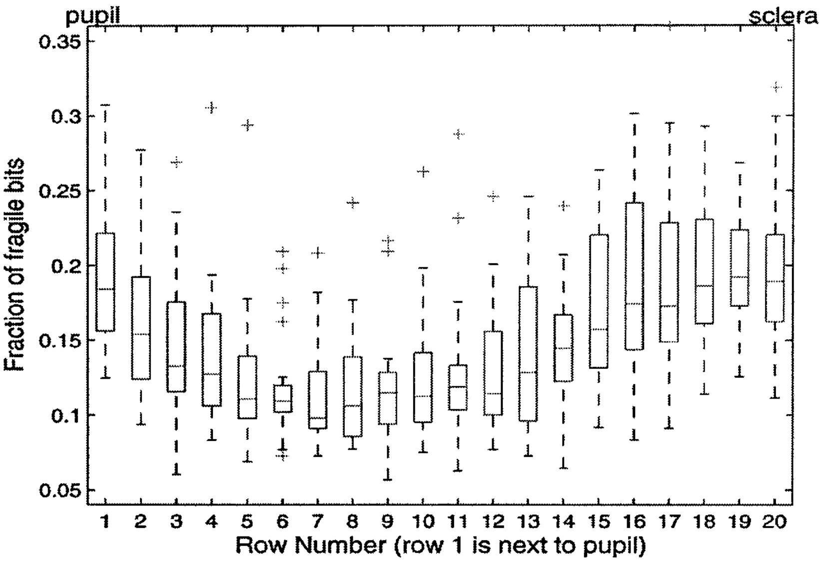


Fig. 3.7 The percent of fragile bits in each row of the iris code.

3.3 Multi-unit iris recognition

Multi-unit iris recognition method is the fusion of both the irises in either one level, i.e. in feature extraction level, or matching score level or decision making level. In this project a new multi-unit iris recognition method is proposed which uses only the best bits in the iris code from both the irises and fusion is performed at the matching score level using weighted sum rule, minimum score rule, maximum score rule and support vector machine classifier. Before fusion the matching scores have to be normalized to enhance the performance. Since the hamming distance values ranges from 0 to 1, there is no need of normalization in this multi-unit iris recognition.

3.3.1 Fusion at the matching score level

Fusion can be performed in three levels as shown in fig.3.8. In this project fusion at the matching score level is performed. The matching scores from both the irises are fused. The various fusion technologies deployed are weighted sum rule, minimum score rule, maximum score rule and support vector machine classifier and each method is discussed as follows.

3.3.1.1 Weighted sum rule

The simplest form of combination would be to take the weighted average of the scores from both the irises. This strategy was applied to all possible combinations of the two modalities. Equal weights were assigned to each modality as the bias of each classifier was not computed.

$$f_i = W\sum_{m=1}^{M} n_i^m, \forall_i \qquad (7)$$

Where W is the weight assigned to each modality, f_i is the fused score for user i, the quantity n_i^m represents the normalized score for matcher m (m=1,2,...,M, where M is the number of matchers) applied to user i (i=1,2,...,I, where I is the number of individuals in the database).

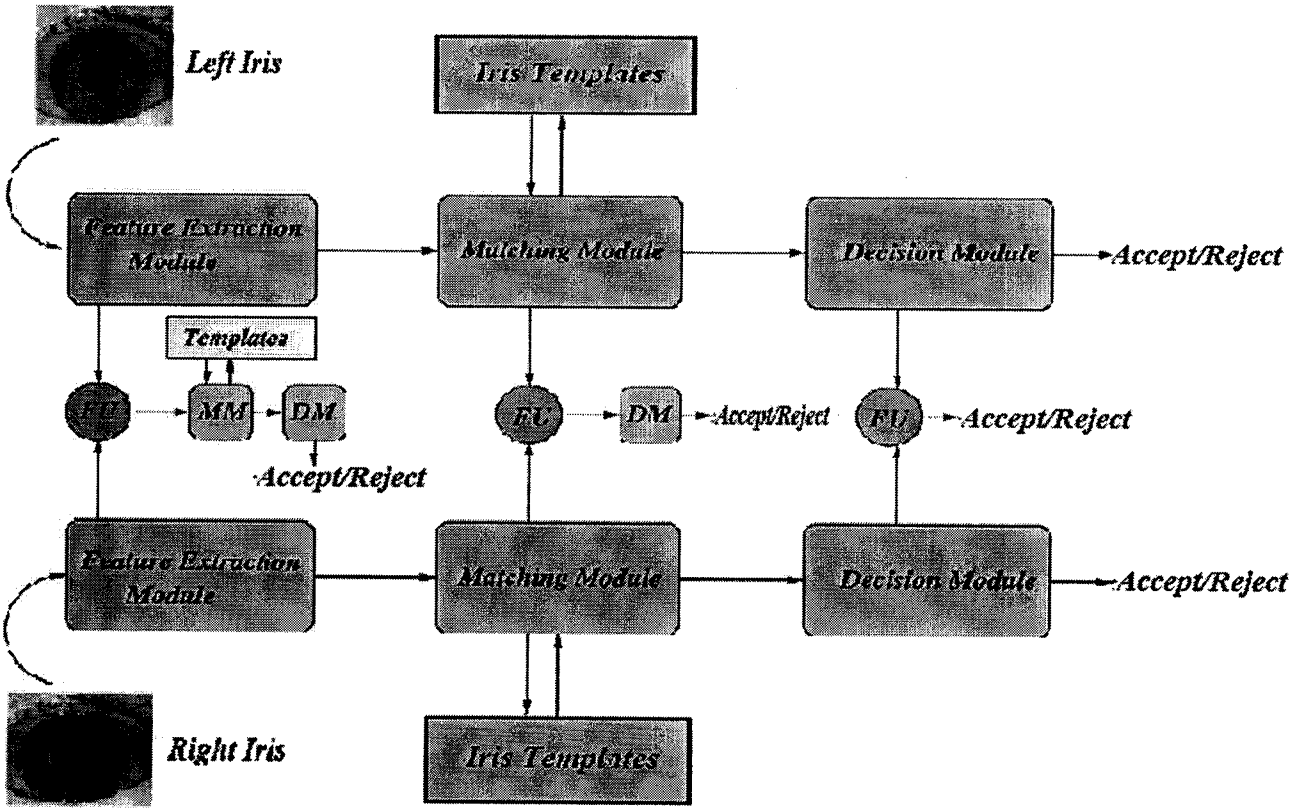


Fig.3.8 Three possible levels of fusion

3.3.1.2 Minimum score rule

The minimum scores from both the irises have to be taken for this rule. From the left and right iris whichever has the lowest hamming distance is used for fusion.

$$f_i = min(n_i^1, n_i^2,..., n_i^M), \forall_i \qquad (8)$$

where, f_i is the fused score for user i, the quantity n_i^1, n_i^2 represents the normalized score for matcher 1,2,...,M, where M is the number of matchers applied to user i (i=1,2,...,I, where I is the number of individuals in the database).

3.3.1.3 Maximum score rule

The maximum scores from both the irises have to be taken for this rule. From the left and right iris whichever has the highest hamming distance is used for fusion.

$$f_i = \max (n_i^1, n_i^2, \ldots, n_i^M), \forall_i \qquad (9)$$

where, f_i is the fused score for user i, the quantity n_i^1, n_i^2 represents the normalized score for matcher 1,2,...,M, where M is the number of matchers applied to user i (i=1,2,...,I, where I is the number of individuals in the database).

3.3.1.4 Support Vector Machine classifier (SVM)

SVMs are a set of related supervised learning methods used for classification and regression. When a set of training examples was given, each marked as belonging to one of two categories; an SVM training algorithm builds a model that predicts whether a new example falls into one category or the other. Intuitively, an SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

A support vector machine constructs a hyperplane or set of hyperplanes in a high or infinite dimensional space, which can be used for classification, regression or other tasks. Naturally a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin lower the generalization error of the classifier.

In SVM, when more number of samples are trained there will be less number of error rate. For example, consider for 100 samples. The accuracy rate will be better when we train 90 samples and the remaining samples are given for testing. In this project the kernel method used for SVM classification is linear, rather than radial or polynomial kernel. Fig.3.9 shows the SVM training model, the model constructs three hyperplanes H1, H2, H3. H3 (green) doesn't separate the 2 classes. H1 (blue) does, with a small margin and H2 (red) with the maximum margin.

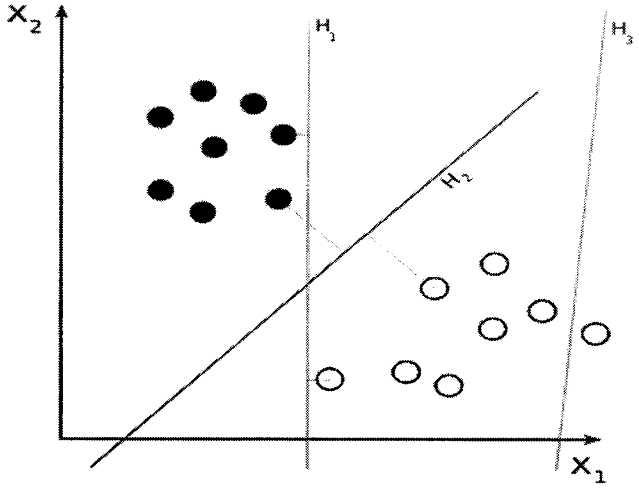


Fig.3.9 SVM separating hyperplanes

Fig.3.10 shows the maximum-margin hyperplane that separates two classes. Samples on the margin are called the support vectors. If the number of support vector are high then there will be more number of error.

