An Efficient and Optimal IRIS Recognition System using MATLAB GUI

Sachin Kumar
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
sachindhankar53@gmail.com

Govindam Sharma
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
ergovindamsharma@gmail.com

Nikhil Arora
Department of Electronics Engineering,
YMCA University, Faridabad, INDIA
nikhil_arora5@yahoo.com

Abstract— A biometric system is used for recognition of individual based on their physical or personal characteristics. Biometric system includes face recognition, fingerprint recognition, voice recognition , the Iris recognition etc. various study has shown that iris recognition is the most efficient biometrics. hence the work presented here involved designing a user friendly GUI based efficient and optimal iris recognition system using MATLAB® GUI. So that one with least knowledge of technology can use it. in order to generate the base templates of iris, we have used Masek and Kovesi's algorithm with some necessary changes. We have used the Image processing toolbox and GUIDE toolbox of MATLAB, to make the GUI for iris recognition system. There are so many methods to design an Iris recognition system having their own pros and cons. Some methods are calculation intensive but they lead in performance while other are less calculation intensive but they lack in performance. To design the iris recognition system we have focused on both the sides i.e. calculation intensity and performance, to make the system efficient and optimal.

In order to use this GUI based iris recognition system first, one need to just select an input eye image that one want to recognize from the iris image database and then just click on recognize button in GUI. now you are done. After the recognition process is complete it shows all the results related to that particular recognition process. like the name or number of the recognized person, segmented eye image, noise template etc and MATLAB's output window shows the hamming distance related to the matching process of the recognition system. We have used the IIT Delhi's iris image dataset for the verification and testing of our GUI based iris recognition system.

Keywords- *Biometrics, Iris recognition, Automatic segmentation, Iris Template, Pattern recognition, Hamming distance, MATLAB GUI*

*****

I. INTRODUCTION

Personal identification plays a very important role in our daily life. Whether we need to withdraw money from our bank account or we need to access our social profile, email, and to use internet-banking and for the recognition process at airports in some countries and so on. So we need a strong, efficient and optimal personal identification system. Biometrics can play a important role in personal identification. Biometrics includes thumb impression, fingerprints, face, voice, signature, keystroke, palm structure, hand geometry, DNA, retina, ear structure, gate identification, and iris.

A Biometrics system is used for personal identification of individual based on their physical or personal characteristics. Biometric systems have been developed based on fingerprints, facial features, voice, hand geometry, handwriting, the retina [1] and the one we designed here, the Iris.

Iris recognition is a method of biometric authentication that uses pattern-recognition techniques based on high resolution Iris images of the eyes of an individual [2]. Pioneer work in iris recognition was proposed by Daugman [3]. Daugman's algorithm forms the basis of today's commercially used iris recognition systems.

In order to generate the base templates of iris, we have used Masek [5] and Kovesi's algorithm [6] with some necessary alteration and the Image processing toolbox and GUIDE toolbaox of matlab, to make the GUI based iris recognition system. The algorithm is based primarily on the methods given by Daugman [4].

The process of identifying and recognizing people using the IRIS goes through the following steps:

1. Iris image Acquisition
2. Pre Processing
3. Segmentation
4. Normalization
5. Feature Extraction
6. Matching
7. Decision

The proposed iris recognition system consists of an automatic segmentation system that is based on the Circular Hough transform, and is able to localise the circular iris and pupil region, occluding eyelids and eyelashes, and reflections. The extracted iris region was then normalised into a rectangular block with constant dimensions to account for imaging inconsistencies. Finally, the phase data from 1D Log-Gabor filters was extracted and quantised to four levels to encode the unique pattern of the iris into a bit-wise biometric template [5]. The Hamming distance was employed for classification of iris templates, and two templates were found to match if a test of statistical independence was failed [5]. The system performed with perfect recognition on a set of 224 eye images.

II. EYE STRUCTURE

The iris is a thin circular structure, in the form of a unique texture which lies between the cornea and the lens of the human eye. A front view of an eye image is shown in Figure The iris is close to its centre by a circular aperture known as the pupil. The function of the iris is to control the amount of light entering through the pupil, and this is done by the sphincter and the dilator muscles, which adjust the size of the pupil. The average diameter of the iris is 12 mm, and the pupil size can vary from 10% to 80% of the iris diameter [2].

