ACKNOWLEDGEMENT

A project is always a coordinated, guided and scheduled team effort aimed at realizing common minimum goals. It is executed by team members, though it never reaches completion by team efforts alone, nor does it reach the far off shores, unless there is lighthouse to guide the ship sailing astray.

We would like to express our deep gratitude and special thanks to all those people who have helped and guided us throughout our project. We would like to thank our project guide **Prof. Jayashri Mittal** for her constant supervision, encouragement and her valuable suggestions. We sincerely thank our project co-ordinator **Prof. G. Anuradha** and **Prof. Safa Hamdare** for their rigorous efforts and constant support in guiding us throughout. We thank **Prof. Yogesh Karunakar** for his help and providing necessary information in implementation of the project. Our efforts would have been incomplete without thanking our principal **Prof. (Dr) A. K. Sen** and our mentor **Prof. Bidisha Roy**, Head of Department; Computer whose guidance made this project complete.

We are also grateful to our college, all our teaching and non-teaching staff, those who have directly or indirectly helped us to make this report a successful one. We would also like thank our friends and family members for their help and support.

-Flevia Fargose (BE/CMPN B/13)

-Glancy Rodrigues (BE/CMPN B/39)

-Olivia Rodrigues (BE/CMPN B/40)

ABSTRACT

Iris recognition is one of the most promising approaches and rapidly expanding method of biometric authentication. The main goal behind this project on Iris Recognition is to build a system which uniquely identifies a human being. Iris is a protected internal organ whose random texture is stable throughout life, it can serve as a kind of living password that one need not remember but one always carries along. Since the degree of freedom of iris textures is extremely high, the probability of finding two identical irises is close to zero, and it cannot be tampered easily, the iris recognition systems are very reliable. The non-contact biometrics such as face and iris has additional benefits over contact-based biometrics such as fingerprint and hand geometry. This innovative technique will be computationally effective as well as reliable in terms of recognition rates. We aim at designing a system which will provide high security to the document based on person's iris which will act as his unique password.

LIST OF FIGURES

LIST OF TABLES

CONTENT

CHAPTER 1
INTRODUCTION

Iris Recognition is one of the biometric identification and authentication technique that can be used to reliably identify a person by analyzing the patterns found in the iris. Iris recognition is regarded as the most reliable and accurate biometric identification system available.

1.1 Description

This system describes in detail an application that provides secure measures for a document based on human iris. It allows the user to browse the directory and select iris image of particular user. This selected image will then be processed as a query input for matching with the already present images. On comparison if match found then it will grant access to the document.

1.2 Motivation

In an extremely computerized world such as today it is necessary to make our life more secure and cost-effective. There arises a need where the data which is private and highly confidential needs to be protected. We always create new accounts either on social networking site, mail accounts, bank accounts, etc. there are many passwords which we need to remember. But if at all we forget this passwords or someone else gets information regarding this, then it might put you in trouble you might even need a third party to handle care of your important documents and accounts. For this we came up with a new idea of Iris recognition in security. This helps us in accessing the particular document by using the human authentication.

1.3 Problem Formulation and Methodology Used

Today's security are in critical need of finding accurate, secure and cost-effective alternatives to passwords and personal identification numbers (PIN) as financial losses increase dramatically year over year from computer-based fraud such as computer hacking and identity theft Biometric solutions address these fundamental problems, because an individual's biometric data is unique and cannot be transferred. Providing iris as the password in order to secure the document is highly efficient as no other person have the same. And thus it will lead to unique identification of a particular person.

The basic idea that our project attempts to use is using iris as the password in order to open the document. Initially the database of iris images of ten users is collected and stored. On this images the processing is been done in order to extract the features of the iris. When any Query image is received the comparison is been done and if image is found then it grants access to the document. This technique is inherently more reliable and capable than traditional knowledge based and token based techniques.

1.4 Relevance of the project

Iris recognition is regarded as the most reliable and accurate biometric identification system available. The iris is so reliable as a form of identification because of the uniqueness of its pattern. Using this iris as the password many data system can be made more secure. We can use this iris as the password in order to secure any confidential document. Also it can be used in various applications such as Colleges, hospitals, offices, etc. where security is to be provided.

1.5 Scope of the project

The objective of this project is to produce a working application program that functions as an iris recognition tool, in order to implement this in an accurate and useful way which is also user-friendly. In particular, our system loads the image, finds the match image after finding match it opens the document. This provides user authentication and security.

1.6 Objectives of the project

The basic idea of our project is to create and application in MATLAB such that the user will have to authenticate the document using iris as password. If the exact match is found then only the access is granted to the user. Else an error message is been displayed which will display that match is not found or that particular user does not exist and user will not be given access to the document.

CHAPTER 2

LITERATURE REVIEW

The literature survey was done in which many IEEE papers were being read and the following inference was being made

Iris Identification Process: Human iris identification process is basically divided into five steps:

1. Image pre-processing:

a. Gabor filter:

In image processing, a Gabor filter, is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination. In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.[4]

b. Histogram Equalization:

Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. 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. This allows for areas of lower local contrast to gain a higher contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. [1]

2. Localization (Segmentation):

The inner and the outer boundaries of the iris are calculated. For Segmentation, there are 2 algorithms which are generally used.

a. The Hough transform:

It 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.

There are a number of problems with the Hough transform method. First of all, it requires threshold values to be chosen for edge detection, and this may result in critical edge points being removed, resulting in failure to detect circles/arcs. Secondly, the Hough transform is computationally intensive due to its ‘brute-force’ approach, and thus may not be suitable for real time applications.

b. Integro-differential operator:

Daugman makes use of an integro-differential operator for locating the circular iris and pupil regions, and also the arcs of the upper and lower eyelids. It does not suffer from the thresholding problems of the Hough transform. However, the algorithm can fail where there is noise in the eye image, such as from reflections, since it works only on a local scale. [2]

c. Edge detection algorithm:

Edges characterize boundaries and are therefore a problem of fundamental importance in image processing. Edges in images are areas with strong intensity contrasts a jump in intensity from one pixel to the next. Edge detecting an image significantly reduces the amount of data and filters out useless information, while preserving the important structural properties in an image. [1]

i. Roberts Operator:

The Roberts cross operator is used in image processing and computer vision for edge detection. It was one of the first edge detector algorithms. As a differential operator, the idea behind the Roberts cross operator is to approximate the gradient of an image through discrete differentiation which is achieved by computing the sum of the squares of the differences between diagonally adjacent pixels. [1]

ii. Prewitt Operator:

The Prewitt operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Prewitt operator is either the corresponding gradient vector or the norm of this vector. The Prewitt operator is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation which it produces is relatively crude, in particular for high frequency variations in the image. [1]

iii. Sobel operators:

The Sobel operator is used in image processing, particularly within edge detection algorithms. Technically, it is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. At each point in the image, the result of the Sobel operator is either the corresponding gradient vector or the norm of this vector. It is based on convolving the image with a small, separable, and integer valued filter in horizontal and vertical direction and is therefore relatively inexpensive in terms of computations. On the other hand, the gradient approximation that it produces is relatively crude, in particular for high frequency variations in the image. [1]

The result of the Sobel operator is a 2-dimensional map of the gradient at each point. It can be processed and viewed as though it is itself an image, with the areas of high gradient (the likely edges) visible as white lines.

iv. Canny operator:

The Canny edge detection algorithm is known to many as the optimal edge detector. A list of criteria to improve current methods of edge detection is the first and most obvious is low error rate. It is important that edges occurring in images should not be missed and that there be no responses to non-edges. The second criterion is that the edge points be well localized. In other words, the distance between the edge pixels as found by the detector and the actual edge is to be at a minimum. A third criterion is to have only one response to a single edge. This was implemented because the first two were not substantial enough to completely eliminate the possibility of multiple responses to an edge. [1]

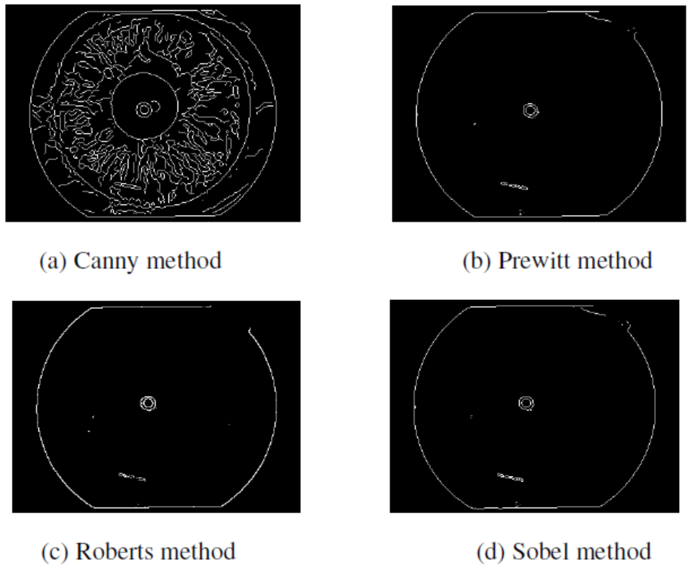


Figure 2.1 Diagram for Edge detection algorithms

d. Clipping of an image:

Clipping refers to the method of separating objects from their background in an image. This offers two major advantages. First, there is the removal of interfering surroundings or shadows to emphasize the product itself. Second, there is the possibility to insert the object harmoniously into a montage with different backgrounds.

3. Normalization:

Iris of different people may be captured in different size; even for the same person size may vary because of the variation in lighting and other factors. Hence for performing Normalization, there are 2 Models which are generally used.

a. Rubber Sheet Model:

It is used which remaps each point within the iris region to a pair of polar coordinates. In Mathematics, the polar coordinate system is a two-dimensional coordinate system in which each point on a plane is determined by a distance from a fixed point and an angle from a fixed direction. [3]

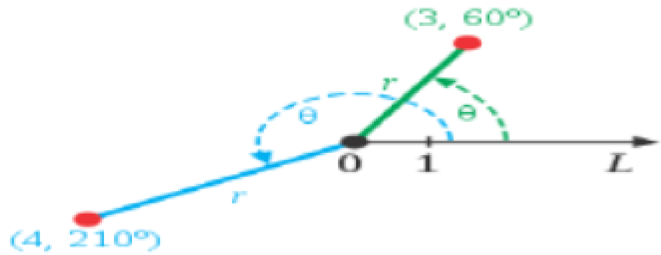


Figure 2.2 Polar coordinates

b. Virtual Circles Model :

Iris images are first scaled to constant diameter so that when comparing two images, one is considered as the reference image. This works differently to the other techniques, since normalization is not performed until attempting to match two iris regions, rather than performing normalization and saving the result for later comparisons. Once the two irises have the same dimensions, features are extracted from the iris region by storing the intensity values along virtual concentric circles, with origin at the centre of the pupil. This is essentially the same as Daugman's rubber sheet model; however scaling is at match time, and is relative to the comparing iris region, rather than scaling to some constant dimensions. [3]

4. Feature extraction:

Iris provides abundant texture information. A feature vector is formed which consists of the ordered sequence of features extracted from the various representation of the iris images.

2D-DWT:

A discrete wavelet transform (DWT) is any wavelet transform for which the wavelets are discretely sampled. The 2D discrete wavelet transform of a signal x is implemented by iterating the 2D analysis filter bank on the low-pass sub-band image. [2]

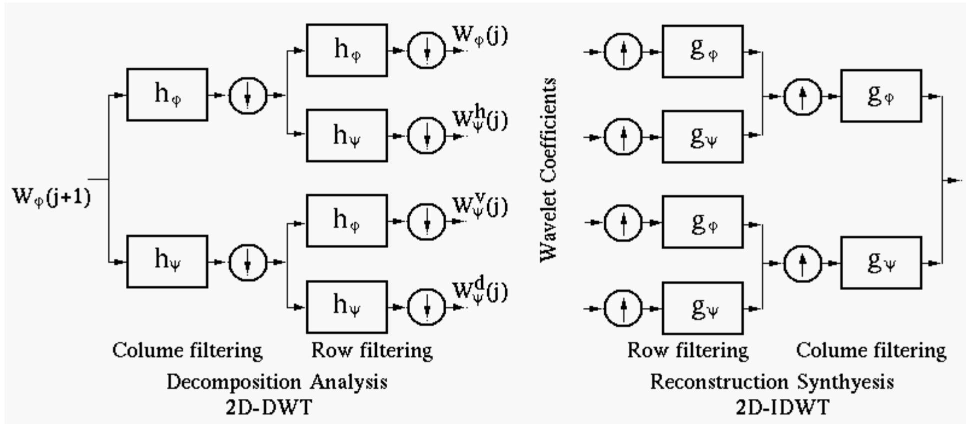


Figure 2.3 2D DWT Decomposition

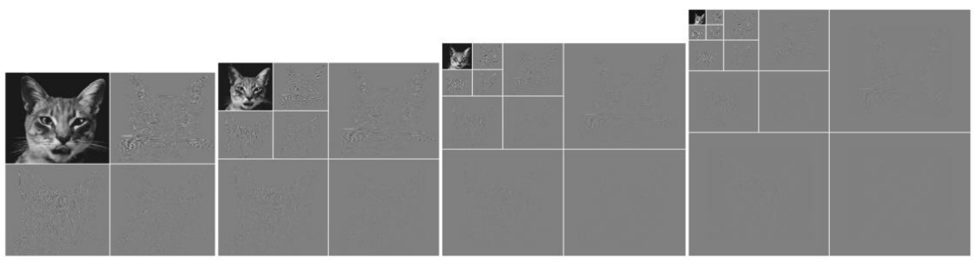


Figure 2.4 2D DWT Image Representation

5. Matching:

The feature vectors are classified through different thresholding(entry level) techniques like Hamming Distance, weight vector and winner selection, dissimilarity function, etc. [2]

CHAPTER 3

SYSTEM STUDY AND ANALYSIS

3.1 Existing System

Iris recognition is the best breed authentication process available today. This system is already used many fields such as ATM authentication, for UID, for personal identification on airport, some hardware applications like door unlocking, etc.