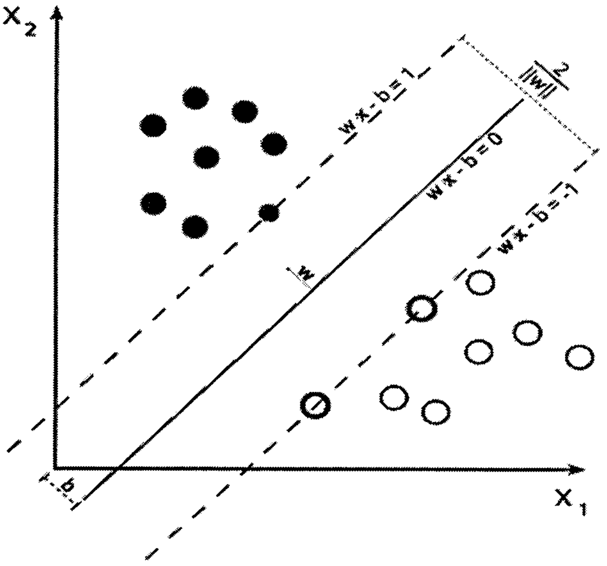


Fig.3.10 SVM maximum separating hyperplane with margin

3.4 Database Technology used

For iris recognition Chinese CASIA (version 3) iris image database collected by Institute of Automation, Chinese Academy of Sciences is used. There are totally three folders namely interval, lamp and twins. Interval is taken for recognition which has 248 subjects with both left and right eye image.

CHAPTER 4

RESULTS AND DISCUSSION

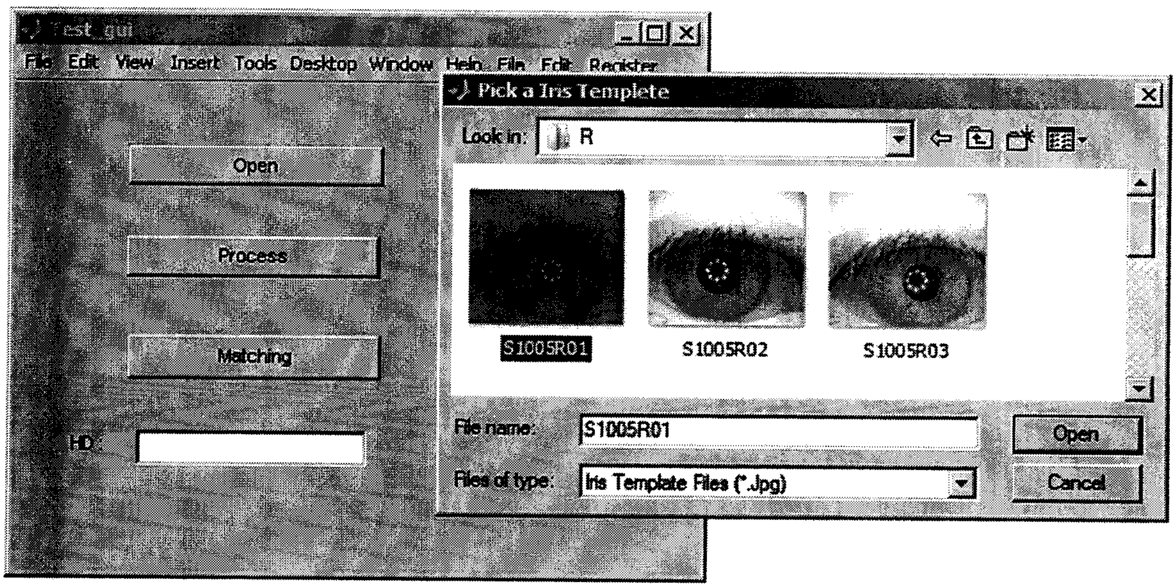
Iris recognition system and the fusion technique are experimented on a Intel core i3 2.27 GHz with 4GB of RAM on Windows 7 Home Premium 64 bit using MATLAB 7.2 software.

4.1 Iris recognition system

The iris recognition system consists of the following steps.

- Open an eye image from the database.
- Process the image i.e. segmentation, normalization, enhancement and encoding.
- Matching is performed, for this step another eye image from the database should be opened and processed. Then the hamming distance values between the two images are displayed. Based on the hamming distance values inter and intra class can be identified.

The various snapshots of the system are provided as follows.

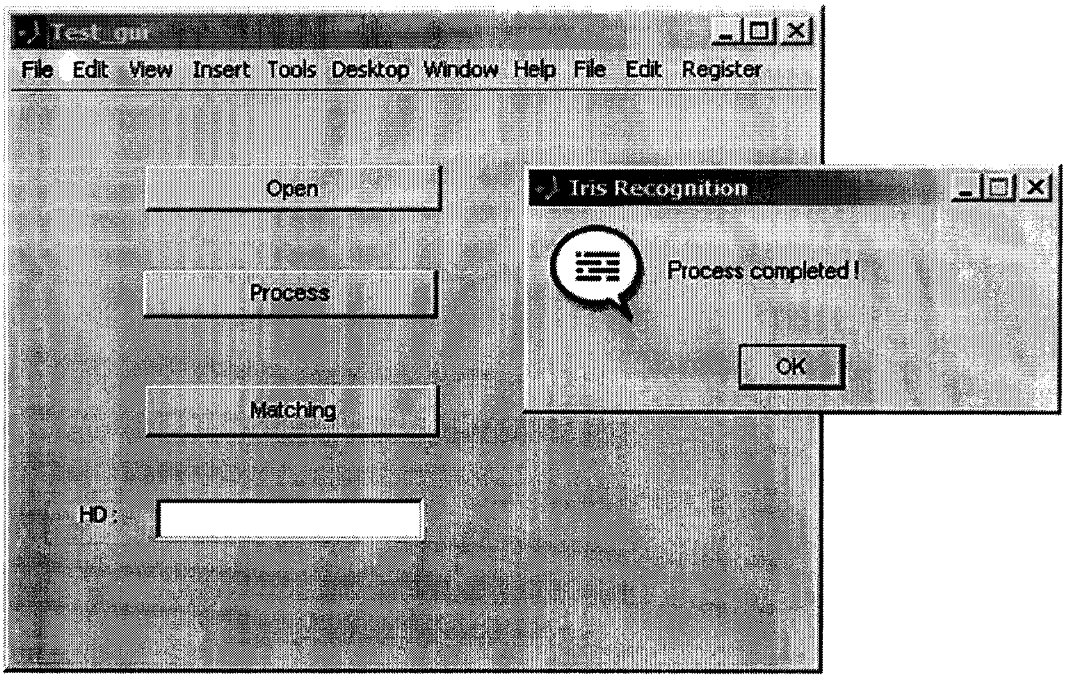


Look in: R

File name: S1005R01

Files of type: Iris Template Files (*.Jpg)

Fig.4.1 Open an eye image from the database



Process completed !