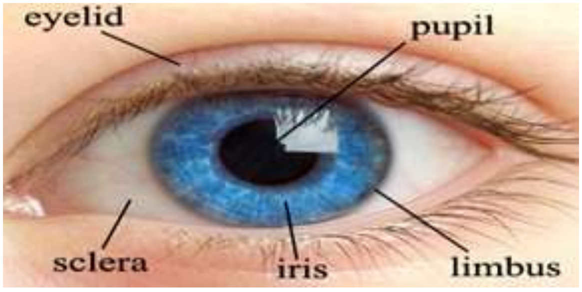

Figure 1. A front view of an Eye image

Iris characteristics— Iris texture is a very unique characteristic possess by every human being. As every individual iris has highly distinguishable texture. And it is formed in third month and completed in eight month of fetal term. Every individual person even twins have different iris textures. moreover left and right eye also have different iris pattern. And changing iris pattern without surgery with high risk is impossible. In addition iris recognition system is a noninvasive method, means it does not disturb or invade on anyone's privacy like DNA does.

III. ALGORITHM: IRIS RECOGNITION

As already discussed that an iris recognition system algorithm includes these seven stages described below:

1. Iris Image Acquisition

The Iris image acquisition system includes a special camera to capture high quality iris image of individual's eye. Eye image should be captured by using a specialized camera, typically very close to the subject, not more than three feet, that uses an infrared imager to illuminate the eye and capture a very high-resolution photograph. The infrared imager used to reduce the reflection's effect in an eye image. This process takes 1 to 2 seconds. But here our main focus was on software performance and to make a GUI. So we have used already captured IIT Delhi's iris eye image data set of 224 grey scale images.

2. Preprocessing

The proposed iris recognition system works on grey scale images. But if you have a RGB or other than grey scale iris image dataset. Or your image acquisition system provides you the image other than grey scale images then you have to change that image into grey scale image. And one more thing is to store the mask and templates of the iris image dataset[5], before using the iris recognition system, to compare the input test image in a one to many fashion with already saved mask and templates of iris image dataset.

3. Segmentation

The next stage of iris recognition system is to isolate the actual iris region from a digital eye image. The iris region, shown in Figure 1, can be separated by two circles, one for the iris/sclera boundary and another, interior to the first, by the iris/pupil boundary. The eyelids and eyelashes normally occlude the upper and lower parts of the iris region. And specular reflections can also occur within the iris region results in corrupting the digital iris pattern. So a technique is required to remove and exclude these unwanted interference as well as locating the circular iris region.

The success of segmentation stage depends on the image quality of eye images. So the iris database should be created by using near infrared light. To avoid specular reflections. That occurs in imaging under natural light and results in corrupting the iris pattern.

Some important segmentation techniques are:

1. Hough transform
2. Daugman's integro-differential operator
3. Active contour models
4. Eyelash and noise detection

Here we have used the circular hough transform as Daugman's system is the most successful and most well known [5], many other systems have been developed. The most notable include the systems of Wildes et al. [7][8], and Hsiung et al. [9]. presented in this paper the best iris segmentation technique for iris recognition is Hough Transform. The results show that Hough Transform techniques capable to recognize the low resolution iris with the accuracy of 100% compared to the Daugman' s Integro Differential Operator with only 86.88%.

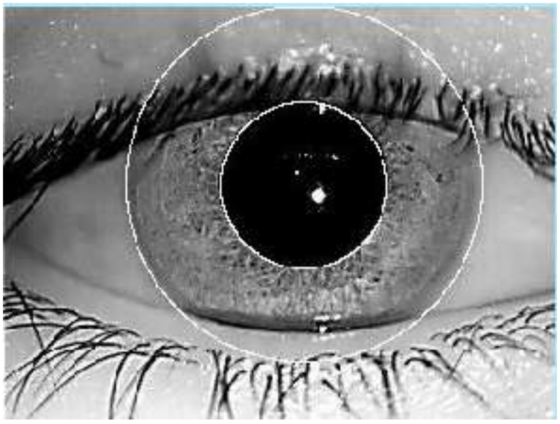

Figure 2. A segmented iris image

Circular Hough Transform

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 used to find the radius and centre coordinates of the pupil and iris regions. An automatic segmentation algorithm based on the circular Hough transform is employed by Wildes et al. [7], Kong and Zhang [14], Tisse et al. [15], and Ma et al. [16]. Firstly, an edge map is generated by calculating the first derivatives of intensity values in an eye image and then thresholding the result. These parameters are the centre coordinates x_c and y_c ,and the radius r, which are able to define any circle according to the equation

$$X_c^2+y_c^2-c^2=0$$

A maximum point in the Hough space will correspond to the radius and centre coordinates of the circle best defined by the edge points. Wildes et al. and Kong and Zhang also make use of the parabolic Hough transform to detect the eyelids, approximating the upper and lower eyelids with parabolic arcs, which are represented as: shown on next page-