Some existing technologies include ‘Iridian’ and ‘Iris ID’ (LG IRIS). These technologies are been used at airport for human authentication and for the purpose of passport and visa verification.

Historically, identity or authentication conventions were based on things one possessed (a key, a passport, or identity credential) or something one knows (a password, the answer to a question, or a PIN). The possession or knowledge was generally all that was required to confirm identity or confer privileges. But the problem associated with this method is it is very difficult to memorize this password all the time: also a person has multiple accounts so remembering this password becomes a tedious issue.

As iris is the most unique feature of any human, it can be used in such cases to provide security. It is generally considered that iris recognition is the most accurate. Coupling this high confidence authentication with factors like outlier group size, speed, usage/human factors, platform versatility and flexibility for use in identification or verification modes, iris recognition has also shown it to be exceedingly versatile and suited for large population applications.

There are many algorithms and procedures available for iris recognition system. These include many image processing, localization, normalization, feature extraction and matching processes. The comparison of various algorithms is as follows:

Table 3.1: Comparison of Algorithms

Algorithm	Localization	Normalization	Feature Extraction	Matching
AVILA	Integro - Differential Operators	Virtual Circles Model	Dyadic Wavelet Transform	Hamming Distance & Euclidean Distance
Ma. Li	Nearest Feature Line (NFL) Method	Daugman's Rubber Sheet Model then Iris Image enhancement and denoising	Circular Symmetric Filters	Hamming Distance
Tisse	Integro - Differential Operators with Hough Transform	Daugman's Rubber Sheet Model	Gabor's Complex 2D Passing Filters	Hamming Distance
Daugman	Integro - Differential Operators	Daugman's Rubber Sheet Model	Gabor's Complex 2D Passing Filters	Hamming Distance

Its performance criteria are as follows:

Table 3.2 Result of Algorithms

Algorithm	FAR	FRR	Overall % Accuracy
AVILA	0.03	2.08	97.89
Ma. Li	0.02	1.98	98.00
Tisse	1.84	8.79	89.37
Daugman	0.01	0.09	99.90

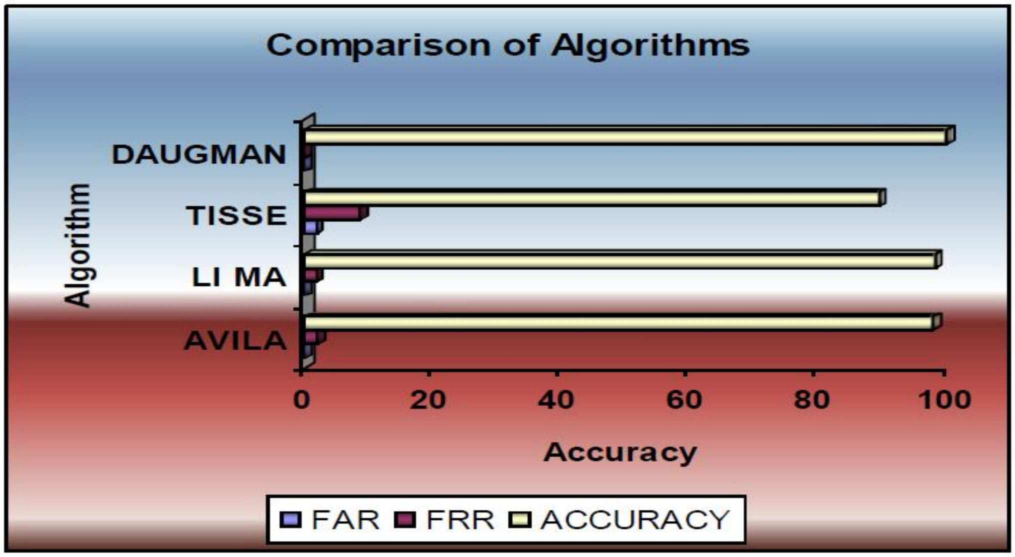


Figure 3.1 Comparison Chart for Algorithms using FAR, FRR and Accuracy

3.2 Proposed System

In order to overcome the drawbacks of existing systems we introduce a system in which all this passwords and important documents will be stored in a file and the iris image so captured will be used as a password.

1. **Image Storage:**

   The system consists of 10 users, each having their 2 iris images of same eye being stored in files making the database size of 20 images.

2. **Image processing :**

   When the users want to access his document, he provides his iris image as a query. This query image and images in storage files undergo feature extraction procedure. The feature vector is extracted as by the following steps:

   a. **Image Preprocessing:**

      In this the image obtained is first converted to Gray scale. On this gray level image, Histogram equalization is performed and equivalent image is obtained. It increases the global contrast of the iris image.

b. Image Localization:

In order to do localization of the exact iris present so as to extract the features, Clipping is done, where we remove the irrelevant part of image present on the edges. After this, to remove the specular reflection present in the eye, masking of the image is done. We mask the input iris image with the bitmap Mask image and get the masked image.

c. Iris code generation:

To obtain the exact features of the image, we do energy level decomposition of the image. For this we apply 2D DWT. And thus decomposition of the image is obtained up to 4^{th} level in order to gain the exact features. The unique features the decomposed image is obtained by applying Canny Edge Detection and the values are been stored into the excel sheet in 0'sand 1'sformat.The output obtained is the iris code which can be used for comparison.

d. Iris code storage:

This generated iris code is converted to vector form and stored in style sheet for further use.

3. Iris code Matching :

The matching module generates a match score by comparing the feature sets of the query image with all images stored. The technique used for comparing two iris codes is hamming distance, in which the number of corresponding bits that differ between the two iris codes is calculated. After this the minimum value is found out by sorting them into the ascending order, and a threshold of minimum distance found through analysis, here the value 0.09 is set in order to identify the exact person. If match found then the output saying match found is displayed else it gives error message.

4. Accessing the file:

If the query so obtained matches with the image present in the database then the user is given the access to his document else error message is been displayed which tells the user is not valid.

3.3 Requirement Analysis

Functional Requirements:

- There are total 10 users present, 2 image of each person are stored in storage files making the size as 20.Another storage file consists of one query images, there are total 10 query images stored.
- The user needs to properly store the iris image.
- In order to access the document, the document to which access is to be granted needs to be present.

Non-functional Requirements:

- The iris image should not be blur.
- The image obtain should not contain any noise.
- After the match is found quick access to the document must be given.

Hardware Requirements:

- Processor : Pentium 4
- RAM : 512MB
- Hard Disk : 160GB

Software Requirements:

- Operating System : Windows 7
- Matlab : R2008a

3.4 Requirement Specification

Capacity Requirements:

- Number of users – About 10 to 20 or more
- Size of Database – About 30

Response time Requirements:

- For image to load – About 5 to 10 seconds.
- For match to be found – About 15 to 20 seconds.
- Access to document – About 7 to 8 seconds.

3.5 Requirement Validation

- If the user is not valid, system should not grant access to any of the document.
- The user must first load the iris image in order to perform any further action such as find match, access the document or view the intermediate steps.
- Without matching the iris code the user cannot access the document.

3.6 Use-Case Diagrams and description

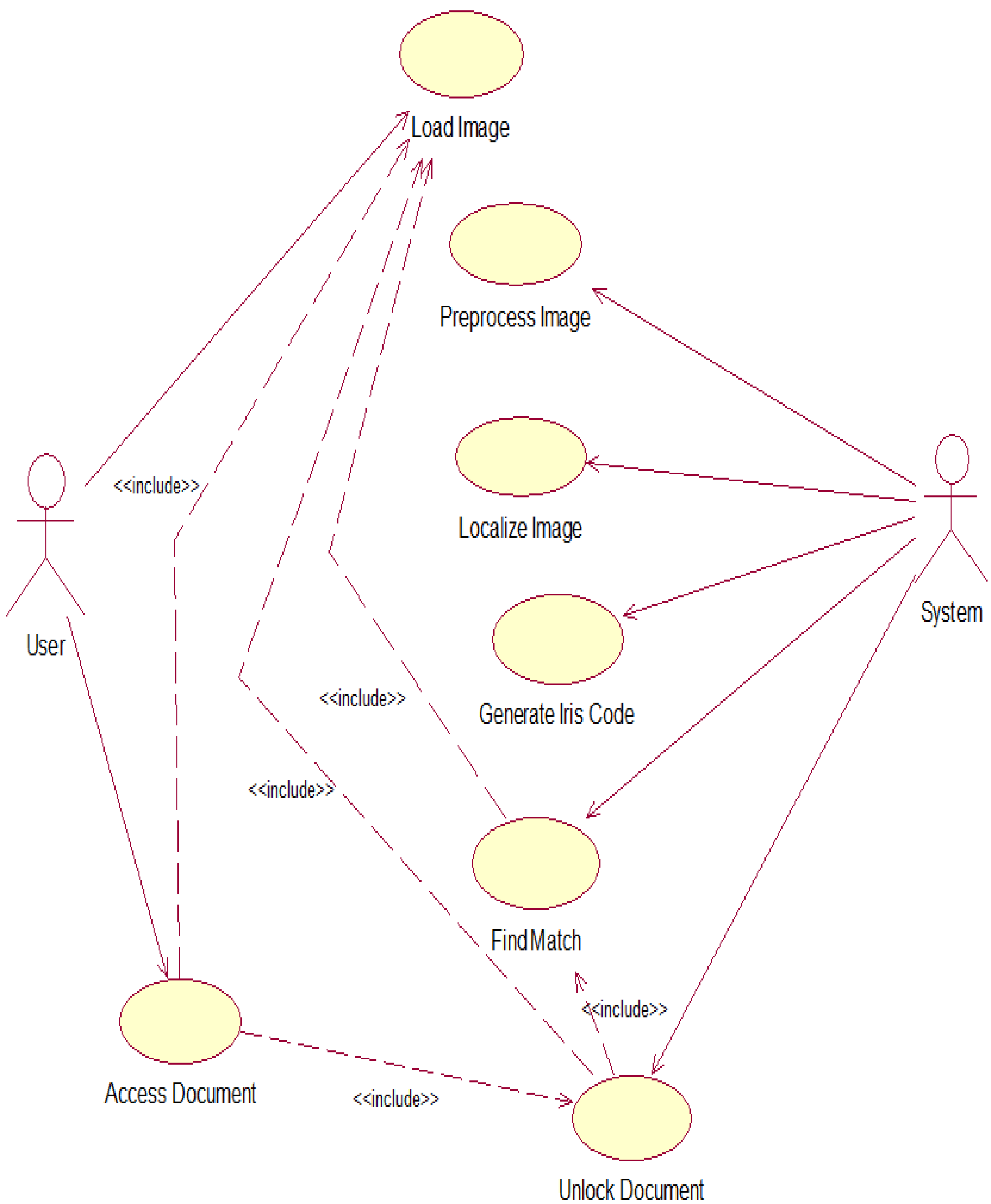


Figure 3.2 Use Case Diagram

Table 3.3 Use case Description

ACTOR	**USE CASE**	**DESCRIPTION**
User	Load Iris Image	Iris image of the user is loaded into the system.
User	Access Document	If iris is matched then user can access the document.
System	Image Preprocessing	The Iris image is pre-processed in order to remove redundancy and enhance it.
System	Image Localization	The actual iris portion is obtained and masked in order to remove specular reflection.
System	Iris code generation	The iris code is been generated out of localized image present.
System	Find matching	The exact match of the query image is found from the storage files whose iris code is stored in style sheets.
System	Unlock Document	After matching it unlocks the document of respective user.