Fig.4.2 Process the image

Processed images are as follows,

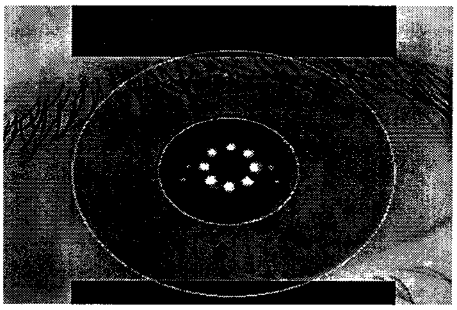


Fig.4.3 Image with noise

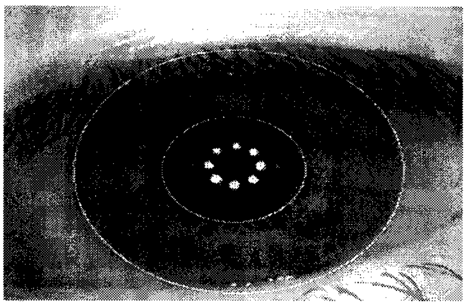


Fig.4.4 Segmented image

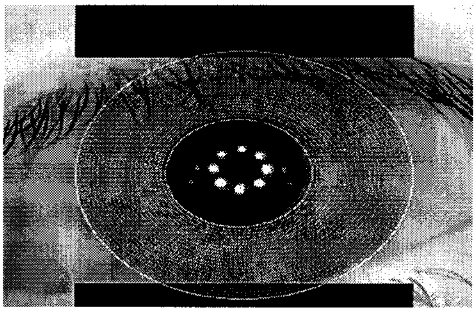


Fig.4.5 Normalized image

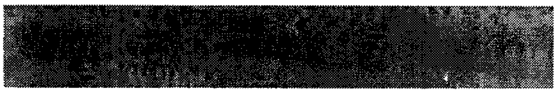
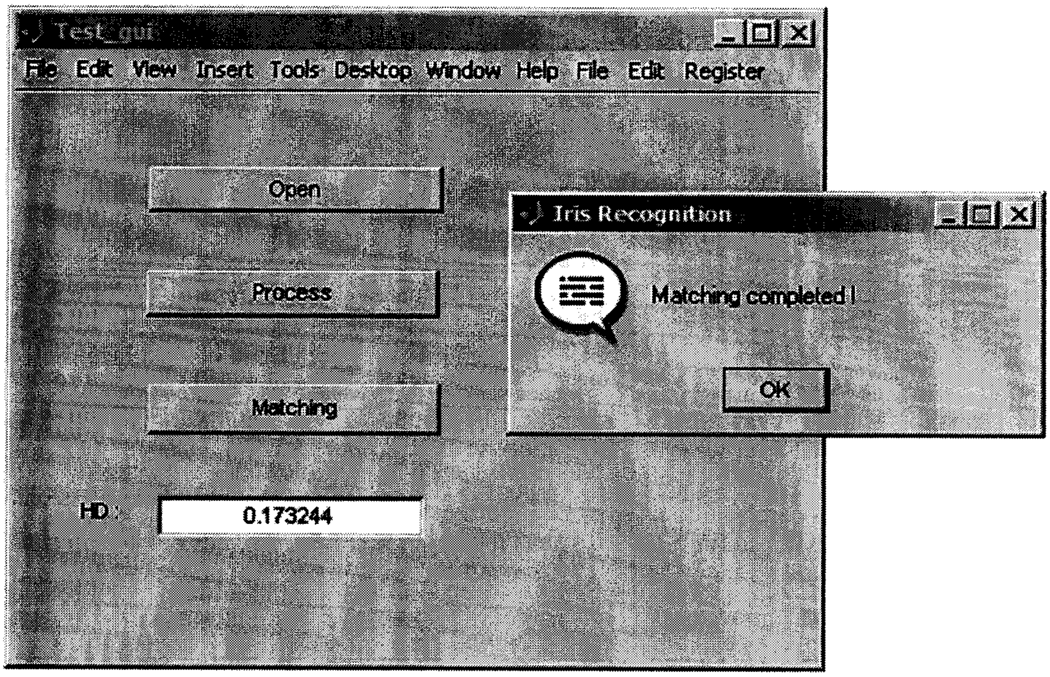


Fig.4.6 Polar representation of normalized image

Fig.4.7 Polar noise image

Fig.4.8 Enhanced image



Matching completed !