$(-(x-h_j)\sin\theta_j + (y-k_j)\cos\theta_j)^2 = a_j((x-h_j)\cos\theta_j + (y-k_j)\sin\theta_j$

Where a_j controls the curvature, (h_j, k_j) is the peak of the parabola and θ_j is the angle of rotation relative to the x-axis.

4. Normalisation

Once the iris region is successfully located in an eye image, the next stage is to transform the iris region so that it has fixed dimensions in order to allow comparisons. The dimensional inconsistencies between eye images are mainly due to the stretching of the iris caused by pupil dilation from varying levels of illumination. Other sources of inconsistency include, varying imaging distance, rotation of the camera, head tilt, and rotation of the eye within the eye socket. The normalisation process will produce iris regions, which have the same constant dimensions, so that two photographs of the same iris under different conditions will have characteristic features at the same spatial location [5].

Some important normalization techniques are:

1. Daugman’s rubber sheet model
2. Image registeration
3. Virtual circles

Here we have implemented the Daugman’s rubber sheet model for normalisation among the all techniques stated above.

Daugman’s Rubber Sheet Model:

The homogenous rubber sheet model devised by Daugman [3] remaps each point within the iris region to a pair of polar coordinates (r,θ) where r is on the interval [0,1] and θ is angle $[0,2\pi]$.

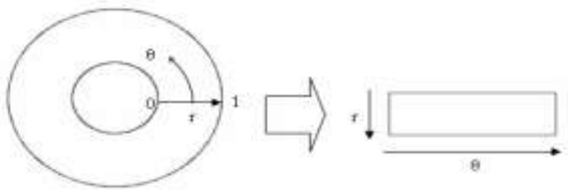

Figure 3. Daugman’s Rubber sheet model.

The remapping of the iris region from *(x, y)* Cartesian coordinates to the normalized non-concentric polar representation is modeled as [3].

$I(x(r,\theta),y(r,\theta)) \rightarrow I(r,\theta)$

with

$x(r,\theta) = (1-r)\, x_p(\theta) + r\, x_i(\theta)$

$y(r,\theta) = (1-r)\, y_p(\theta) + r\, y_i(\theta)$

Where $I(x, y)$ is the iris region image, (x, y) are the original Cartesian coordinates, (r, θ) are the corresponding normalized polar coordinates and x_p , y_p x_l , y_l coordinates of the pupil and iris boundaries along the direction. 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 modeled as a flexible rubber sheet anchored at the iris boundary with the pupil centre as the reference point.

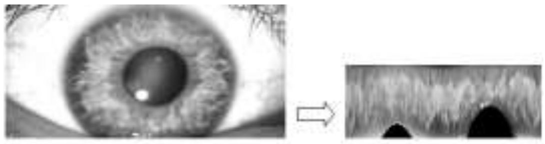

Figure 4. A normalised image

5. Feature Extraction

“One of the most interesting aspects of the world is that it can be considered to be made up of patterns. A pattern is essentially an arrangement. It is characterized by the order of the elements of which it is made, rather than by the intrinsic nature of these elements” (Nobert Wiener)[G.book] [13].

Hence to recognize the individuals, now we have to extract the information present inside that normalized iris template.

To make the system fast and efficient only the significant features of the iris must be encoded so that comparisons between templates can be made fast. Most iris recognition systems make use of a band pass decomposition of the iris image to create a biometric template [5].

The template that is generated in the feature encoding process will also need a corresponding matching metrics, which gives the two set of values one for intra class comparisons and another for intra class comparisons. These two cases should give distinct and separate values, so that a decision can be made with high confidence as to whether two templates are from the same iris, or from two different irises [6].