CHAPTER 4

ANALYSIS MODELING

4.1 Activity Diagram

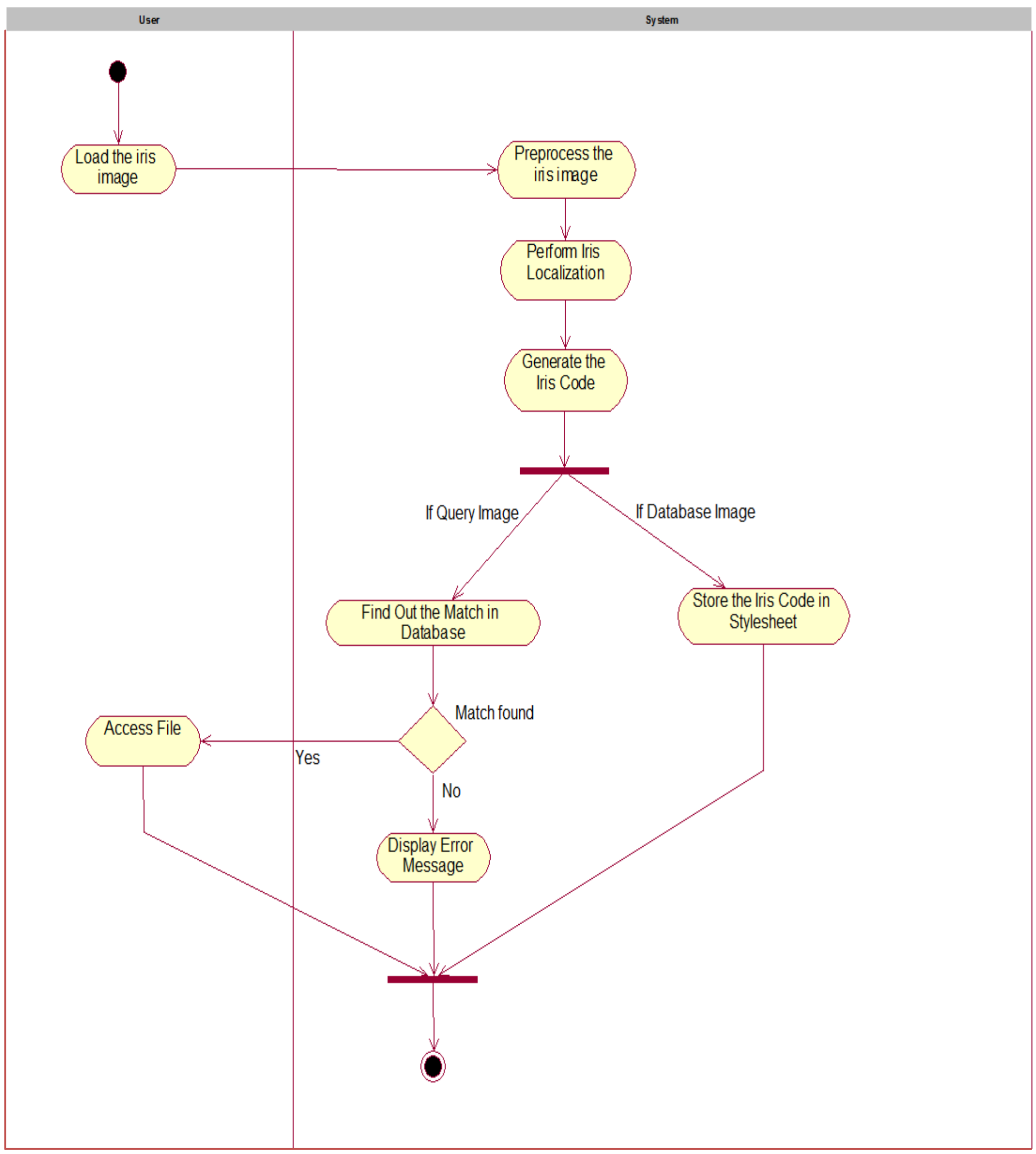


Figure 4.1 Activity Diagram

4.2 Functional Modelling (DFD)

DFD Level 0:

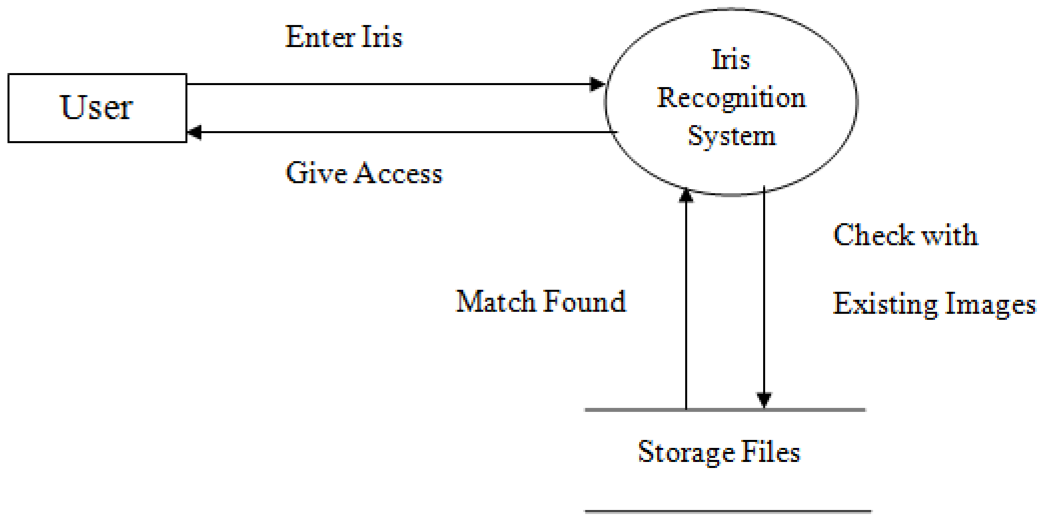


Figure 4.2 Data Flow Diagram level 0

DFD Level 1:

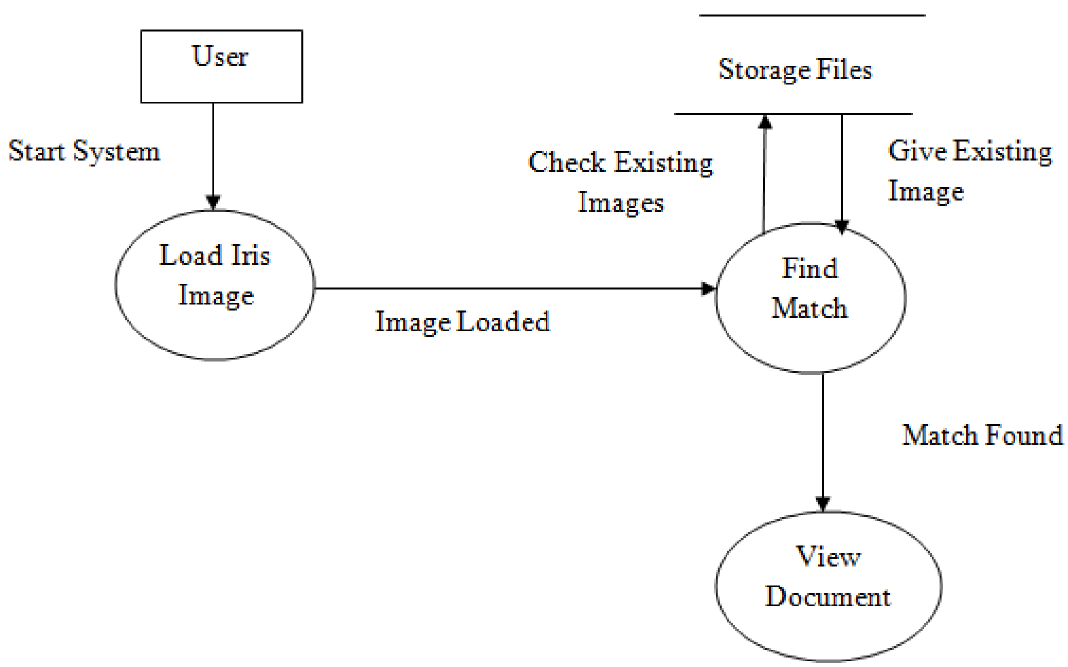


Figure 4.3 Data Flow Diagram level 1

DFD Level 2:

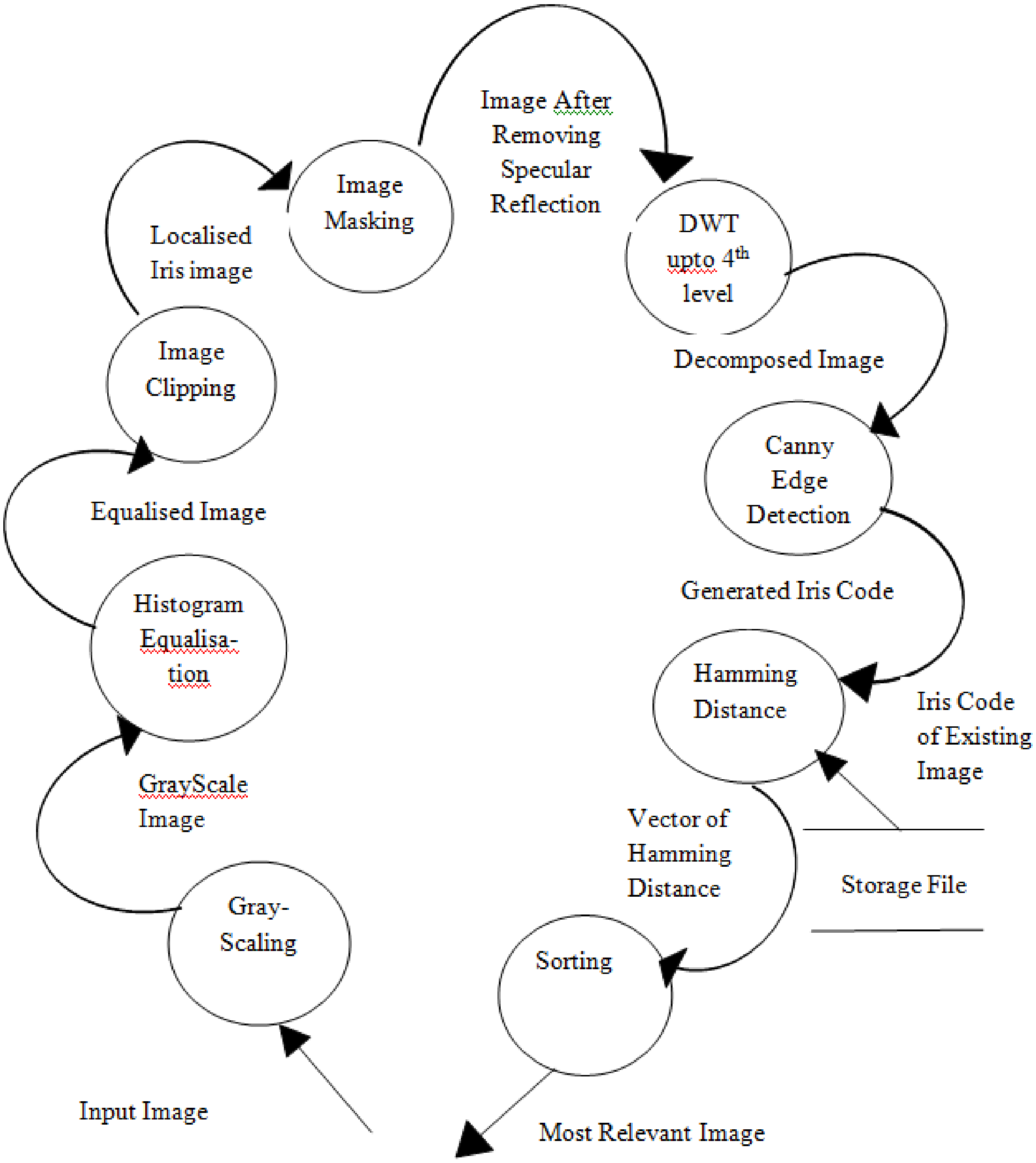


Figure 4.4 Data Flow Diagram level 2

4.3 TimeLine Chart

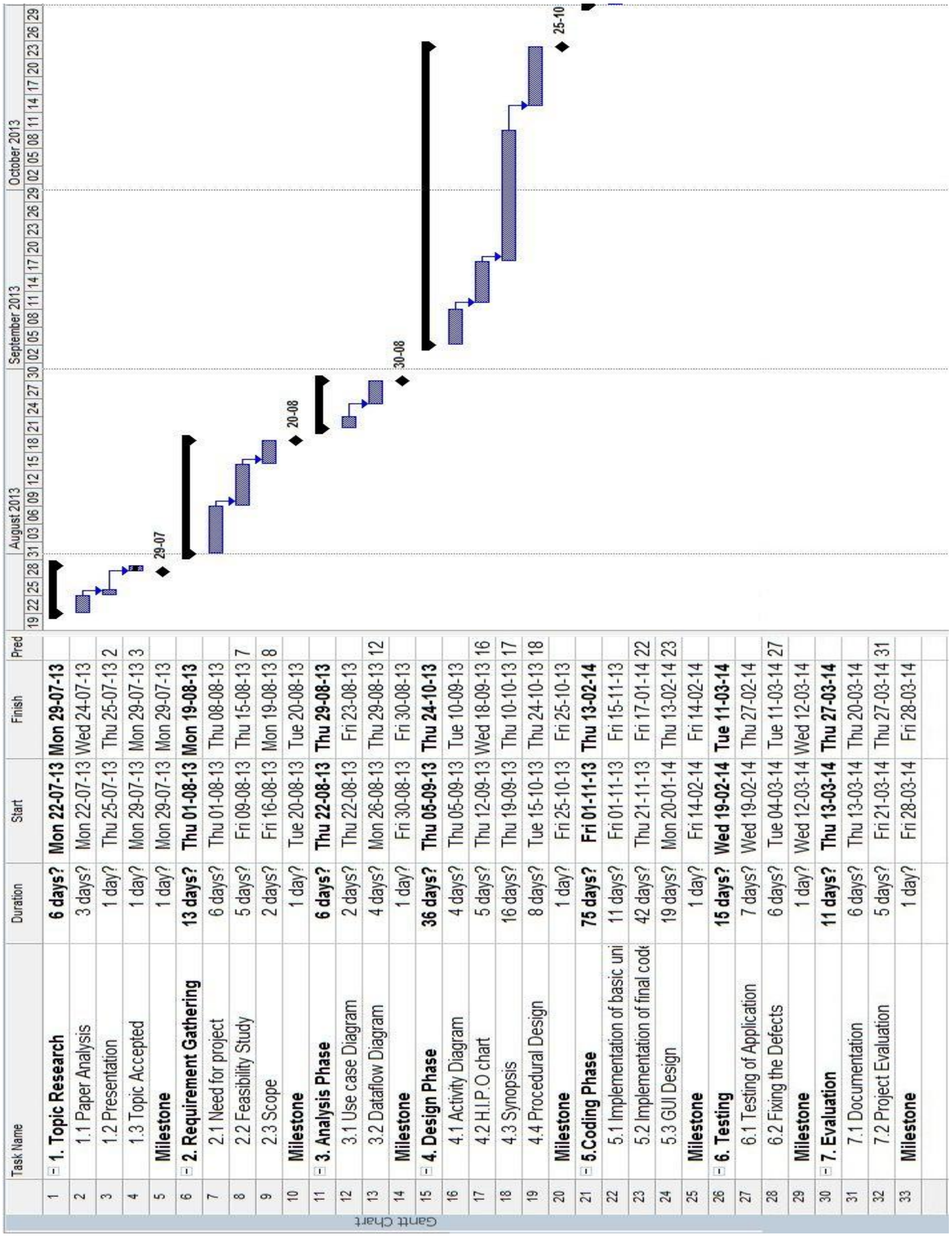
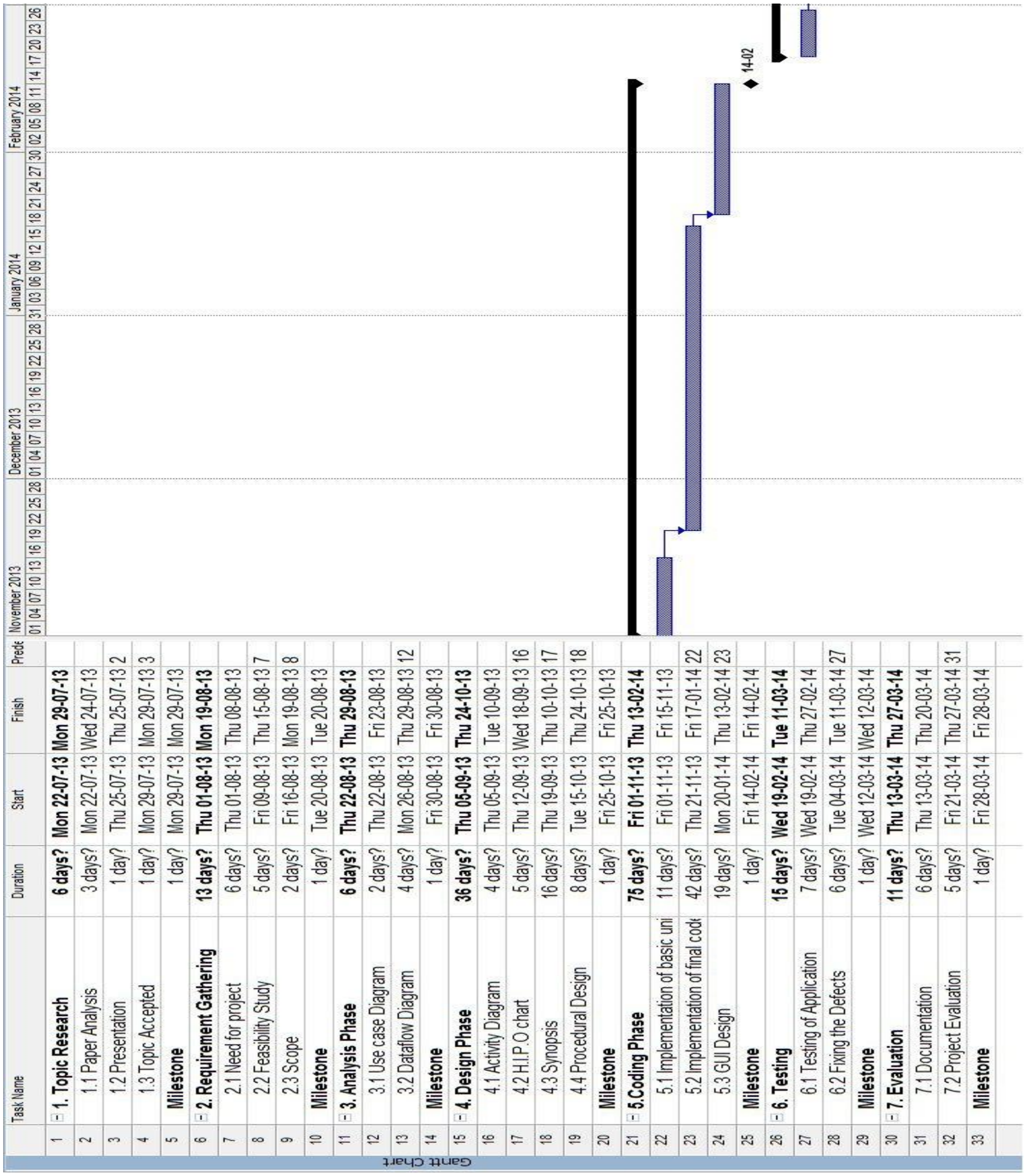
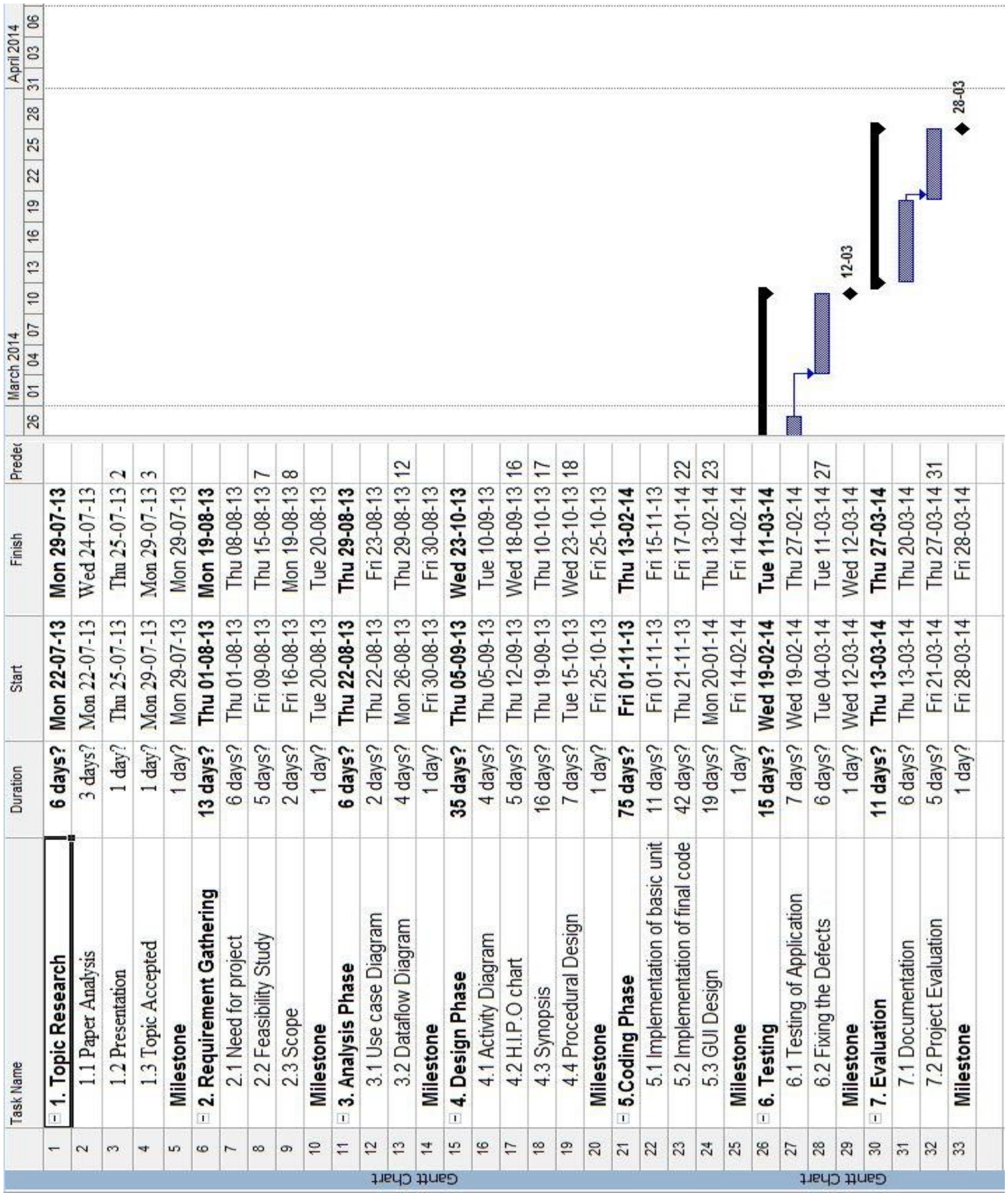


Figure 4.5 TimeLine Chart



	Task Name	Duration	Start	Finish	Prede
1	**1. Topic Research**	**6 days?**	**Mon 22-07-13**	**Mon 29-07-13**	
2	1.1 Paper Analysis	3 days?	Mon 22-07-13	Wed 24-07-13	
3	1.2 Presentation	1 day?	Thu 25-07-13	Thu 25-07-13	2
4	1.3 Topic Accepted	1 day?	Mon 29-07-13	Mon 29-07-13	3
5	**Milestone**	1 day?	Mon 29-07-13	Mon 29-07-13	
6	**2. Requirement Gathering**	**13 days?**	**Thu 01-08-13**	**Mon 19-08-13**	
7	2.1 Need for project	6 days?	Thu 01-08-13	Thu 08-08-13	
8	2.2 Feasibility Study	5 days?	Fri 09-08-13	Thu 15-08-13	7
9	2.3 Scope	2 days?	Fri 16-08-13	Mon 19-08-13	8
10	**Milestone**	1 day?	Tue 20-08-13	Tue 20-08-13	
11	**3. Analysis Phase**	**6 days?**	**Thu 22-08-13**	**Thu 29-08-13**	
12	3.1 Use case Diagram	2 days?	Thu 22-08-13	Fri 23-08-13	
13	3.2 Dataflow Diagram	4 days?	Mon 26-08-13	Thu 29-08-13	12
14	**Milestone**	1 day?	Fri 30-08-13	Fri 30-08-13	
15	**4. Design Phase**	**36 days?**	**Thu 05-09-13**	**Thu 24-10-13**	
16	4.1 Activity Diagram	4 days?	Thu 05-09-13	Tue 10-09-13	
17	4.2 H.I.P.O chart	5 days?	Thu 12-09-13	Wed 18-09-13	16
18	4.3 Synopsis	16 days?	Thu 19-09-13	Thu 10-10-13	17
19	4.4 Procedural Design	8 days?	Tue 15-10-13	Thu 24-10-13	18
20	**Milestone**	1 day?	Fri 25-10-13	Fri 25-10-13	
21	**5.Coding Phase**	**75 days?**	**Fri 01-11-13**	**Thu 13-02-14**	
22	5.1 Implementation of basic uni	11 days?	Fri 01-11-13	Fri 15-11-13	
23	5.2 Implementation of final code	42 days?	Thu 21-11-13	Fri 17-01-14	22
24	5.3 GUI Design	19 days?	Mon 20-01-14	Thu 13-02-14	23
25	**Milestone**	1 day?	Fri 14-02-14	Fri 14-02-14	
26	**6. Testing**	**15 days?**	**Wed 19-02-14**	**Tue 11-03-14**	
27	6.1 Testing of Application	7 days?	Wed 19-02-14	Thu 27-02-14	
28	6.2 Fixing the Defects	6 days?	Tue 04-03-14	Tue 11-03-14	27
29	**Milestone**	1 day?	Wed 12-03-14	Wed 12-03-14	
30	**7. Evaluation**	**11 days?**	**Thu 13-03-14**	**Thu 27-03-14**	
31	7.1 Documentation	6 days?	Thu 13-03-14	Thu 20-03-14	
32	7.2 Project Evaluation	5 days?	Fri 21-03-14	Thu 27-03-14	31
33	**Milestone**	1 day?	Fri 28-03-14	Fri 28-03-14	