HD : 0.173244

Fig.4.9 Image displays the HD value

4.1.1 Interclass and Intraclass variations graph for recognition system

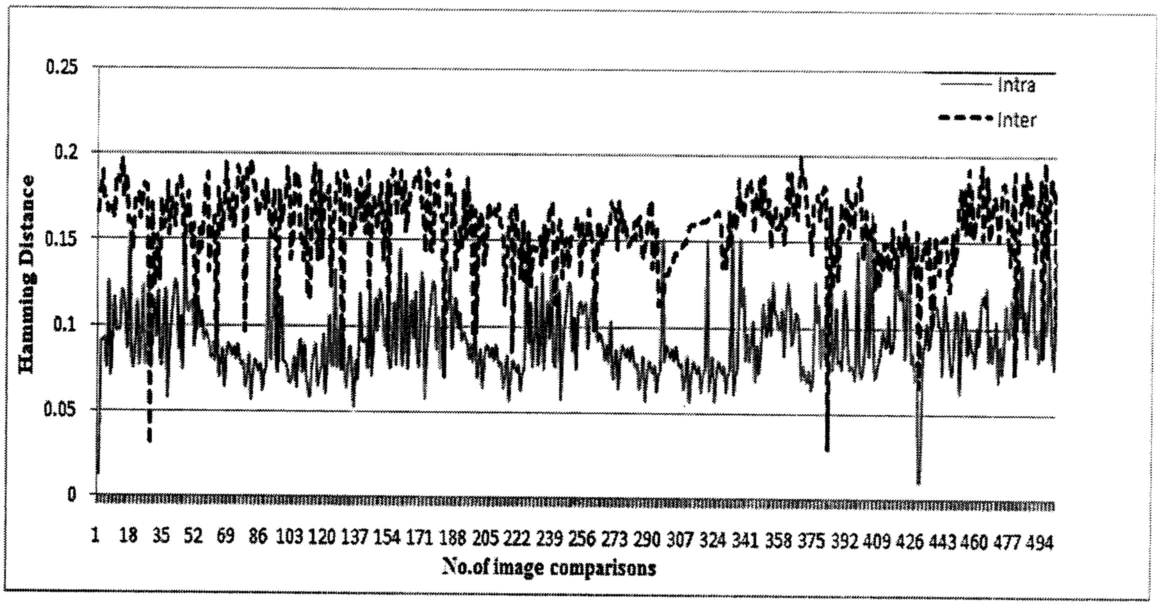


Fig.4.10 Left iris recognition

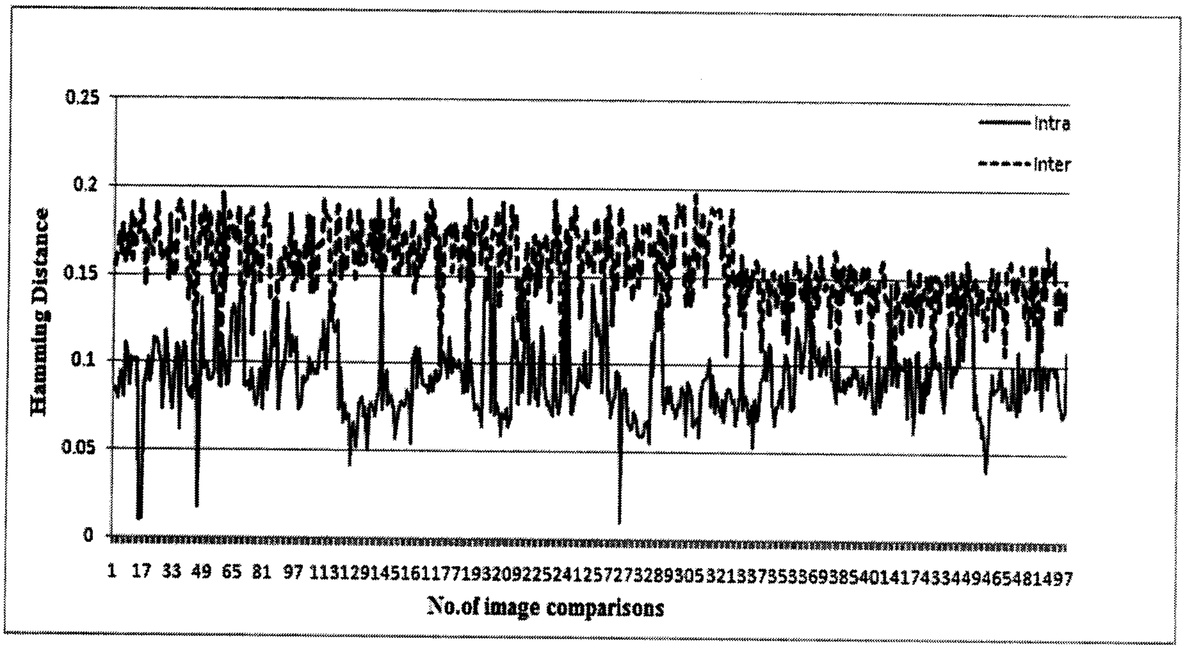


Fig.4.11 Right iris recognition

4.1.2 Interclass and Intraclass variations graph for fusion technologies

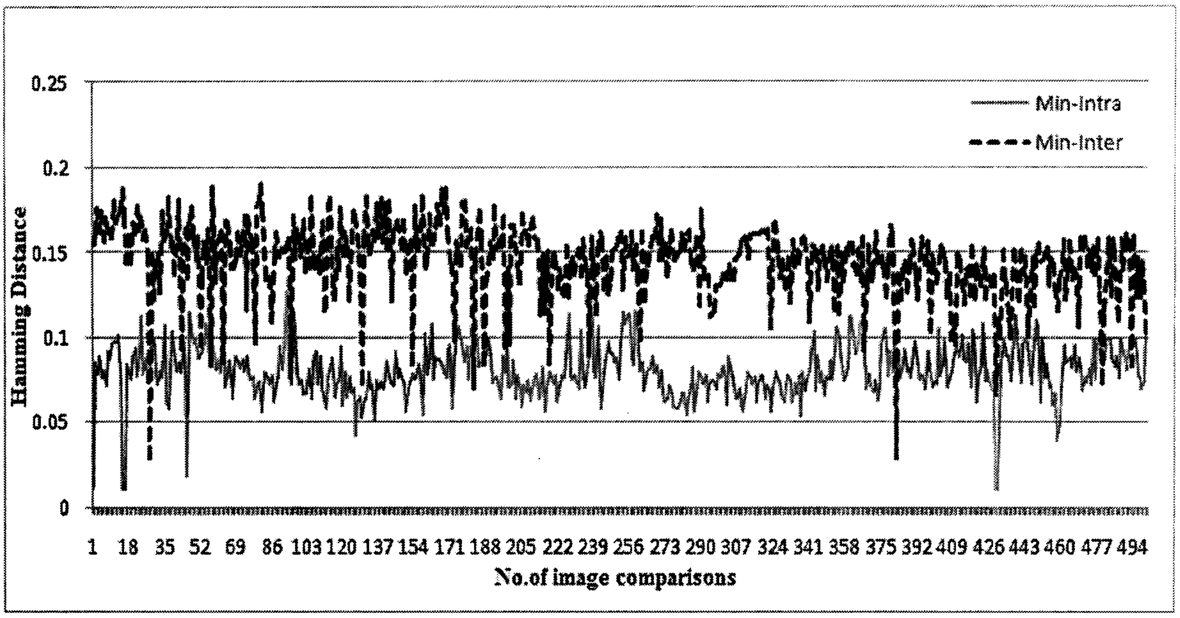


Fig.4.12 Minimum score level fusion

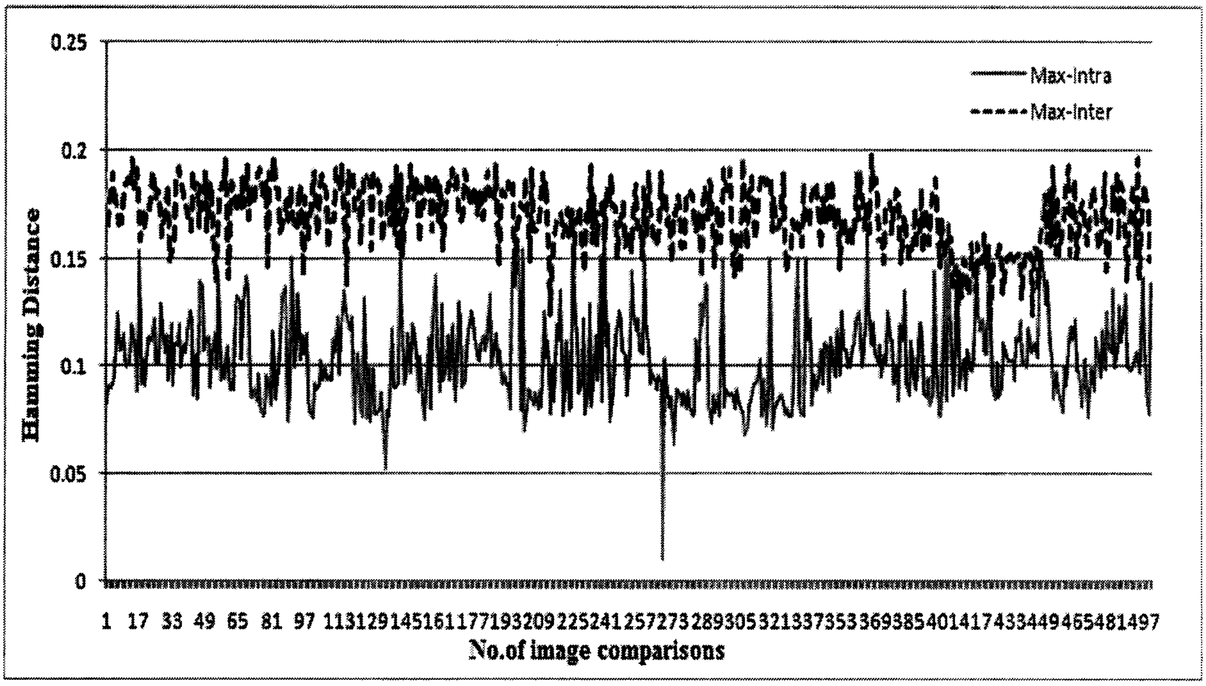


Fig.4.13 Maximum score level fusion

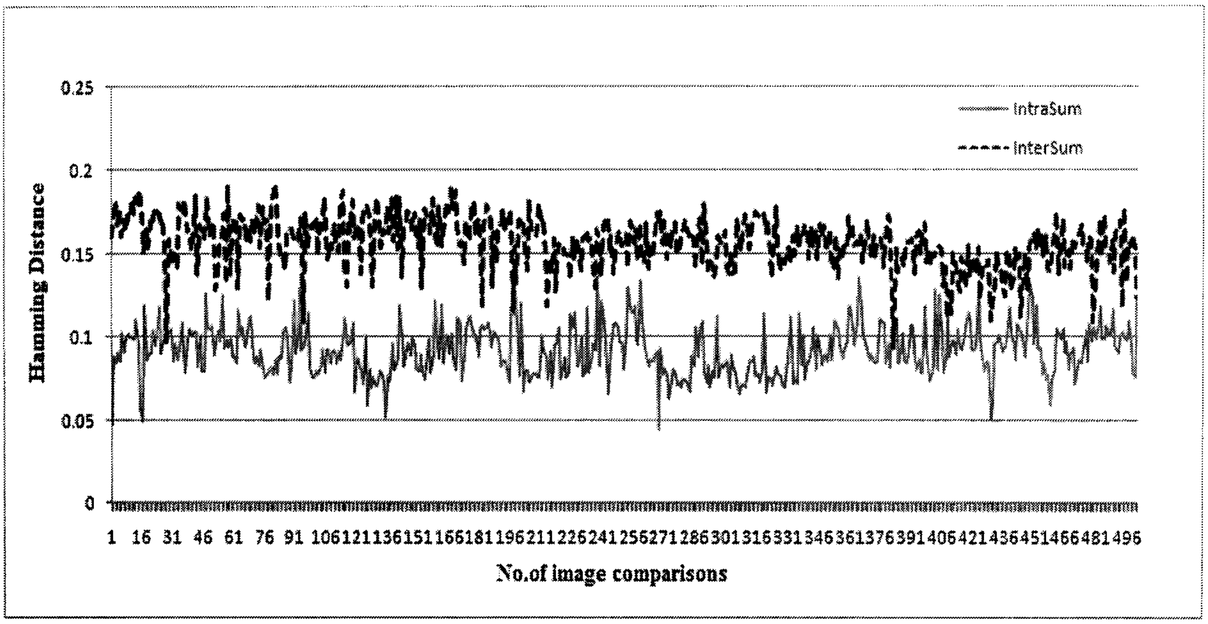


Fig.4.14 Weighted sum rule fusion

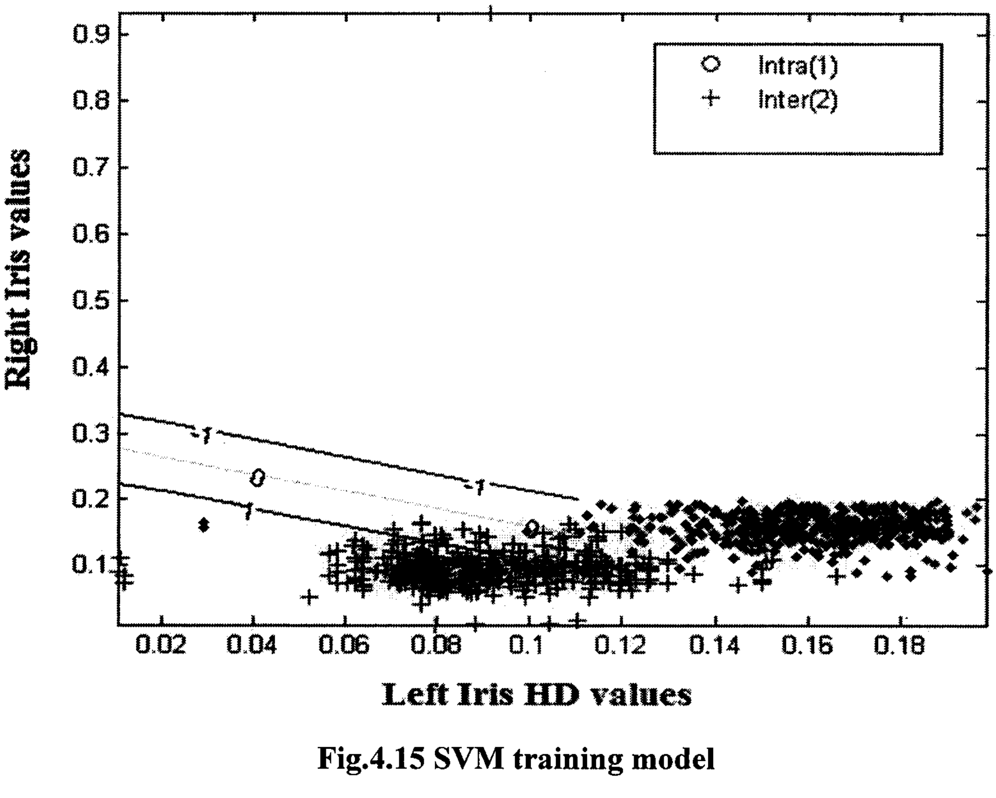


Fig.4.15 SVM training model

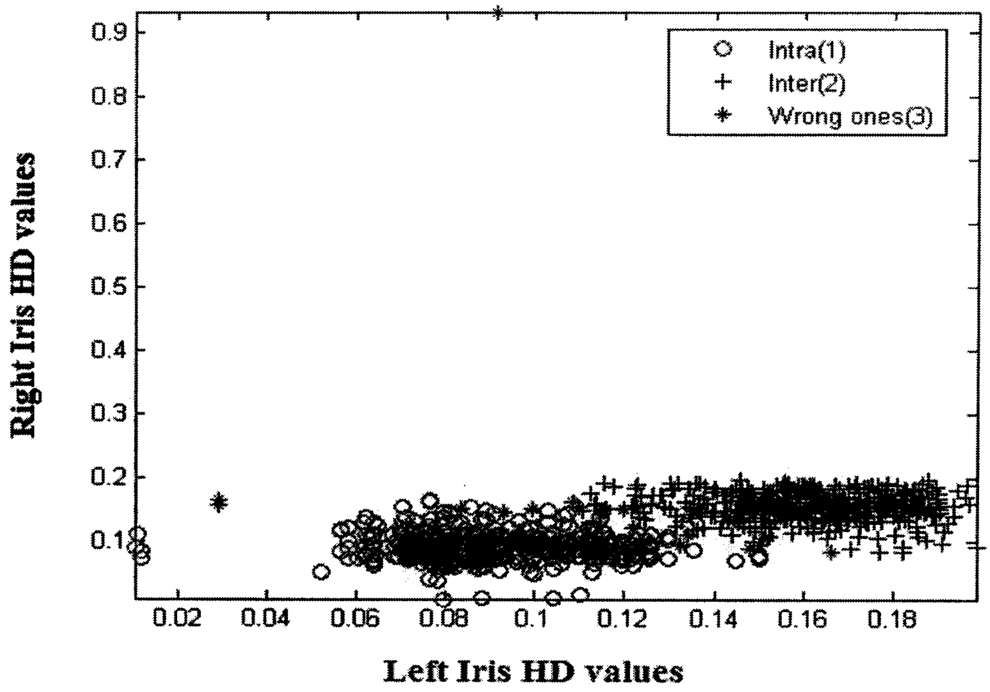


Fig.4.16 SVM classification

4.2 Performance Results

At various threshold levels Genuine Acceptance Rate (GAR), False Acceptance Rate (FAR), False Rejection Rate (FRR) are tabulated for individual left and right iris recognition system and also for various fusion technologies.