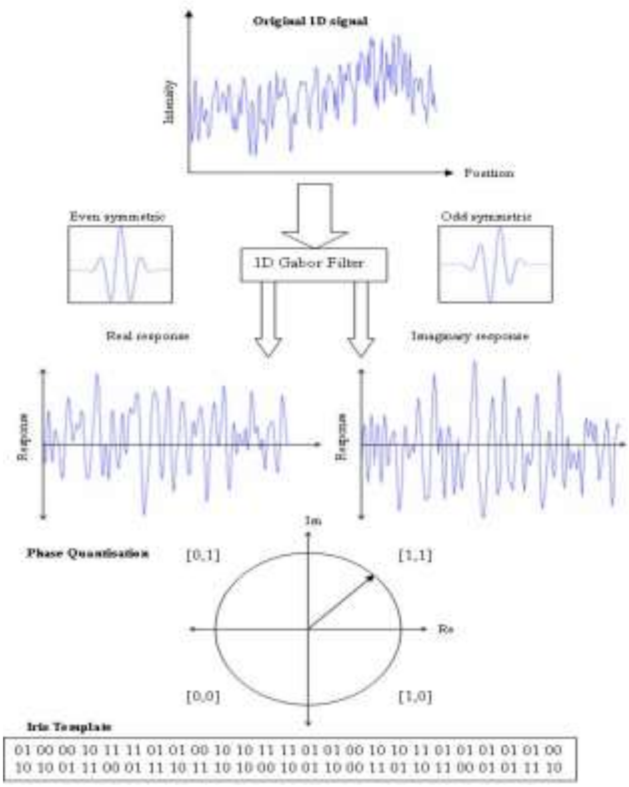

Figure 5 feature encoding explanation [5].

ISSN: 2321-8169
3085 – 3091

Some important Feature extraction methods are:

1. Wavelet encoding
2. Gabor filters
3. Log-gabor filters
4. Zero crossings of the 1D wavelet
5. Haar wavelet
6. Laplacian of Gaussian filters

We have used here the efficient 1-D Log gabor filter method for feature extraction method.

1-D Log Gabor Filter:

Feature encoding was implemented by convolving the normalised iris pattern with 1D Log-Gabor wavelets. The 2D normalised pattern is broken up into a number of 1D signals, and these 1D signals are convolved with 1D Gabor wavelets. The rows of 2D normalised pattern are taken as the 1D 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 normalised pattern, since maximum independence occurs in the angular direction [5].

6. Matching

Matching is also an important step by which we compare and get the result whether two iris belong to same person or different person. Some popular Matching techniques are:

1. Hamming distance
2. Weighted Euclidean distance
3. Normalized correlation

Hamming Distance

The Hamming distance gives a measure of how many bits are same between two bit patterns. Using the Hamming distance of two bit patterns, a decision can be made as to whether the two patterns were generated from different irises or from the same one.

In comparing the bit patterns *X* and *Y*, the Hamming distance, *HD*, is defined as the sum of disagreeing bits (sum of the exclusive-OR between *X* and *Y*) over *N*, the total number of bits in the bit pattern.

where Xj and Yj are the two bit-wise templates to compare, Xn_j and Yn_j are the corresponding noise masks for Xj and Yj, and N is the number of bits represented by each template [5].

$$HD = \frac{1}{N-\sum_{k=1}^{N} Xnk(OR)Ynk} \sum_{j=1}^{N} Xj(XOR)Yj(AND)Yn'j$$

If two bits patterns are completely independent, the Hamming distance between the two patterns should equal 0.5. This occurs because independence implies the two bit patterns will be totally random, so there is 0.5 chance of setting any bit to 1, and vice versa. Therefore, half of the bits will agree and half will disagree between the two patterns. If two patterns are derived from the same iris, the Hamming distance between them will be close to 0.0, since they are highly correlated. The Hamming distance is the matching metrics employed by Daugman[1],

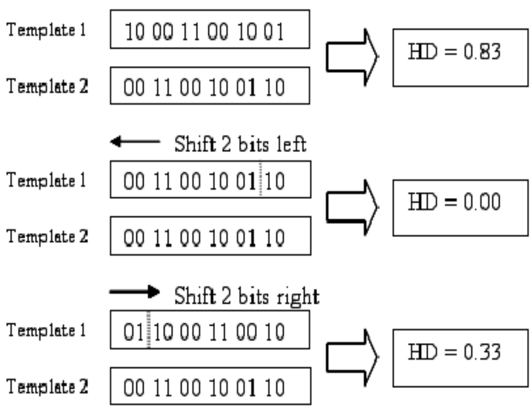

Figure 6: An illustration of the shifting process.