Figure 4.6 TimeLine Chart



	Task Name	Duration	Start	Finish	Predec
1	**1. Topic Research**	**6 days?**	**Mon 22-07-13**	**Mon 29-07-13**	
2	1.1 Paper Analysis	3 days?	Mon 22-07-13	Wed 24-07-13	
3	1.2 Presentation	1 day?	Thu 25-07-13	Thu 25-07-13	2
4	1.3 Topic Accepted	1 day?	Mon 29-07-13	Mon 29-07-13	3
5	**Milestone**	1 day?	Mon 29-07-13	Mon 29-07-13	
6	**2. Requirement Gathering**	**13 days?**	**Thu 01-08-13**	**Mon 19-08-13**	
7	2.1 Need for project	6 days?	Thu 01-08-13	Thu 08-08-13	
8	2.2 Feasibility Study	5 days?	Fri 09-08-13	Thu 15-08-13	7
9	2.3 Scope	2 days?	Fri 16-08-13	Mon 19-08-13	8
10	**Milestone**	1 day?	Tue 20-08-13	Tue 20-08-13	
11	**3. Analysis Phase**	**6 days?**	**Thu 22-08-13**	**Thu 29-08-13**	
12	3.1 Use case Diagram	2 days?	Thu 22-08-13	Fri 23-08-13	
13	3.2 Dataflow Diagram	4 days?	Mon 26-08-13	Thu 29-08-13	12
14	**Milestone**	1 day?	Fri 30-08-13	Fri 30-08-13	
15	**4. Design Phase**	**35 days?**	**Thu 05-09-13**	**Wed 23-10-13**	
16	4.1 Activity Diagram	4 days?	Thu 05-09-13	Tue 10-09-13	
17	4.2 H.I.P.O chart	5 days?	Thu 12-09-13	Wed 18-09-13	16
18	4.3 Synopsis	16 days?	Thu 19-09-13	Thu 10-10-13	17
19	4.4 Procedural Design	7 days?	Tue 15-10-13	Wed 23-10-13	18
20	**Milestone**	1 day?	Fri 25-10-13	Fri 25-10-13	
21	**5.Coding Phase**	**75 days?**	**Fri 01-11-13**	**Thu 13-02-14**	
22	5.1 Implementation of basic unit	11 days?	Fri 01-11-13	Fri 15-11-13	
23	5.2 Implementation of final code	42 days?	Thu 21-11-13	Fri 17-01-14	22
24	5.3 GUI Design	19 days?	Mon 20-01-14	Thu 13-02-14	23
25	**Milestone**	1 day?	Fri 14-02-14	Fri 14-02-14	
26	**6. Testing**	**15 days?**	**Wed 19-02-14**	**Tue 11-03-14**	
27	6.1 Testing of Application	7 days?	Wed 19-02-14	Thu 27-02-14	
28	6.2 Fixing the Defects	6 days?	Tue 04-03-14	Tue 11-03-14	27
29	**Milestone**	1 day?	Wed 12-03-14	Wed 12-03-14	
30	**7. Evaluation**	**11 days?**	**Thu 13-03-14**	**Thu 27-03-14**	
31	7.1 Documentation	6 days?	Thu 13-03-14	Thu 20-03-14	
32	7.2 Project Evaluation	5 days?	Fri 21-03-14	Thu 27-03-14	31
33	**Milestone**	1 day?	Fri 28-03-14	Fri 28-03-14	

Figure 4.7 TimeLine Chart

CHAPTER 5

DESIGN

5.1 Architectural Design

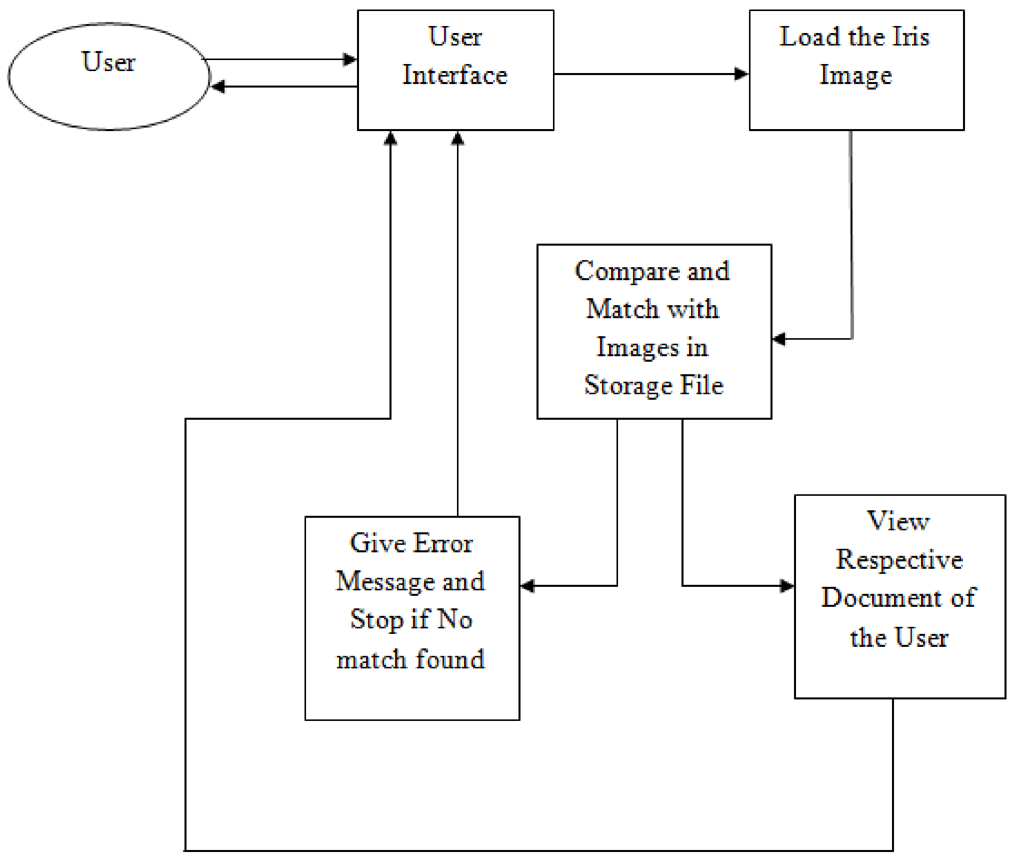


Figure 5.1 Architectural Flow Diagram

5.2 User Interface Design

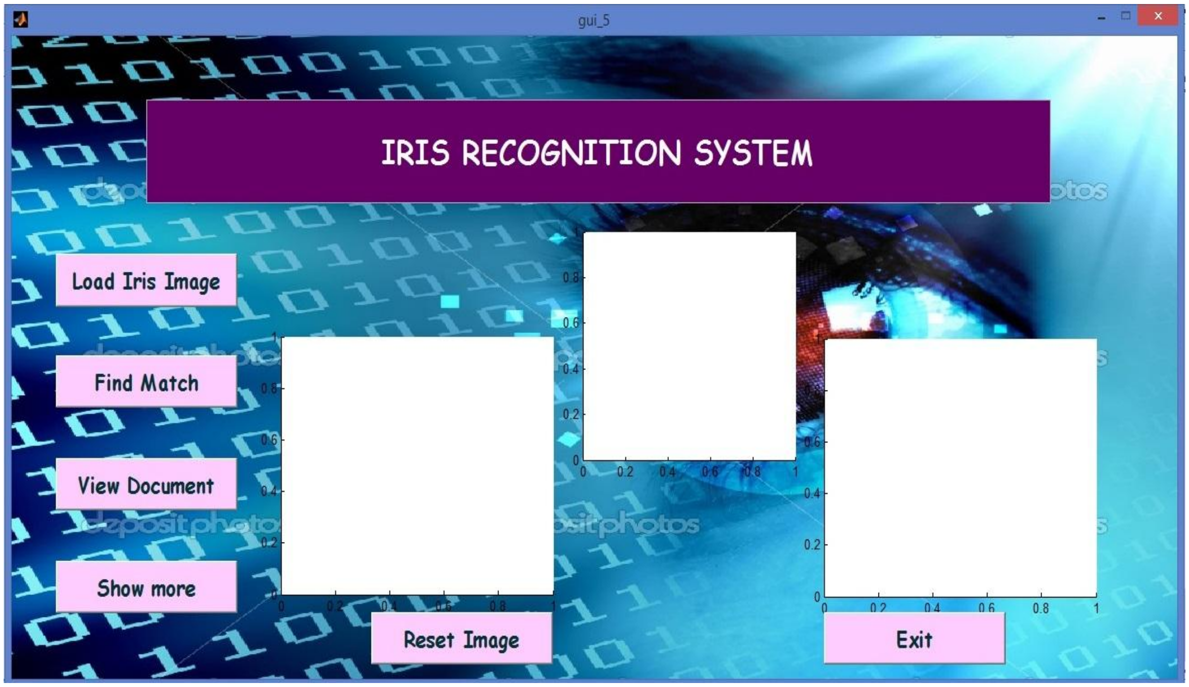


Figure 5.2 Snapshot 1 for UI

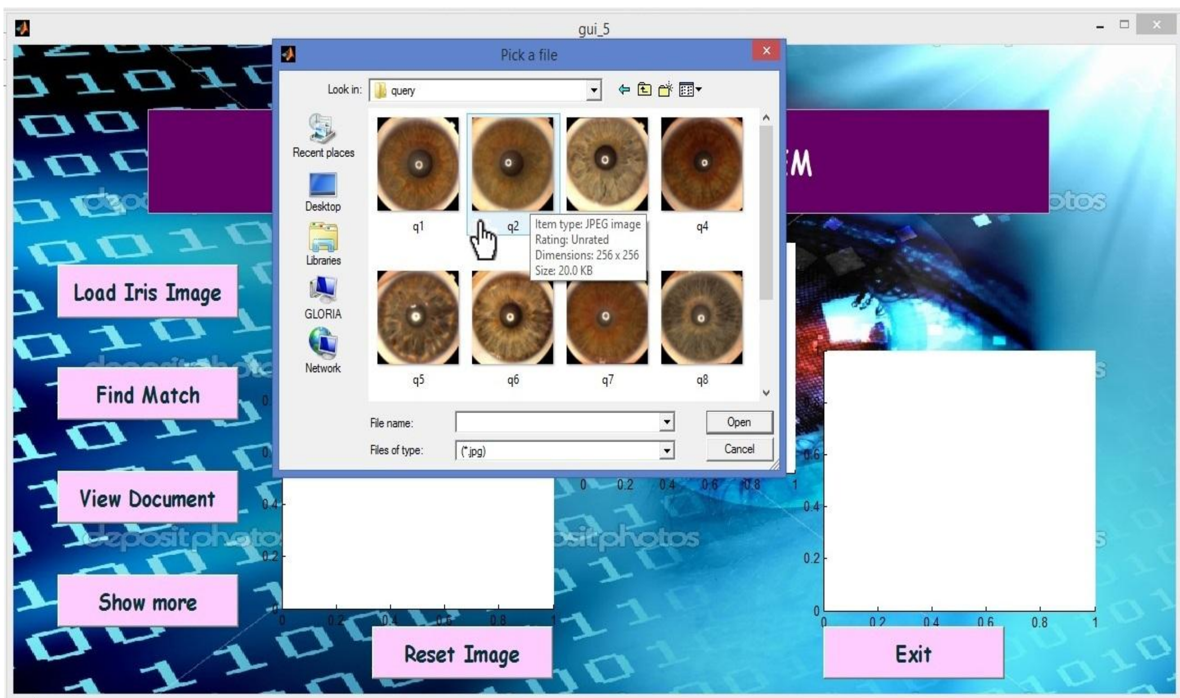


Figure 5.3 Snapshot 2 for UI (Browsing and selecting image)

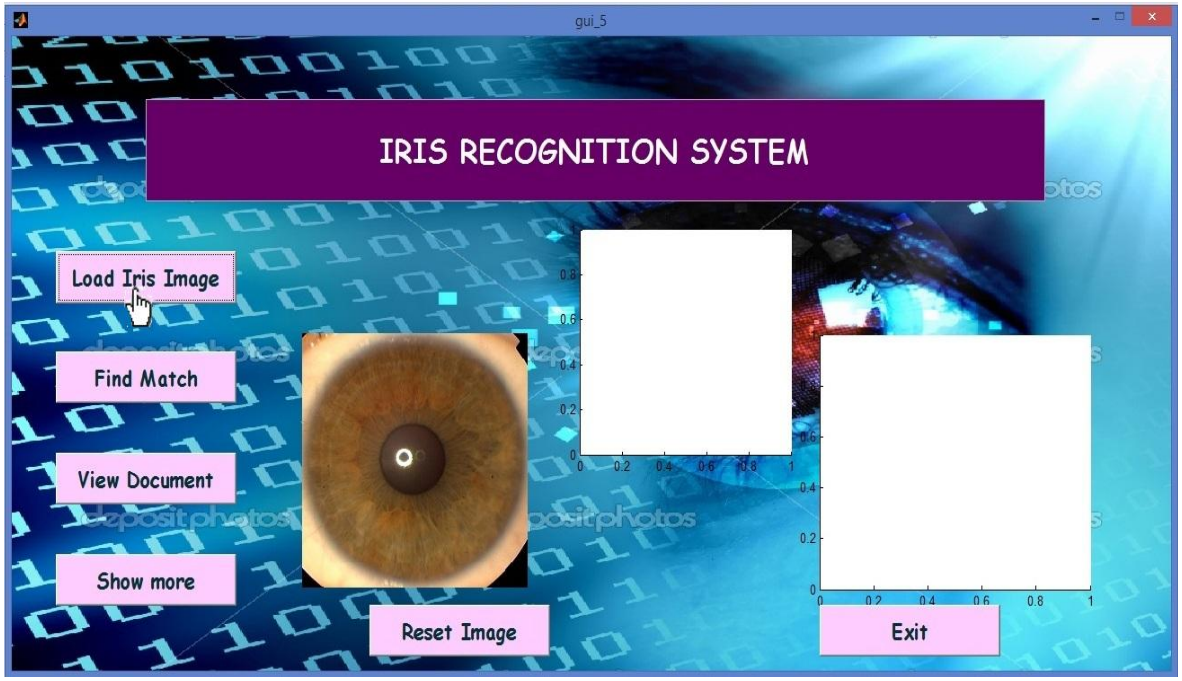


Figure 5.4 Snapshot 3 for UI (Loaded image)