Threshold	GAR	FAR	FRR
0.05	2.6	0.4	97.4
0.07	33.2	1.4	66.8
0.09	68.4	3.6	31.6
0.11	90.2	5.2	9.8
0.13	97	15.6	3
0.14	98	27.4	2
0.15	99.8	47.4	0.2
0.17	100	82	0
0.19	100	100	0

Table 4.1 For Left iris (in %)

Threshold	GAR	FAR	FRR
0.05	3.8	0	96.2
0.07	31.8	0	68.2
0.09	72.4	2	27.6
0.11	90.6	5.8	9.4
0.13	96.6	19	3.4
0.15	99.4	57.4	0.6
0.16	100	69.8	0
0.17	100	82.6	0
0.19	100	100	0

Table 4.2 For Right iris (in %)

Threshold	GAR	FAR	FRR
0.05	6	0.4	94
0.06	19	0.6	81
0.07	54.8	1.4	45.2
0.09	91.2	5.6	0.8
0.11	99.6	11	0.4
0.13	100	31.6	0
0.14	100	52.6	0
0.15	100	77.6	0
0.17	100	96.8	0
0.18	100	100	0

Table 4.3 For Minimum score level fusion (in %)

Threshold	GAR	FAR	FRR
0.05	0.4	0	99.6
0.07	10.2	0	89.8
0.08	29.6	0	70.4
0.09	49.6	0	50.4
0.11	81.2	0	18.8
0.13	93.6	3	6.4
0.15	99.2	27.2	0.8
0.16	100	47	0
0.18	100	93.4	0
0.19	100	100	0

Table 4.4 For Maximum score level fusion (in %)

Threshold	GAR	FAR	FRR
0.05	1.6	0	98.4
0.07	23	0	77
0.09	70	0.4	30
0.11	96.4	2.6	3.6
0.13	99.8	12.8	0.2
0.15	100	51.6	0
0.16	100	78	0
0.17	100	94.2	0
0.19	100	100	0

Table 4.5 For Weighted sum rule (in %)

4.2.1 Region of Characteristics Curve for the whole system

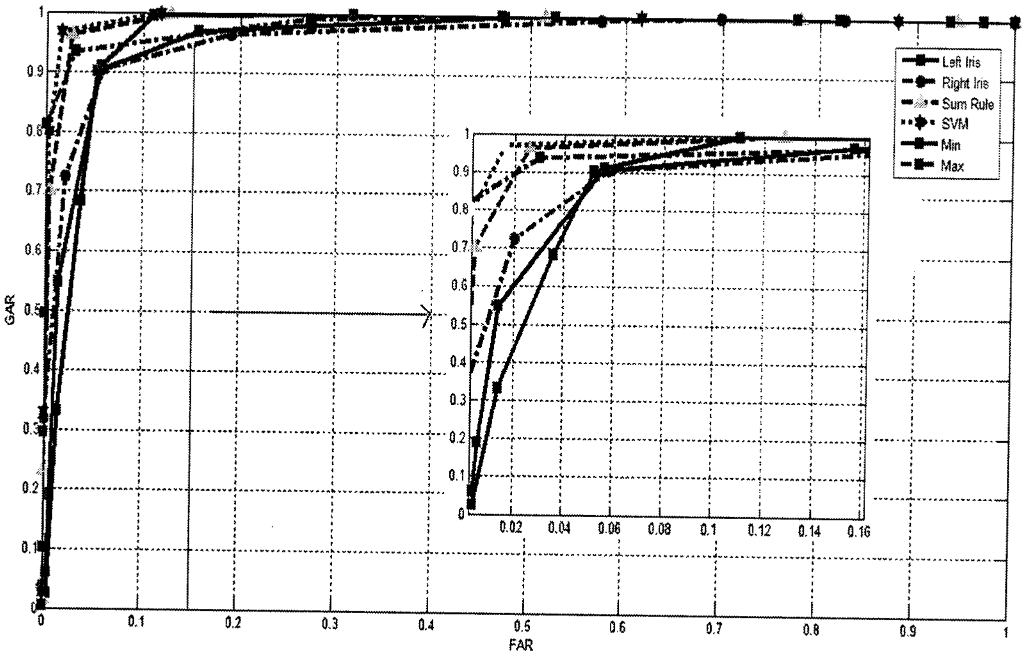


Fig.4.17 ROC against GAR and FAR

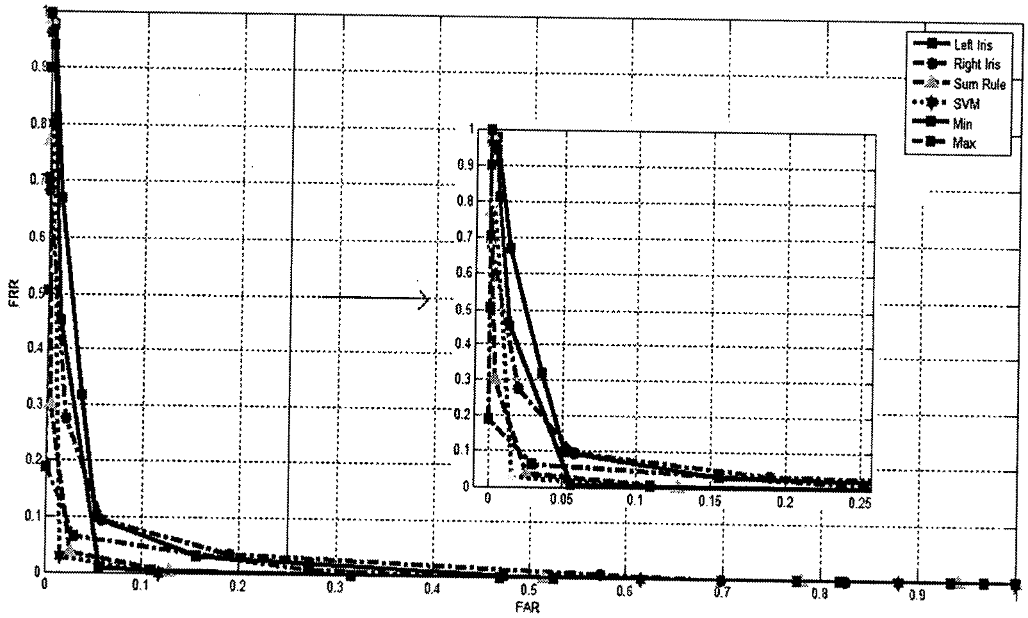


Fig.4.18 ROC against FAR and FRR

4.2.2 Performance analysis of various fusion technologies

Method	Recognition Rate	EER	Time Taken
Right Iris	88%	0.06%	21.3297
Left Iris	90.4%	0.051%	21.9515
Fusion using Min Rule	90.6%	0.048%	43.2865
Fusion using Max Rule	93.2%	0.03%	43.2865
Fusion using Sum Rule	96.4%	0.025%	43.2861
Fusion using SVM	97.89%	0.02%	43.2875

Table 4.6 Represent the comparison of all methods

CHAPTER 5

CONCLUSION AND FUTURE OUTLOOK

Thus a new multi-unit iris recognition system is proposed which has better recognition rate than the previous approaches. By considering the best bits in an iris code the storage size is reduced and also the time taken for recognition is less. Out of the fusion technologies deployed SVM performs better than the minimum score, maximum score and weighted sum rule methods. The error rate is reduced significantly. The time taken for fusion is insignificant when compared to the individual left and right iris recognition system. In future, real time images will be captured and then the authentication is made. Several fusion technologies have been deployed and then performance may be increased.