One shift is defined as one shift left, and one shift right of a reference template. In this example one filter is used to encode the templates, so only two bits are moved during a shift. The lowest Hamming distance, in this case zero, is then used since this corresponds to the best match between the two templates.

7. DECISION

The next stage is to take decision based on the resultant hamming distance corresponding to inter-class or intra-class comparisons that whether the input iris image is recognized or not. So here we have put this condition in our code that:

If H.D <= 0.30 decide that, this is **same person**

If H.D > 0.30 decide that, this is **different person**

This technique can be made more faster by weight map technique[17] and by exclusive OR operation[18].

IV. RESULTS

Iris recognition system: The MATLAB® GUI

We have developed the MATLAB GUI (Graphical user interface) system by using the GUIDE toolbox of MATLAB. The GUI of iris recognition system contains two image windows to show input and output iris image. The first window shows the input image that we select to recognize. And the window will show the output image after the recognition process completes.

Above the first image window, there are two edit boxes. The first edit box shows always a static character “SELECTED IMAGE” while the other edit box just above the input image window shows the input image’s name or number.

In a similar way there are two more boxes above the output image window. The first box is a list box that include various options to show all output images related to the particular recognition process like,

1. Recognized person name/number
2. Iris boundary
3. Normalized image
4. Iris noise
5. Iris image template
6. Template noise

So by using the list box we can see by selecting all these images related to current recognition process.

And there is one another box just above the output image window. That is the edit box that shows the recognized person's name or number.

For the good understanding of the functioning or working of this GUI system. We have shown step by step process of this system that how it works:

A. An input image is being selected to recognize

To select an input image just click on 'SELECT IMAGE TO RECOGNISE' button then a file explorer will open. After that select the input iris image to recognize from the Iris image data set.

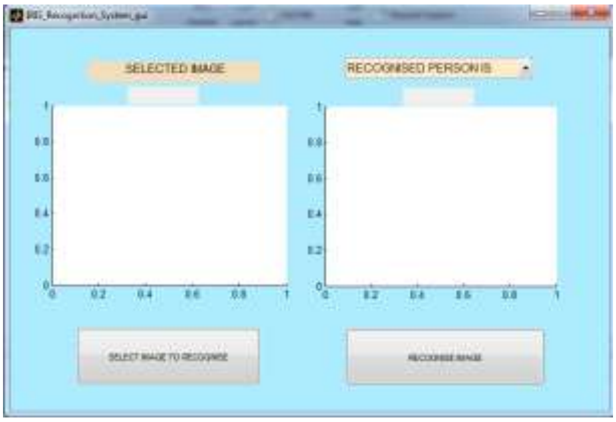

Figure 7 iris recognition main Gui window

In second image the options of list box are shown. It shows that you may select any option by clicking on list box to see other images related to that particular recognition after the recognition process is complete.

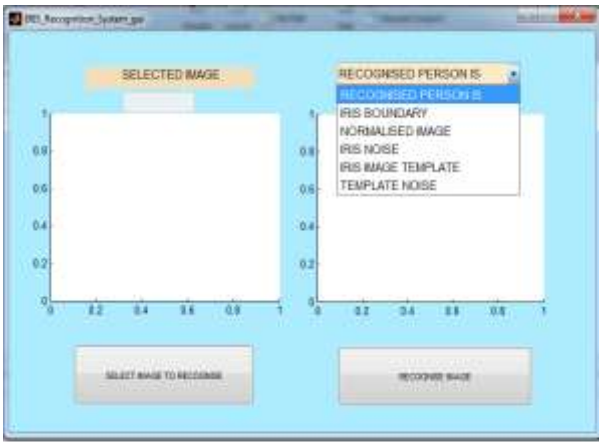

Figure 8 available options in GUI window

B. Input image will be shown in first image window of GUI

When you select an input image to recognise after that the image will be shown In the first image window the selected input image will be shown in the first image window and the name or number of particular image will be shown in edit box above the input image window.