Figure 5.5 Snapshot 4 for UI (Match Found)

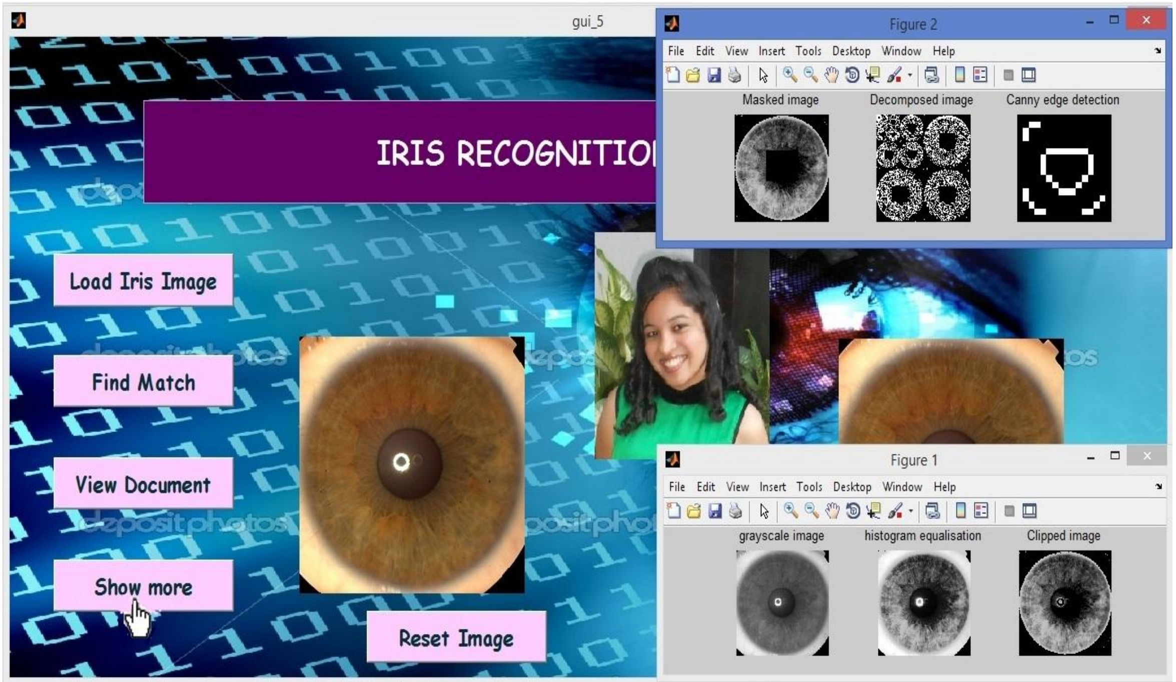


Figure 5.6 Snapshot 5 for UI (Viewing Intermediate steps)

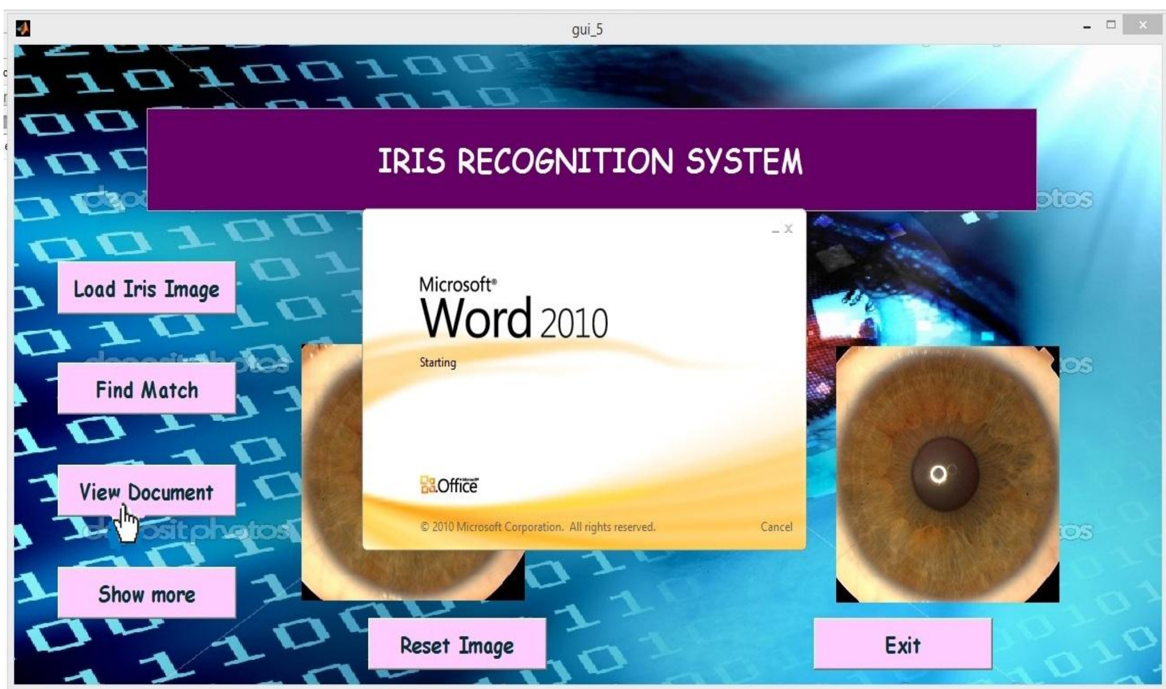


Figure 5.7 Snapshot 6 for UI (Opening the document in MS word)

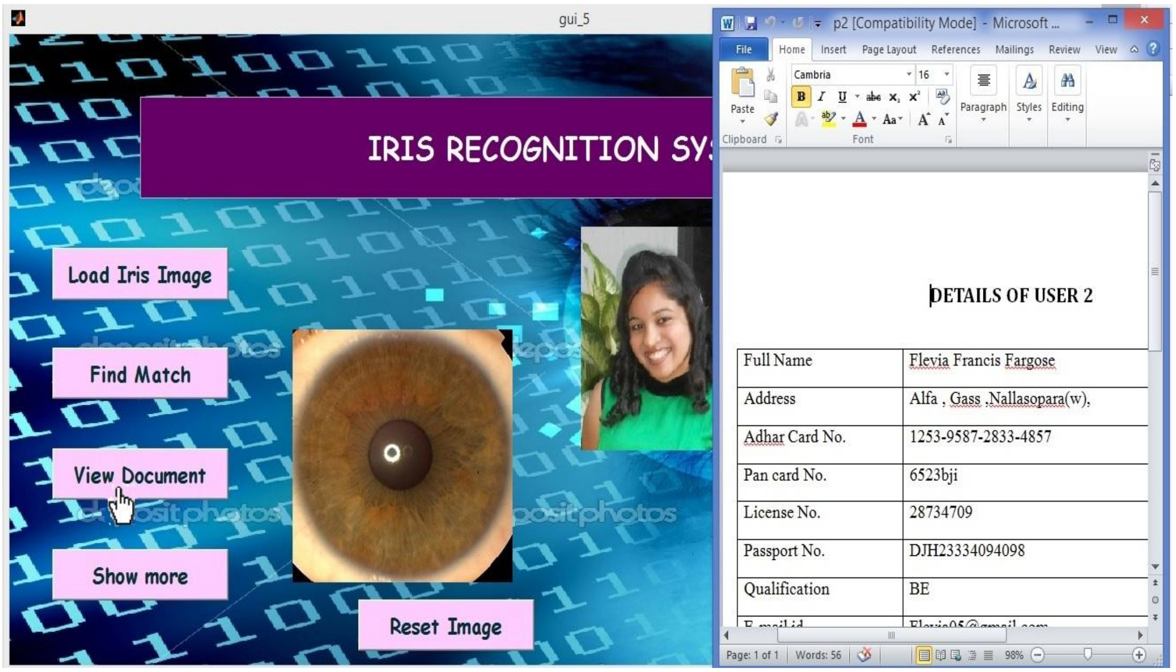


Figure 5.8 Snapshot 7 for UI (Accessing the document)

CHAPTER 6

IMPLEMENTATION

6.1 Hardware and Software Used

Hardware Requirements:

Table 6.1 Hardware Requirements

Processor	Intel Core i5
RAM	2Gb+
Hard Disk	100Gb+

Software Requirements:

Table 6.2 Software Requirements

Operating System	Windows 8
Mat lab	R2008

6.2 Algorithms / Methods Used

Algorithm for Processing:

1. Start.
2. Take the images from user.
3. Get the iris code of the image by extracting the feature vector.
4. Calculate the hamming distance between iris code of query image and that of images stored in file.
5. Sort and check for threshold.
6. If distance less than or equal to threshold (0.09) then match found and give access to document.

   Else

   Generate error message.
7. Stop.

Algorithm for feature vector Extraction:

1. Start.
2. Get image from System loaded by user.
3. Convert the RGB iris image into gray scale.
4. Equalize histogram of the gray scale iris image.
5. Perform clipping in order to localize actual iris.
6. Mask the clipped iris image and remove specular reflection.
7. Apply 2D-DWT up to 4^{th} energy level to decompose the image.
8. Perform canny edge detection algorithm to generate iris code.
9. Give the iris code to system.
10. Stop.

6.3 Working of the project (code)

- **feature _vector.m :**

```
function [v]=feature_vect(a)

b1=rgb2gray(a);

 c1=uint8(b1);
 h1=histeq(c1); % For Histogram equ

 [row col]=size(b1);
 C=row*col;

 for i=1:1:row     % for clipping
   for j=1:1:col
            if h1(i,j)>=0 && h1(i,j)<28
                     h1(i,j)=0;
            end
   end
 end

 for i=1:1:row
   for j=1:1:col
          if h1(i,j)>=200 && h1(i,j)<256
                   h1(i,j)=0;
          end
      end
```

```
end

M=imread('C:\Iris Recognition\Project 2014\process\Mask2.bmp');
M=uint8(M);

inw=h1.*M;
%figure,imshow(inw),title('output after masking')

[E F G H]=dwt2(inw,'db1');%1st level decomposition
first=[E F;G H];
final1=[E F;G H];

[I J K L]=dwt2(E,'db1');%2nd level decomposition
second=[I J;K L];
final2=[second F;G H];

[M N O P]=dwt2(I,'db1');%3RD level decomposition
third=[M N;O P];
mid=[third J;K L];
final3=[mid F;G H];

[Q R S T]=dwt2(M,'db1');%4th level decomposition
fourth=[Q R;S T];
mid41=[fourth N;O P];
mid42=[mid41 J;K L];
final4=[mid42 F;G H];

U=round(Q);

%figure,imshow(uint8(final1))
%figure,imshow(uint8(final2))
%figure,imshow(uint8(final3))
%figure,imshow(uint8(final4))

gb1=edge(Q,'canny');
%figure,imshow(gb1),title('canny')

v=gb1(:);

end
```