APPENDIX

Source code for iris recognition system

Creating Iris template main file

```
function [template, mask] = createiristemplate(eyeimage_filename)
 % path for writing diagnostic images
global DIAGPATH
DIAGPATH = 'diagnostics';
 %normalisation parameters
radial_res = 20;
angular_res = 240;
'%feature encoding parameters
nscales=1;
minWaveLength=18;
mult=1; % not applicable if using nscales = 1
sigmaOnf=0.5;
eyeimage = imread(eyeimage_filename);
 savefile = [eyeimage_filename,'-houghpara.mat'];
 [stat,mess]=fileattrib(savefile);
 if stat == 1
   % if this file has been processed before then load the circle      %parameters and noise
information for that file.
   load(savefile);
 else
  % if this file has not been processed before then perform automatic %segmentation and
save the results to a file
   [circleiris circlepupil imagewithnoise] = segmentiris(eyeimage);
    save(savefile,'circleiris','circlepupil','imagewithnoise');
   end
 % WRITE NOISE IMAGE
imagewithnoise2 = uint8(imagewithnoise);
imagewithcircles = uint8(eyeimage);
 %get pixel coords for circle around iris
[x,y] = circlecoords([circleiris(2),circleiris(1)],circleiris(3),size(eyeimage));
ind2 = sub2ind(size(eyeimage),double(y),double(x));
 %get pixel coords for circle around pupil
[xp,yp] = circlecoords([circlepupil(2),circlepupil(1)],circlepupil(3),size(eyeimage));
ind1 = sub2ind(size(eyeimage),double(yp),double(xp));
 % Write noise regions
imagewithnoise2(ind2) = 255;
imagewithnoise2(ind1) = 255;
% Write circles overlayed
imagewithcircles(ind2) = 255;
imagewithcircles(ind1) = 255;
w = cd;
cd(DIAGPATH);
imwrite(imagewithnoise2,[eyeimage_filename,'-noise.jpg'],'jpg');
imwrite(imagewithcircles,[eyeimage_filename,'-segmented.jpg'],'jpg');
cd(w);
```

```
 % perform normalisation
[polar_array noise_array] = normaliseiris(imagewithnoise, circleiris(2),...
   circleiris(1), circleiris(3), circlepupil(2), circlepupil(1), circlepupil(3),eyeimage_filename,
radial_res, angular_res);
% WRITE NORMALISED PATTERN, AND NOISE PATTERN
w = cd;
cd(DIAGPATH);
imwrite(polar_array,[eyeimage_filename,'-polar.jpg'],'jpg');
imwrite(noise_array,[eyeimage_filename,'-polarnoise.jpg'],'jpg');
cd(w);
% perform feature encoding
 [template mask] = encode(polar_array, noise_array, nscales, minWaveLength, mult,
sigmaOnf);
```

Segmentation

```
function [circleiris, circlepupil, imagewithnoise] = segmentiris(eyeimage)
% define range of pupil & iris radii
%CASIA
lpupilradius = 28;
upupilradius = 75;
lirisradius = 80;
uirisradius = 150;
% define scaling factor to speed up Hough transform
scaling = 0.4;
reflecthres = 240;
 % find the iris boundary
[row, col, r] = findcircle(eyeimage, lirisradius, uirisradius, scaling, 2, 0.20, 0.19, 1.00, 0.00);
 circleiris = [row col r];
 rowd = double(row);
cold = double(col);
rd = double(r);
 irl = round(rowd-rd);
iru = round(rowd+rd);
icl = round(cold-rd);
icu = round(cold+rd);
 imgsize = size(eyeimage);
 if irl < 1
   irl = 1;
end
 if icl < 1
   icl = 1;
end
 if iru > imgsize(1)
   iru = imgsize(1);
end
 if icu > imgsize(2)
   icu = imgsize(2);
end
 % to find the inner pupil, use just the region within the previously detected iris boundary
imagepupil = eyeimage( irl:iru,icl:icu);
```

```
 %find pupil boundary
[rowp, colp, r] = findcircle(imagepupil, lpupilradius, upupilradius ,0.6,2,0.25,0.25,1.00,1.00);
 rowp = double(rowp);
colp = double(colp);
r = double(r);
 row = double(irl) + rowp;
col = double(icl) + colp;
 row = round(row);
col = round(col);
 circlepupil = [row col r];
 % set up array for recording noise regions noise pixels will have NaN values
imagewithnoise = double(eyeimage);
 %find top eyelid
topeyelid = imagepupil(1:(rowp-r),:);
lines = findline(topeyelid);
 if size(lines,1) > 0
    [xl yl] = linecoords(lines, size(topeyelid));
    yl = double(yl) + irl-1;
    xl = double(xl) + icl-1;
     yla = max(yl);
     y2 = 1:yla;
     ind3 = sub2ind(size(eyeimage),yl,xl);
    imagewithnoise(ind3) = NaN;
    imagewithnoise(y2, xl) = NaN;
end
 %find bottom eyelid
bottomeyelid = imagepupil((rowp+r):size(imagepupil,1),:);
lines = findline(bottomeyelid);
 if size(lines,1) > 0
      [xl yl] = linecoords(lines, size(bottomeyelid));
      yl = double(yl)+ irl+rowp+r-2;
      xl = double(xl) + icl-1;
       yla = min(yl);
       y2 = yla:size(eyeimage,1);
       ind4 = sub2ind(size(eyeimage),yl,xl);
    imagewithnoise(ind4) = NaN;
    imagewithnoise(y2, xl) = NaN;
 end
 %For CASIA, eliminate eyelashes by thresholding
ref = eyeimage < 100;
coords = find(ref==1);
imagewithnoise(coords) = NaN;
function [gradient, or] = canny(im, sigma, scaling, vert, horz)
xscaling = vert;
yscaling = horz;
hsize = [6*sigma+1, 6*sigma+1];   % The filter size.
gaussian = fspecial('gaussian',hsize,sigma);
im = filter2(gaussian,im);        % Smoothed image.
im = imresize(im, scaling);
[rows, cols] = size(im);
```

```
h = [ im(:,2:cols) zeros(rows,1) ] - [ zeros(rows,1) im(:,1:cols-1) ];
v = [ im(2:rows,:); zeros(1,cols) ] - [ zeros(1,cols); im(1:rows-1,:) ];
d1 = [ im(2:rows,2:cols) zeros(rows-1,1); zeros(1,cols) ] - ...
                    [ zeros(1,cols); zeros(rows-1,1) im(1:rows-1,1:cols-1) ];
d2 = [ zeros(1,cols); im(1:rows-1,2:cols) zeros(rows-1,1); ] - ...
                    [ zeros(rows-1,1) im(2:rows,1:cols-1); zeros(1,cols) ];
X = ( h + (d1 + d2)/2.0 ) * xscaling;
Y = ( v + (d1 - d2)/2.0 ) * yscaling;
gradient = sqrt(X.*X + Y.*Y); % Gradient amplitude.
or = atan2(-Y, X);          % Angles -pi to + pi.
neg = or<0;                 % Map angles to 0-pi.
or = or.*~neg + (or+pi).*neg;
or = or*180/pi;             % Convert to degrees.
```

Normalization

```
function [polar_array, polar_noise] = normaliseiris(image, x_iris, y_iris, r_iris, x_pupil,
y_pupil, r_pupil,eyeimage_filename, radpixels, angulardiv)
global DIAGPATH
radiuspixels = radpixels + 2;
angledivisions = angulardiv-1;
r = 0:(radiuspixels-1);
theta = 0:2*pi/angledivisions:2*pi;
x_iris = double(x_iris);
y_iris = double(y_iris);
r_iris = double(r_iris);
x_pupil = double(x_pupil);
y_pupil = double(y_pupil);
r_pupil = double(r_pupil);
% calculate displacement of pupil center from the iris center
ox = x_pupil - x_iris;
oy = y_pupil - y_iris;
if ox <= 0
   sgn = -1;
elseif ox > 0
   sgn = 1;
end
if ox==0 && oy > 0
   sgn = 1;
end
 r = double(r);
theta = double(theta);
 a = ones(1,angledivisions+1)* (ox^2 + oy^2);
 % need to do something for ox = 0
if ox == 0
   phi = pi/2;
else
   phi = atan(oy/ox);
end
 b = sgn.*cos(pi - phi - theta);
```

```
% calculate radius around the iris as a function of the angle
r = (sqrt(a).*b) + ( sqrt( a.*(b.^2) - (a - (r_iris^2))));
r = r - r_pupil;
rmat = ones(1,radiuspixels)'*r;
rmat = rmat.* (ones(angledivisions+1,1)*[0:1/(radiuspixels-1):1])';
rmat = rmat + r_pupil;
% exclude values at the boundary of the pupil iris border, and the iris scelra border  as these
%may not correspond to areas in the iris region and will introduce noise.
% ie don't take the outside rings as iris data.
rmat  = rmat(2:(radiuspixels-1), :);
% calculate cartesian location of each data point around the circular iris region
xcosmat = ones(radiuspixels-2,1)*cos(theta);
xsinmat = ones(radiuspixels-2,1)*sin(theta);
xo = rmat.*xcosmat;
yo = rmat.*xsinmat;
xo = x_pupil+xo;
yo = y_pupil-yo;
% extract intensity values into the normalised polar representation through  interpolation
[x,y] = meshgrid(1:size(image,2),1:size(image,1));
polar_array = interp2(x,y,image,xo,yo);
% create noise array with location of NaNs in polar_array
polar_noise = zeros(size(polar_array));
coords = find(isnan(polar_array));
polar_noise(coords) = 1;
polar_array = double(polar_array)./255;
% start diagnostics, writing out eye image with rings overlayed get rid of outling points in
%order to write out the circular pattern
coords = find(xo > size(image,2));
xo(coords) = size(image,2);
coords = find(xo < 1);
xo(coords) = 1;
coords = find(yo > size(image,1));
yo(coords) = size(image,1);
coords = find(yo<1);
yo(coords) = 1;
xo = round(xo);
yo = round(yo);
xo = int32(xo);
yo = int32(yo);
ind1 = sub2ind(size(image),double(yo),double(xo));
image = uint8(image);
image(ind1) = 255;
%get pixel coords for circle around iris
[x,y] = circlecoords([x_iris,y_iris],r_iris,size(image));
ind2 = sub2ind(size(image),double(y),double(x));
%get pixel coords for circle around pupil
[xp,yp] = circlecoords([x_pupil,y_pupil],r_pupil,size(image));
ind1 = sub2ind(size(image),double(yp),double(xp));
image(ind2) = 255;
image(ind1) = 255;
```

```
% write out rings overlaying original iris image
w = cd;
cd(DIAGPATH);
imwrite(image,[eyeimage_filename,'-normal.jpg'],'jpg');
cd(w);
% end diagnostic\s
%replace NaNs before performing feature encoding
coords = find(isnan(polar_array));
polar_array2 = polar_array;
polar_array2(coords) = 0.5;
avg = sum(sum(polar_array2)) / (size(polar_array,1)*size(polar_array,2));
polar_array(coords) = avg;
```