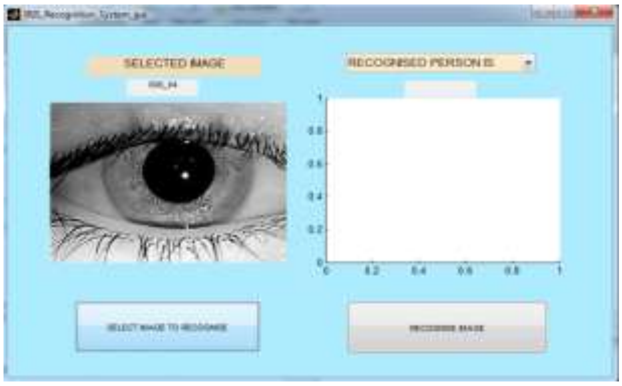

Figure 9 GUI showing the input image

C. Select location of saved Iris templates to compare

When you click on recognize button then a file explorer will open, then select the location of already saved templates/mask to match with the input iris template.

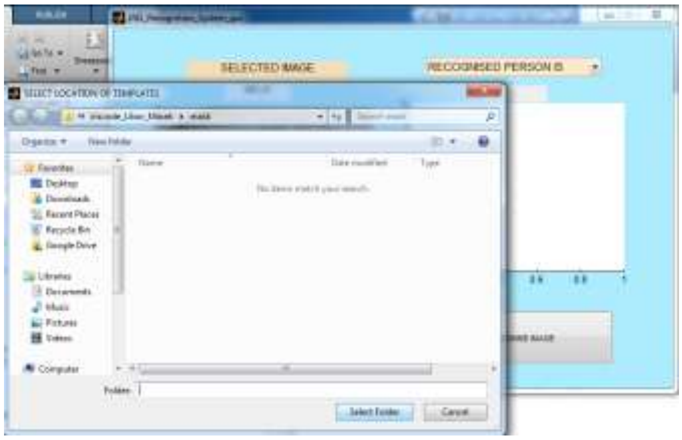

Figure 10 file explorer to select template location

D. Hamming distance will be shown in output window of MATLAB

After the recognition process is complete output window shows the value of HD (hamming distance) corresponding to the output image.

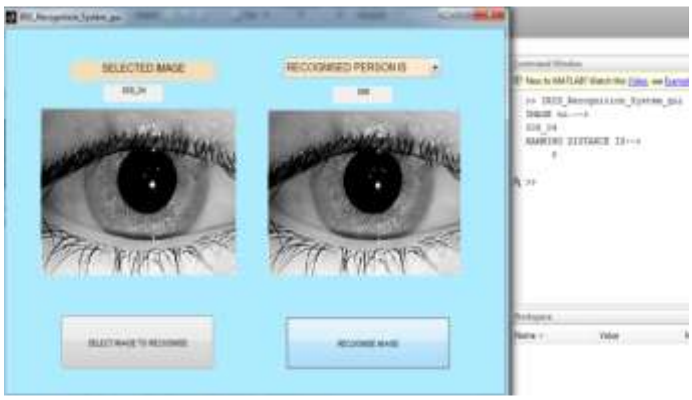

Figure 11 main gui with output window showing hamming distance

ISSN: 2321-8169
3085 – 3091

E. Click on list box to see other output images

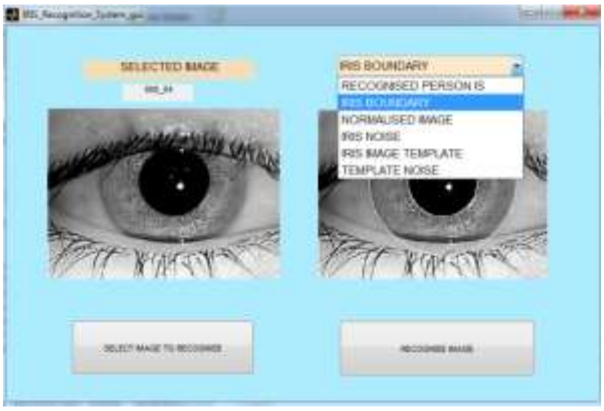

Figure 12 list box showing all options for output images

We can choose any available option to select the output image to see from the list box. Here we have chosen the segmented input Iris image by the name iris boundary.