- **Output.m :**

```
function [image image1 person1 index_sdi1 sdi1]=Output1(g);
a=imread('C:\Iris Recognition\Project 2014\database\im1.jpg');
[v]=feature_vect(a);
v1=v;
a=imread('C:\Iris Recognition\Project 2014\database\im2.jpg');

[v]=feature_vect(a);
v2=v;

a=imread('C:\Iris Recognition\Project 2014\database\im4.jpg');
[v]=feature_vect(a);
v3=v;

a=imread('C:\Iris Recognition\Project 2014\database\im5.jpg');
[v]=feature_vect(a);
v4=v;

a=imread('C:\Iris Recognition\Project 2014\database\im7.jpg');
[v]=feature_vect(a);
v5=v;

a=imread('C:\Iris Recognition\Project 2014\database\im8.jpg');
[v]=feature_vect(a);
v6=v;

a=imread('C:\Iris Recognition\Project 2014\database\im10.jpg');
[v]=feature_vect(a);
v7=v;

a=imread('C:\Iris Recognition\Project 2014\database\im11.jpg');
[v]=feature_vect(a);
v8=v;

a=imread('C:\Iris Recognition\Project 2014\database\im13.jpg');
[v]=feature_vect(a);
v9=v;

a=imread('C:\Iris Recognition\Project 2014\database\im14.jpg');
[v]=feature_vect(a);
v10=v;
```

```
a=imread('C:\Iris Recognition\Project 2014\database\im16.jpg');
[v]=feature_vect(a);
v11=v;

a=imread('C:\Iris Recognition\Project 2014\database\im17.jpg');
[v]=feature_vect(a);
v12=v;

a=imread('C:\Iris Recognition\Project 2014\database\im19.jpg');
[v]=feature_vect(a);
v13=v;

a=imread('C:\Iris Recognition\Project 2014\database\im20.jpg');
[v]=feature_vect(a);
v14=v;

a=imread('C:\Iris Recognition\Project 2014\database\im22.jpg');
[v]=feature_vect(a);
v15=v;

a=imread('C:\Iris Recognition\Project 2014\database\im23.jpg');
[v]=feature_vect(a);
v16=v;

a=imread('C:\Iris Recognition\Project 2014\database\im25.jpg');
[v]=feature_vect(a);
v17=v;

a=imread('C:\Iris Recognition\Project 2014\database\im26.jpg');
[v]=feature_vect(a);
v18=v;

a=imread('C:\Iris Recognition\Project 2014\database\im28.jpg');
[v]=feature_vect(a);
v19=v;

a=imread('C:\Iris Recognition\Project 2014\database\im29.jpg');
[v]=feature_vect(a);
v20=v;

[v]=feature_vect(g);
v21=v;

y1=xlswrite('C:\Iris Recognition\Project 2014\process\irisxlsheet.xls',[v1 v2 v3 v4
v5 v6 v7 v8 v9 v10 v11 v12 v13 v14 v15 v16 v17 v18 v19 v20 v21]);
```

```
z1=xlsread('C:\Iris Recognition\Project 2014\process\irisxlsheet.xls');

 for j=1:20
 for i=1:256
         f = z1(i,j);
         g1 = v21(i,1);

         w1(i) = bitxor(f,g1);
 end
 di1(j)=sum(w1)/256;
 end

 [sdi1,index_sdi1]=sort(di1,'ascend');    %sorting of indexes as per hamming


 if (index_sdi1(1)==1 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im1.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\glancy.jpg');

 elseif (index_sdi1(1)==2 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im2.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\glancy.jpg');

 elseif (index_sdi1(1)==3 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im4.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\flevia.jpg');

 elseif (index_sdi1(1)==4 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im5.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\flevia.jpg');

 elseif (index_sdi1(1)==5 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im7.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\olivia.jpg');

 elseif (index_sdi1(1)==6 && sdi1(1)<= 0.09)
         image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
         image1=imread('C:\Iris Recognition\Project 2014\database\im8.jpg');
         person1=imread('C:\Iris Recognition\Project 2014\person\olivia.jpg');

 elseif (index_sdi1(1)==7 && sdi1(1)<= 0.09)
```

```
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg') ;
        image1=imread('C:\Iris Recognition\Project 2014\database\im10.jpg');
        person1=imread('C:\Iris Recognition\Project2014\person\flavian.jpg');

elseif (index_sdi1(1)==8 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im11.jpg');
        person1=imread('C:\Iris Recognition\Project2014\person\flavian.jpg');

elseif (index_sdi1(1)==9 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im13.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\stalen.jpg');

elseif (index_sdi1(1)==10 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im14.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\stalen.jpg');

elseif (index_sdi1(1)==11 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im16.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\ruchita.jpg');

elseif (index_sdi1(1)==12 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im17.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\ruchita.jpg');

elseif (index_sdi1(1)==13 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im19.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\sarah.jpg');

elseif (index_sdi1(1)==14 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg') ;
        image1=imread('C:\Iris Recognition\Project 2014\database\im20.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\sarah.jpg');

elseif (index_sdi1(1)==15 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im22.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\gladwin.jpg');

elseif (index_sdi1(1)==16 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
```

```
        image1=imread('C:\Iris Recognition\Project 2014\database\im23.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\gladwin.jpg');

elseif (index_sdi1(1)==17 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im25.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\viviana.jpg');

elseif (index_sdi1(1)==18 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im26.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\viviana.jpg');

elseif (index_sdi1(1)==19 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im28.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\ansley.jpg');


elseif (index_sdi1(1)==20 && sdi1(1)<= 0.09)
        image=imread('C:\Iris Recognition\Project 2014\process\Match.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\database\im29.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\ansley.jpg');

else
        image=imread('C:\Iris Recognition\Project 2014\process\Nomatch.jpg');
        image1=imread('C:\Iris Recognition\Project 2014\process\Error.jpg');
        person1=imread('C:\Iris Recognition\Project 2014\person\unknown.jpg');
end
end
```

- **Gui.m:**

```
function varargout = gui_5(varargin)

% GUI_5 M-file for gui_5.fig
% GUI_5, by itself, creates a new GUI_5 or raises the existingsingleton*.
% H = GUI_5 returns the handle to a new GUI_5 or the handle to
% the existing singleton*.
%
% GUI_5('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in GUI_5.M with the given input arguments.
%
% GUI_5('Property','Value',...) creates a new GUI_5 or raises the
% existing singleton*. Starting from the left, property value pairs are
% applied to the GUI before gui_5_OpeningFcn gets called. An
% unrecognized property name or invalid value makes property application
% stop. All inputs are passed to gui_5_OpeningFcn via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help gui_5

% Last Modified by GUIDE v2.5 09-Apr-2014 13:23:10

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
'gui_Singleton',  gui_Singleton, ...
'gui_OpeningFcn', @gui_5_OpeningFcn, ...
'gui_OutputFcn',  @gui_5_OutputFcn, ...
'gui_LayoutFcn',  [] , ...
'gui_Callback',   []);
ifnargin&&ischar(varargin{1})
gui_State.gui_Callback = str2func(varargin{1});
end

ifnargout
 [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```
% --- Executes just before gui_5 is made visible.
function gui_5_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_5 (see VARARGIN)
handles.a1='';
handles.a2='';
handles.a3='';

guidata(hObject,handles);
% Choose default command line output for gui_5
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_5 wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
functionvarargout = gui_5_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
ah = axes('unit', 'normalized', 'position', [0 0 1 1]);
% import the background image and show it on the axes
bg = imread('C:\Iris Recognition\Project 2014\back ground GUI\bg2.jpg');
imagesc(bg);
%prevent plotting over the background and turn the axis off
set(ah,'handlevisibility','off','visible','off')
% making sure the background is behind all the other uicontrols
uistack(ah, 'bottom');

% Update handles structure
guidata(hObject, handles);


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
ifispc&&isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile('*.jpg', 'Pick a file');
ifisequal(filename,0) || isequal(pathname,0)
msgbox(sprintf('User has Pressed Cancel'),'Error','Error');
else
filename=strcat(pathname,filename);
%     pathname
 a=imread(filename);
axes(handles.axes1)
imshow(a)
end
handles.a1=a;
guidata(hObject,handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1)
clareset
axes(handles.axes2)
clareset
axes(handles.axes4)
clareset
```

```
handles.a1='';

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
try

i=handles.a1;
[m m1 p1 index dist]=Output1(i);
y2=xlswrite('C:\Iris Recognition\Project 2014\process\distance.xls',[dist]);
handles.a2=index;
guidata(hObject,handles);

axes(handles.axes2);
imshow(m1);
axes(handles.axes4);
imshow(p1);

figure,imshow(m),title('match found????');
catch
msgbox(sprintf('Please Load Iris Image!!!!'),'Error','Error');
end


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
try
i1=handles.a1;
[im2 im3 im4 im5 im6 im7]=F_v(i1);
figure;

subplot(1,3,1),imshow(im2),title('grayscale image');
subplot(1,3,2),imshow(im3),title('histogram equalisation');
subplot(1,3,3),imshow(im4),title('Clipped image');

figure;
subplot(1,3,1),imshow(im5),title('Masked image');
subplot(1,3,2),imshow(im6),title('Decomposed image');
subplot(1,3,3),imshow(im7),title('Canny edge detection');
catch
msgbox(sprintf('Please Load Iris Image!!!!'),'Error','Error');
```

```
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
try
        ind=handles.a2;
         d=xlsread('C:\Iris Recognition\Project 2014\process\distance.xls');
        if((ind(1,1)==1 || ind(1,1)==2) && d(1,1)<=0.09)
        open('p1.doc');
        elseif((ind(1,1)==3 || ind(1,1)==4) && d(1,1)<=0.09)
        open('p2.doc');
        elseif((ind(1,1)==5 || ind(1,1)==6) && d(1,1)<=0.09)
        open('p3.doc');
        elseif((ind(1,1)==7 || ind(1,1)==8) && d(1,1)<=0.09)
        open('p4.doc');
        elseif((ind(1,1)==9 || ind(1,1)==10) && d(1,1)<=0.09)
        open('p5.doc');
        elseif((ind(1,1)==11 || ind(1,1)==12) && d(1,1)<=0.09)
        open('p6.doc');
        elseif((ind(1,1)==13 || ind(1,1)==14) && d(1,1)<=0.09)
        open('p7.doc');
        elseif((ind(1,1)==15 || ind(1,1)==16) && d(1,1)<=0.09)
        open('p8.doc');
                elseif((ind(1,1)==17 || ind(1,1)==18) && d(1,1)<=0.09)
        open('p9.doc');
        elseif((ind(1,1)==19 || ind(1,1)==20) && d(1,1)<=0.09)
        open('p10.doc');
        else
        msgbox(sprintf('You are not valid user!!!'),'Error','Error');
        end
catch
msgbox(sprintf('You Can not access the file. Either you have not loaded the
image or have not processed it!!!'),'Error','Error');
end

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
close all
```

CHAPTER 7
RESULTS AND DISCUSSIONS

In our system on Iris Recognition the following results were observed:

- The user could successfully load iris image from the specified location.
- If the user is valid and loads a correct query image then we observed that he can view and access his own document consisting his important details as shown in fig. 7.1
- If the user is not a valid user then we observed that access to the document is not given as shown in fig.7.2

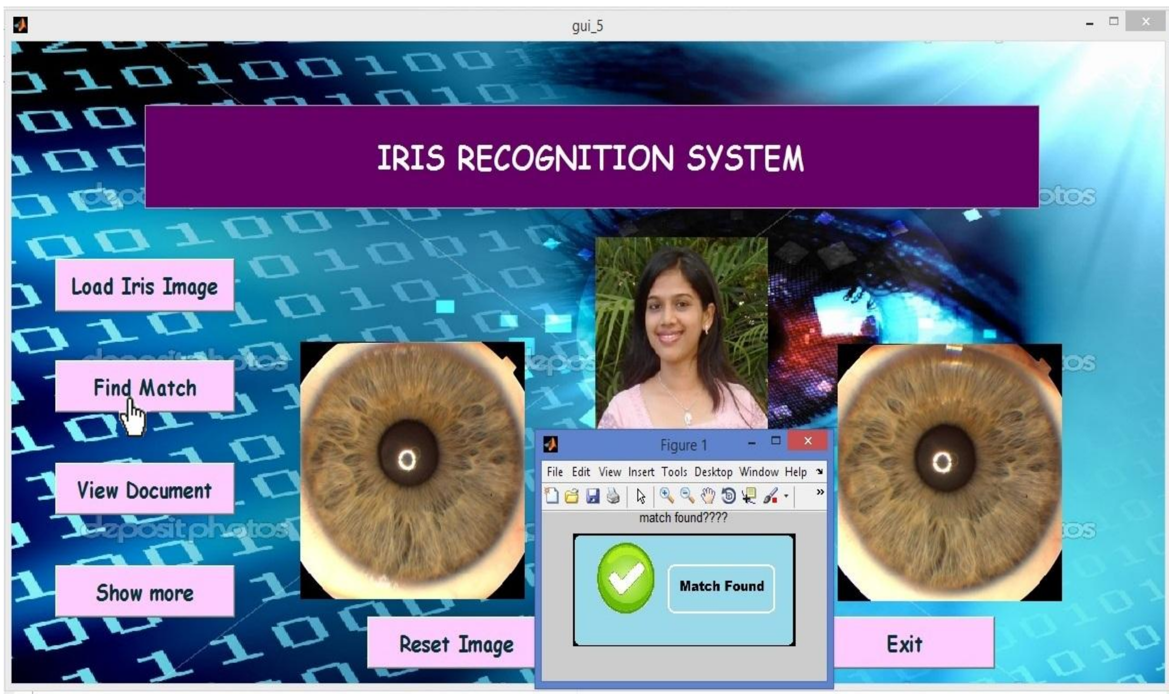


Figure 7.1 Result of match found for valid user

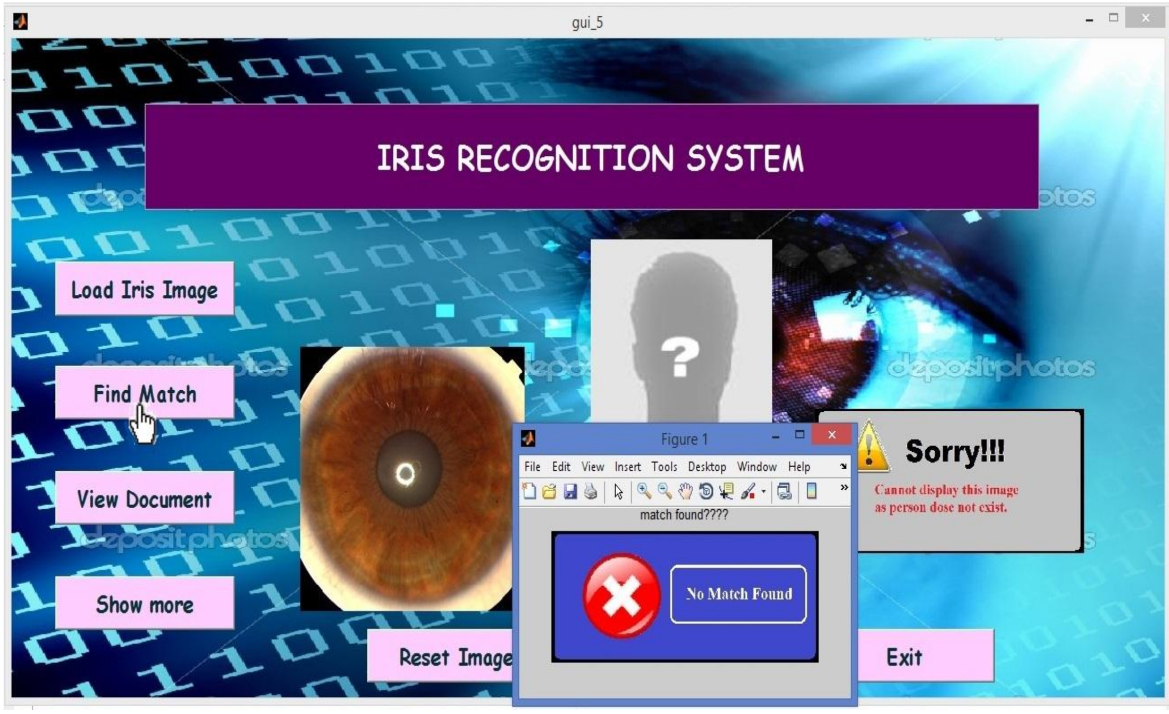


Figure 7.2 Result of no match found for invalid user

- The analysis of this system is carried out using two parameters i.e. FAR and FRR.
- FAR for this system is 0.05 and FRR is 0.95.
- The overall accuracy is 95%.