Encode

```
function [template, mask] = encode(polar_array,noise_array, nscales, minWaveLength, mult,
sigmaOnf)
% convolve normalised region with Gabor filters
[E0 filtersum] = gaborconvolve(polar_array, nscales, minWaveLength, mult, sigmaOnf);
length =size(polar_array,2)*2*nscales;
template = zeros(size(polar_array,1), length);
length2 = size(polar_array,2);
h =5:12;%1:size(polar_array,1);
%create the iris template
mask = zeros(size(template));
for k=1:nscales
   E1 = E0{k};
   %Phase quantisation
   H1 = real(E1) > 0;
   H2 = imag(E1) > 0;
   % if amplitude is close to zero then  phase data is not useful, so mark off in the noise mask
   H3 = abs(E1) < 0.0001;
   for i= 4:(length2-1) %0:(length2-1)
ja = double(2*nscales*(i));
%construct the biometric template
 template(h,ja+(2*k)-1) = H1(h, i+1);
 template(h,ja+(2*k)) = H2(h,i+1);
  %create noise mask
  mask(h,ja+(2*k)-1) = noise_array(h, i+1) | H3(h, i+1);
  mask(h,ja+(2*k)) =  noise_array(h, i+1) | H3(h, i+1);
   end
end
function [EO, filtersum] = gaborconvolve(im, nscale, minWaveLength, mult, ...
   sigmaOnf)
[rows cols] = size(im);
filtersum = zeros(1,size(im,2));
EO = cell(1, nscale);         % Pre-allocate cell array
ndata = cols;
if mod(ndata,2) == 1            % If there is an odd No of data points
   ndata = ndata-1;             % throw away the last one.
```

```
end
logGabor  = zeros(1,ndata);
result = zeros(rows,ndata);
radius =  [0:fix(ndata/2)]/fix(ndata/2)/2;  % Frequency values 0 - 0.5
radius(1) = 1;
wavelength = minWaveLength;        % Initialize filter wavelength
for s = 1:nscale,                  % For each scale.
   % Construct the filter - first calculate the radial filter component.
   fo = 1.0/wavelength;                 % Centre frequency of filter.
   rfo = fo/0.5;                  % Normalised radius from centre of frequency plane
   % corresponding to fo.
   logGabor(1:ndata/2+1) = exp((-(log(radius/fo)).^2) / (2 * log(sigmaOnf)^2));
   logGabor(1) = 0;
   filter = logGabor;
   filtersum = filtersum+filter;
   % for each row of the input image, do the convolution, back transform
   for r = 1:rows  % For each row
      signal = im(r,1:ndata);
      imagefft = fft( signal );
      result(r,:) = ifft(imagefft .* filter);
   end
   % save the ouput for each scale
   EO{s} = result;
   wavelength = wavelength * mult;       % Finally calculate Wavelength of next filter
end                             % ... and process the next scale
filtersum = fftshift(filtersum);
```

Enhancement

```
function [out,T] = histeq(a,cm,hgram)
iptchecknargin(1,3,nargin,mfilename);
 NPTS = 256;
 if nargin==1 % Histogram equalization of intensity image
   n = 64; % Default n
   hgram = ones(1,n)*(numel(a)/n);
   n = NPTS;
   kind = 1;
elseif nargin==2
   if numel(cm)==1
     % histeq(I,N); Histogram equalization of intensity image
     iptcheckinput(a,{'uint8','uint16','double','int16','single'}, ...
        {'nonsparse','2d'}, mfilename,'I',1);
     m = cm;
     hgram = ones(1,m)*(numel(a)/m);
     n = NPTS;
     kind = 1;
   elseif size(cm,2)==3 && size(cm,1)>1
     % histeq(X,map); Histogram equalization of indexed image
     iptcheckinput(a,{'uint8','double','uint16','single'}, ...
```

```
      {'nonsparse','2d'},mfilename,'X',1);
    if isa(a, 'uint16')
      msg = 'Histogram equalization of UINT16 indexed images is not supported.';
      eid = sprintf('Images:%s:unsupportedUint16IndexedImages',mfilename);
      error(eid, msg);
    end
    n = size(cm,1);
    hgram = ones(1,n)*(numel(a)/n);
    kind = 2;
  else
    % histeq(I, HGRAM); Histogram modification of intensity image
    iptcheckinput(a,{'uint8','uint16','double','int16','single'}, ...
      {'nonsparse','2d'}, mfilename,'I',1);
    hgram = cm;
    n = NPTS;
    kind = 1;
  end
else  % Histogram modification of indexed image
  iptcheckinput(a,{'uint8','double','uint16','single'}, ...
    {'nonsparse','2d'},mfilename,'X',1);
  if isa(a, 'uint16')
    msg = 'Histogram equalization of UINT16 indexed images is not supported.';
    eid = sprintf('Images:%s:unsupportedUint16IndexedImages',mfilename);
    error(eid, msg);
  end
  n = size(cm,1);
  if length(hgram)~=n
    msg = 'HGRAM must be the same size as MAP.';
    eid = sprintf('Images:%s:HGRAMmustBeSameSizeAsMAP',mfilename);
    error(eid, msg);
  end
  kind = 2;
end
 classChanged = false;
if isa(a,'int16')
  classChanged = true;
  a = im2uint16(a);
end
 if min(size(hgram))>1
  msg = 'HGRAM must be a vector.';
  eid = sprintf('Images:%s:hgramMustBeAVector',mfilename);
  error(eid, msg);
end
 % Normalize hgram
hgram = hgram*(numel(a)/sum(hgram));       % Set sum = numel(a)
m = length(hgram);
 % Compute cumulative histograms
if kind==1,
  nn = imhist(a,n)';
  cum = cumsum(nn);
```

```
else
  % Convert image to equivalent gray image
  I = ind2gray(a,cm);
  nn = imhist(I,n)';
  cum = cumsum(nn);
end
cumd = cumsum(hgram*numel(a)/sum(hgram));
 % Create transformation to an intensity image by minimizing the error  between desired and
%actual cumulative histogram.
tol = ones(m,1)*min([nn(1:n-1),0;0,nn(2:n)])/2;
err = (cumd(:)*ones(1,n)-ones(m,1)*cum(:)')+tol;
d = find(err < -numel(a)*sqrt(eps));
if ~isempty(d)
  err(d) = numel(a)*ones(size(d));
end
[dum,T] = min(err); %#ok
T = (T-1)/(m-1);
 if kind == 1 % Modify intensity image
  b = grayxform(a, T);
else % Modify colormap by extending the (r,g,b) vectors.
 % Compute equivalent colormap luminance
  ntsc = rgb2ntsc(cm);
   % Map to new luminance using T, store in 2nd column of ntsc.
  ntsc(:,2) = T(floor(ntsc(:,1)*(n-1))+1)';
   % Scale (r,g,b) vectors by relative luminance change
  map = cm.*((ntsc(:,2)./max(ntsc(:,1),eps))*ones(1,3));
   % Clip the (r,g,b) vectors to the unit color cube
  map = map ./ (max(max(map')',1)*ones(1,3));
end
 if nargout==0,
  if kind==1
    imshow(b);
  else
    imshow(a,map),
  end
  return
end
 if kind==1
  if classChanged
   out = im2int16(b);
  else
    out = b;
  end
else
  out = map;
end
```

Matching

```
function hd = gethammingdistance(template1, mask1, template2, mask2, scales)
 template1 = logical(template1);
mask1 = logical(mask1);
 template2 = logical(template2);
mask2 = logical(mask2);
 hd = NaN;
 % shift template left and right, use the lowest Hamming distance
for shifts=-8:8
   template1s = shiftbits(template1, shifts,scales);
   mask1s = shiftbits(mask1, shifts,scales);
   mask = mask1s | mask2;
   nummaskbits = sum(sum(mask == 1));
   totalbits = (size(template1s,1)*size(template1s,2)) - nummaskbits;
   C = xor(template1s,template2);
  C = C & ~mask;
   bitsdiff = sum(sum(C==1));
  if totalbits == 0
      hd = NaN;
   else
      hd1 = bitsdiff / totalbits;
             if  hd1 < hd || isnan(hd)
           hd = hd1;
       end
end
end
function templatenew = shiftbits(template, noshifts,nscales)
templatenew = zeros(size(template));
width = size(template,2);
s = round(2*nscales*abs(noshifts));
p = round(width-s);
if noshifts == 0
   templatenew = template;
      % if noshifts is negatite then shift towards the left
elseif noshifts < 0
      x=1:p;
     templatenew(:,x) = template(:,s+x);
      x=(p + 1):width;
      templatenew(:,x) = template(:,x-p);
  else
     x=(s+1):width;
      templatenew(:,x) = template(:,x-s);
      x=1:s;
      templatenew(:,x) = template(:,p+x);
   end
```