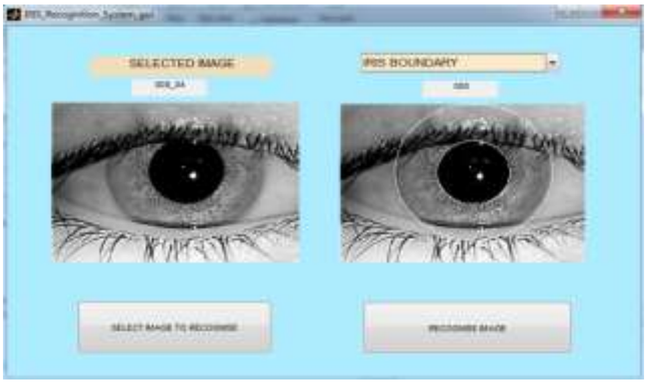

Figure 13 GUI showing iris boundary image

F. Select normalized image to see the interference caused by upper and lower eyelid

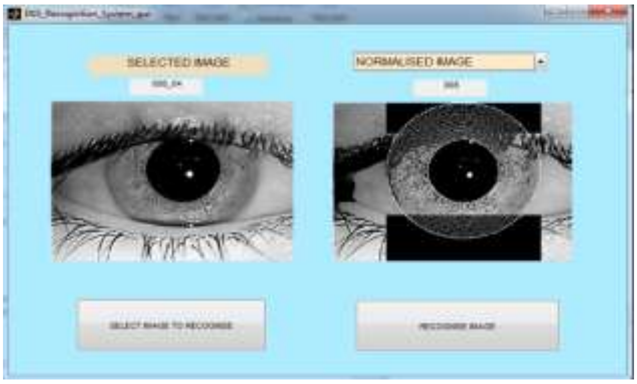

Figure 14 GUI showing normalised image

G. Select iris noise to see noise present in the Iris of eye image

Iris noise may be due to bad quality of imaging or due to the interference caused by the upper and lower eyelids. Here the GUI window is showing the noise present in the Iris region of input eye image.