SVM training

```
function [alphaStar, bStar, SVIndex] = yxcSVMtrain(X, Y, C, kernel, sigma)
[num, dim] = size(X);
if dim ~= 2
   return;
end
if num ~= length(Y)
   return;
end
Y = Y(:);
H = (Y*Y').*yxcSVMkernel(X, X, kernel, sigma);
f = -ones(num, 1);
A = zeros(1, num);
b = 0;
Aeq = Y';
beq = 0;
lb = zeros(num, 1);
ub = C .* ones(num, 1);
x0 = zeros(num,1);
%qp_options = optimset('Display','off');
qp_options = optimset('MaxIter',10^3, 'LargeScale', 'off', 'Display','off');
[alphaStar,fval,exitflag,output] = quadprog(H,f,[],[],Aeq,beq,lb,ub,x0,qp_options);
nearZero = 10^-12;
% Assume the minor ones are all zero.
alphaStar(find(abs(alphaStar) < nearZero)) = 0;
alphaStar(find(alphaStar > C - nearZero)) = C;
% support vectors are those whose alpha value > 0
SVIndex = find(alphaStar > 0);
% support vectors on bound are those whose alpha value == max(alphaStar) == C
SVNotOnBoundIndex = find(alphaStar > 0 & alphaStar < max(alphaStar));
% bStar is an average value, not a random one.
if ~isempty(SVNotOnBoundIndex)
   bStar = sum( Y(SVNotOnBoundIndex) - H(SVNotOnBoundIndex, SVIndex) *
alphaStar(SVIndex) .* Y(SVNotOnBoundIndex) )...
            / length(SVNotOnBoundIndex);
else
   bStar = 0;
end
```

SVM classifier

```
function [YClassified, Z, err] = yxcSVMclassifer(Xtrain, Xtest, Y, alphaStar, bStar, kernel,
sigma)
[num, dim] = size(Xtrain);
if dim ~= 2
   return;
end
if num ~= length(Y)
   return;
end
```

```
Y = Y(:);
alphaStar = alphaStar(:);
% Algorithm 5.4.12, step (4)
H = yxcSVMkernel(Xtrain, Xtest, kernel, sigma)';
Z = H * (alphaStar .* Y);
Z = Z + bStar;
YClassified(find(Z > 0)) = 1;
YClassified(find(Z <= 0)) = -1;
YClassified = YClassified';
if(size(Xtest) == size(Xtrain))
   if(Xtest == Xtrain)
      err = length(find(YClassified ~= Y)) / length(Y);
   end
else
   err = NaN;
end
function K = yxcSVMkernel(a, b, kernel, sigma)
[n1, dim1] = size(a);
[n2, dim2] = size(b);
if dim1 ~= dim2
   error('columns of a and b must agree');
end
switch kernel
   case 'linear'
      K = a*b';
   case 'poly'
      K = (a * b' + 1).^sigma;
   case 'rbf'
      [N1, d] = size(a);
      [N2, d] = size(b);
      dist2 = repmat(sum((a.^2)', 1), [N2 1])' + ...
            repmat(sum((b.^2)',1), [N1 1]) - 2*a*(b');
      K = exp(-dist2/(sigma^2));
   otherwise
      error('kernel type must be specified.');
end
function yxcSVMplot(X, Y, SVIndex, alphaStar, bStar, kernel, sigma, plotAxis)
[num, dim] = size(X);
if dim ~= 2
   return;
end
if num ~= length(Y)
   return;
end
Y = Y(:);
plusIndex = find(Y==1);
minusIndex = find(Y==-1);
figure;
plot(X(plusIndex, 1), X(plusIndex, 2), 'r+');
if nargin == 8
```

```
  axis(plotAxis);
end
hold on;
plot(X(minusIndex, 1), X(minusIndex, 2), 'b.');
hold on;
plot(X(SVIndex, 1), X(SVIndex, 2), 'o');
hold on;
 % plot the decision isoline
[cx, cy] = meshgrid(min( X(:, 1)) : .1 : max(X(:, 1)), min( X(:, 2)) : .1 : max(X(:, 2)));
[ncx, ncy] = size(cx);
 Z = zeros(ncx, ncy);
[cx, cy] = meshgrid(min( X(:, 1)) : .1 : max(X(:, 1)), min( X(:, 2)) : .1 : max(X(:, 2)));
[ncx, ncy] = size(cx);
Xtest = [cx(:) cy(:)];
[YClassified, Z]=yxcSVMclassifer(X, Xtest, Y, alphaStar, bStar, kernel, sigma);
% reshape it.
Z = reshape(Z, ncx, ncy);
% plot the decision isoline
[C,h] = contour(cx, cy, Z, [0,0]);
set(h,'ShowText','on','TextStep',get(h,'LevelStep'));
hold on;
[C,h] = contour(cx, cy, Z, [1,1]);
set(h,'ShowText','on','TextStep',get(h,'LevelStep'));
hold on;
[C,h] = contour(cx, cy, Z, [-1,-1]);
set(h,'ShowText','on','TextStep',get(h,'LevelStep'));
```

REFERENCES

[1] Karen P. Hollingsworth , Kevin W. Bowyer, Patrick J. Flynn, " The Best Bits in an Iris Code", IEEE Trans. Pattern Analysis and Machine Intelligence Vol. 31 no. 6, pp. 964-973, June 2009.

[2] Byung Jun Kang, Kang Ryoung Park, "A new multi-unit iris authentication based on quality assessment and score level fusion for mobile phones", Dec 2008, Springer

[3] P. Padma Polash and M. Maruf Monwar, "Human iris recognition for biometric identification", IEEE 10th international conference on Computer and information technology, pp. 1-5, 2007

[4] "CASIA Iris Image Database", http://www.sinobiometrics.com, 2007.

[5] Rafael C. Gonzalez and Richard E. Woods eds, "Digital image processing", Low Price Edition 2nd edition 2003.

[6] V.C. Subbarayudu and Munaga VNK Prasad , " Multimodal Biometric System", IEEE First International Conference on Emerging Trends in Engineering and Technology, 2008.

[7] Sudha.N, Puhan. N.B, Xia. H, Jiang.X, " Iris recognition on edge maps", IEEE Transaction on Computer Vision,Vol.3, pp. 1-7, 2009.

[8] A. Ross, A.K. Jain, "Information Fusion in Biometrics", Pattern Recognition Letters, vol. 24, No1, pp. 2115.2125,2003

[9] Sonia Garcia Salicetti, Mohamed Anouar Mellakh, Lorene Allano, Bernadette Dorizzi, "Multimodal Biometric Score Fusion : The Mean Rule Vs. Support Vector Classifiers", 2008, Springer.

[10] R. T. Al-Zubi, D. I. Abu-Al-Nadi, "Automated personal identification system based on human iris analysis", Nov 2006, Springer.

[11] Daugman JG (2002), " How iris recognition works", In: Proceedings of 2002 international conference on image processing, vol 1.

[12] Wildes RP, Asmuth JC, Hanna JK, Hsu SC, Kolezynski RJ, Matey JR, McBride SE, "Automated, non-invasive iris recognition system and method", Patent no.5,572,596,1996.

[13] Lim S, Lee K, Byeon O, Kim T, "Efficient iris recognition through improvement of feature vector and classifier", ETRI J 23(2):61–70, 2001.

[14] Ma L, Wang Y, Tan T, Zhang D, "Personal identification based on iris texture analysis", IEEE Transaction on Pattern Analysis and Machine Intelligence 25(12):1519–1533, 2003.

[15] Boles WW, Boashash B , "A human identification technique using images of the iris and wavelet transform", IEEE Trans Signal Process 46(4):1185–1188, 1998.

[16] Zhu Y, Tan T, Wang Y, "Biometric personal identification based on iris patterns", Proceedings of International Conference on Pattern Recognition, 2:805–808, 2000.

[17] Tisse C, Martin L, Torres L, Robert M, "Person identification technique using human iris recognition", Proc Vis Interface 294–299, 2002.

[18] Ma L, Wang Y, Tan T, "Iris recognition based on multichannel Gabor filtering", In: Proceedings of ACCV'2002: the 5th Asian conference on computer vision, 23–25 January, Melbourne, 2002.

[19] Ma L, Wang Y, Tan T, "Iris recognition using circular symmetric filters", Proceedings of International Conference on Pattern Recognition 2:414–417, 2002.

[20] Sabra Dinerstein, Jonathan Dinerstein, and Dan Ventura, "Robust Multi-Modal Biometric Fusion via Multiple SVMs", IEEE in Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), vol. 04, pp. 473-476, 2006.

[21] Jain, J., Son, J., Lee, Y., Park, K.R, “Multi-unit iris recognition system by image check algorithm”, In: Lecture Notes in Computer Science on ICBA2004, vol. 3072, pp. 450–457 2004.

[22] Robert Snelick, Umut Uludag, Alan Mink, Michael Indovina, Anil Jain, “Large-scale evaluation of multimodal biometric authentication using state-of-the art systems”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.27, No.3, March 2005.

[23] J.Kittler, M.Hatef, R.P.W.Duin, and J.Matas, “On Combining Classifiers”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.20, No.3, pp.226-239,March 1998.

[24] S.Ben-Yacoub, Y.Abdeljaoued, and E.Mayoraz, “Fusion of face and speech data for person identity verification”, IEEE Transactions on Neural networks, Vol.10, No.5, pp.1065-1075, 1999.