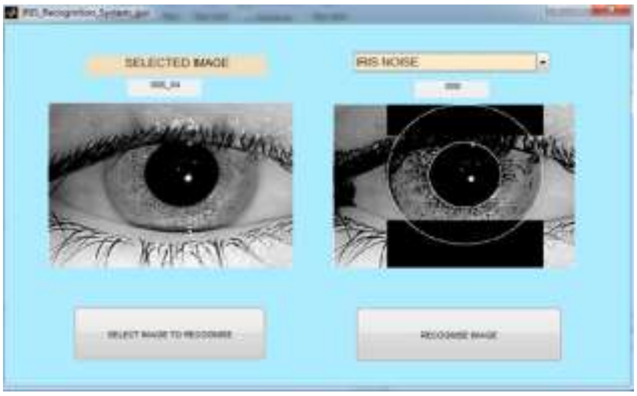

Figure 15 gui showing iris noise image

H. Select iris template to see iris image template

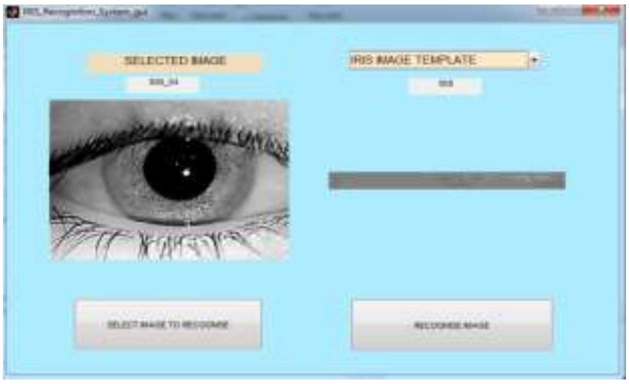

Figure 16. Gui showing iris image template

I. Iris template showing noise present in input iris template

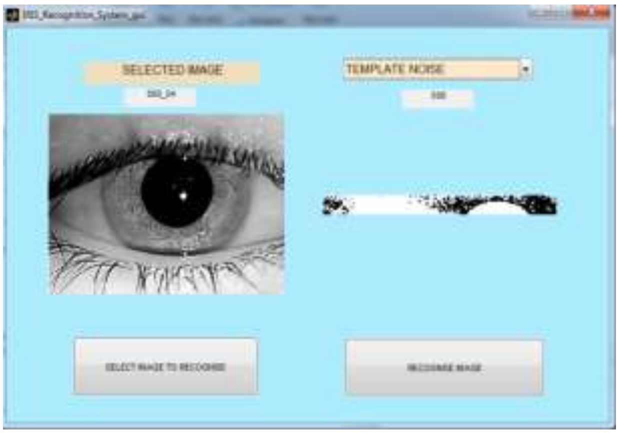

Figure 17. GUI showing noise present in iris template

V. CONCLUSION

We have successfully developed an Efficient and Optimal Iris recognition system using MATLAB GUI and tested it on 224 eye images of IIT Delhi Iris image dataset successfully. There were some areas they need to be improved like, segmentation and matching stage as they are the one of the most important stage of the Iris recognition system. And in this research work we have not considered the image acquisition stage, and Iris image acquisition from video or live recordings to make an efficient real time GUI based Iris recognition system. So this may be the further scope of development.

ACKNOWLEDGEMENT

Some Portions of the research in this paper use the algorithms developed by Libor Masek [5] and Kovesi [6]. The authors would like to thank them for their valuable help. And Authors would like to express their gratitude to Indian Institute of technology Delhi (INDIA) also, for providing us the iris eye image dataset.

REFERENCES

[1] S. Sanderson, J. Erbetta, "Authentication for secure environments based on iris scanning technology". *IEEE Colloquium on Visual Biometrics*, 2000a.

[2] Sudipta Roy, Abhijit Biswas, "A Personal Biometric Identification Technique based on Iris Recognition," (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 2 (4) , 2011, 1474-1477 .

[3] J. Daugman, "How iris recognition works," IEEE Trans. on Circuits and Systems for Video Technology., vol. 14, no. 1, 2004.

[4] J. Daugman, "How Iris Recognition Works", Proceedings of 2002 International conference on Image processing val. 1, 2002.

[5] MATLAB 14, source code for the iris recognition software presented in this publication is available via the World Wide Web at the address. http://www.csse.uwa.edu.au/~masekl01/.

[6] P. Kovesi. "MATLAB Functions for Computer Vision and Image Analysis". Available at: http://www.cs.uwa.edu.au/~pk/Research/MatlabFns/index.html

[7] R. Wildes, J. Asmuth, G. Green, S. Hsu, R. Kolczynski, J. Matey, S. McBride. "A system for automated iris recognition". *proceedings IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, pp. 121-128, 1994.

[8] R. Wildes. "Iris recognition: an emerging biometric technology". *Proceedings of the IEEE*, Vol. 85, No. 9, 1997.

[9] Teh Wei Hsiung and Shahrizat Shaik Mohamed, "Performance of Iris Recognition using Low Resolution Iris Image for Attendance Monitoring", 2011 International Conference on Computer Applications and Industrial Electronics (lCCAIE 2011). 978-1-4577 -2059-8/11/$26.00 ©2011 IEEE.

[10] L. Ma, Y. Wang, T. Tan. "Iris recognition using circular symmetric filters". National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 2002.

[11] S. Sanderson, J. Erbetta. "Authentication for secure environments based on iris scanning technology". IEEE Colloquium on Visual Biometrics, 2000.

[12] W. Boles, B. Boashash. "A human identification technique using images of the iris and wavelet transform". IEEE Transactions on Signal Processing, Vol. 46, No. 4, 1998.

[13] Gonzalez, R.C., Woods, R.E, "*Digital Image Processing*", 2nd ed., Prentice Hall (2002).

[14] W. Kong, D. Zhang. Accurate iris segmentation based on novel reflection and eyelash detection model. *Proceedings of 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, Hong Kong, 2001.

[15] C. Tisse, L. Martin, L. Torres, M. Robert. Person identification technique using human iris recognition. *International Conference on Vision Interface*, Canada, 2002.

[16] L. Ma, Y. Wang, T. Tan. Iris recognition using circular symmetric filters. National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 2002.

[17] Wenbo Dong, "Matching Based on Personalized Weight Map" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 33, NO. 9, SEPTEMBER 2011.

[18] Zhaofeng He, "Toward Accurate and Fast Iris Segmentation for Iris Biometrics" IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 31, NO. 9, SEPTEMBER 2009