CHAPTER 8

TESTING

8.1 Test cases:

The various test cases have been derived and tested by the various persons involved in the implementation of the application. Test cases helps to find out the errors or flaws in the software and thus correct it and improve the software quality .Some of the test cases that we have applied are described below with the various modules.

User Interface is one of the important modules in our project. User Interface provides flexible interaction and reduces the user's memory load. Some of the various test cases involved in user Interface module are as follows:

1. **Test case 1:**

Figure 8.1 Test case 1(Matching for invalid user)

Figure 8.2 Test case 1(Viewing document for invalid user)

Table 8.1 Description of test case 1

Test Case ID	1
Module	Wrong Query image
Test Data	Index at first position, threshold
Expected Results	If a query image of a person whose image does not exist in the database is entered an error message should be displayed and no access should be granted for the invalid user.
Actual Results	When query image of an invalid user is entered an error message is displayed

2. Test case 2:

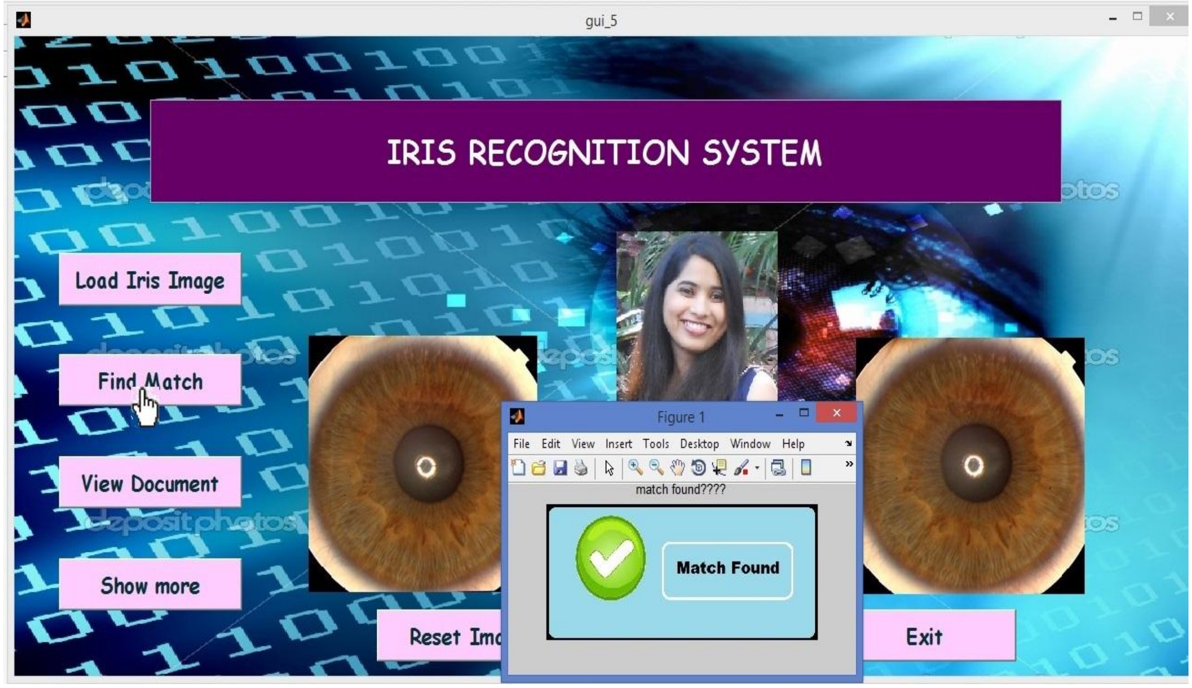


Figure 8.3 Test case 2 (Matching for valid user)

Table 8.2 Description of test case 2

Test Case ID	2
Module	valid input
Test Data	Iris image
Expected Results	If the user clicks on find match the iris image of that particular person should be displayed.
Actual Results	If the user tries to find match of a valid iris image with the storage file output is displayed.

3. Test case 3:

Figure 8.4 Test case 3(Incomplete Input)

Table 8.3 Description of test case 3

Test Case ID	3
Module	Incomplete input
Test Data	Iris image
Expected Results	If the user clicks on find match or any other option before loading the iris image an error message should be displayed suggesting user to load his iris image.
Actual Results	If the user tries to find match with the storage file images without entering the iris image an error message is been displayed.

4. Test case 4:

Figure 8.5 Test case 4 (Viewing document without any input image)

Table 8.4 Description of test case 4

Test Case ID	4
Module	Viewing document without any input image
Test Data	Iris image
Expected Results	If the user clicks on view document before loading the iris image an error message should be displayed suggesting user to load his iris image.
Actual Results	If the user tries to directly access the document without entering the iris image an error message is been displayed.

5. Test case 5:

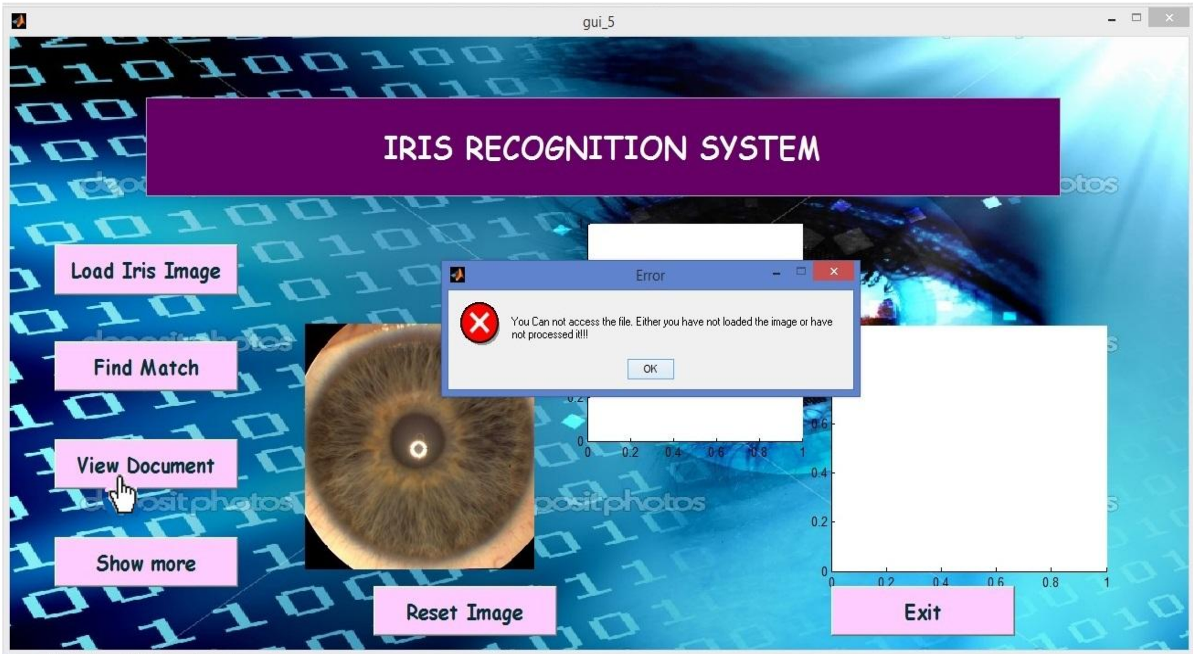


Figure 8.6 Test case 5 (Viewing document without finding match)

Table 8.5 Description of test case 5

Test Case ID	5
Module	View document without finding match
Test Data	Find match (index, distance)
Expected Results	If the user tries to access the document without finding match between his entered iris image and all storage file images the system should display an error message suggesting the user to find match.
Actual Results	An error message is displayed when a user wants to access the document without finding match.

8.2 Type of Testing used:

1. **Unit Testing:**

   The testing plan that we have implemented in our software as time progresses is described further. Unit testing tests if the smallest units of software are working satisfactorily. This testing started with the coding itself. It was conducted on the basic modules of the system.

   a. **Loading Image:** The query image of a person was successfully loaded.

   b. **Finding Matched:** It was found when a query image is been loaded, if its exact match is present than the image from the database is uploaded else error message is displayed.

   c. **Opening document:** If the person uploads the valid iris image, and if the match is present in the database then the document of only that specific person opens.

2. **Integration testing:**

   Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing, Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

   After creating individual units, testing them separately and debugging errors that arose we integrated these individual units as one and tested them.

CHAPTER 9
CONCLUSION

As the technology is growing continuously Security in every aspect has a very important role to play. Iris of a person is such a unique entity which will always remain constant and will never change. Iris recognition is proven to be very useful and versatile security measure. It is a quick and accurate way of identifying an individual with no chance for human error.

Iris recognition is widely used in transportation industry and can have many applications in other fields where security is necessary. Using Iris as the password in order to provide security user documents are authenticated.

CHAPTER 10

FUTURE SCOPE

- Real time database can be used by capturing the images of the users at a particular time, and processing can be done on the same in order to extract the features of the person.
- An application can be made such as Client and Server where the document that is to be authenticated is present on the Server, when the client gives the Iris Image, after the match is found the document can be retrieved.
- This System can be further extended as we can provide our iris as password to open the Laptop or PCs.
- Our system can be integrated with any other security system in order to provide more security.

APPENDIX

1. **HD – Hamming Distance:**

$$HM = \frac{length(code)}{\sum_{i=1}^{length(code)} (1/code(i))}$$

$$HD = \frac{1}{N} \sum_{i=1}^{N} X_I \otimes Y_I$$

2. **FAR – False Acceptance Rate:**

FAR = Number of false acceptant / Number of attempts

3. **FRR – False Rejection Rate**

FRR = Number of false rejectant / Number of attempts

GLOSSARY

- DFD – Data Flow Diagram
- 2D DWT – Two Dimensional Discrete Wavelet Transform
- UML – Universal Modelling Language
- UID – User Interface Design
- RGB – Red Green Blue

BIBLIOGRAPHY AND REFERENCES

[1] Khattab M. Ali Alheeti," Biometric Iris Recognition Based On Hybrid Technique", *International Journal On Soft Computing (IJSC)* Vol.2, No.4, November 2011.

[2] Vanaja Roselin. E. Chirchi, Dr. L.M. Waghmare, E. R. Chirchi," Iris Biometric Recognition For Person Identification In Security Systems", *International Journal Of Computer Applicati*ons (0975 – 8887)Volume 24– No.9, June 2011.

[3] Keyur B. Shah, Jayesh Solanki, " Biometric Security Using Iris Patterns For Sustainable Growth", Lecturer, *Shri Chimanbhai Patel Institute of Computer Applications, And Associate Professor, L.D. Arts College, Ahmedabad.*

[4] Vishnu Naresh Boddeti, B. V. K. Vijaya Kumar," Extended-Depth-Of-Field Iris Recognition Using Unrestored Wavefront-Coded Imagery", *IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans,* Vol. 40, No. 3, May 2010.

[5] Sreelekha.K.V1, Jackulin Mahariba," A Human Iris Recognition Using Haar Wavelet Decomposition", *Department Of Computer Science And Engineering, SRM University, Chennai.*

[6] Arun Ross,"Iris Recognition: The Path Forward", *IEEE Computer Society* 0018-9162/10/$26.00 © 2010 IEEE.

[7] Surjeet Singh, Kulbir Singh," Segmentation Techniques For Iris recognition System", *International Journal Of Scientific & Engineering Research* Volume 2, Issue 4, April-2011.

[8] John Daugman," New Methods In Iris Recognition", *IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics,* Vol. 37, No. 5, October 2007.

[9] http://www.projectiris.co.uk/

[10] http://phoenix.inf.upol.cz/iris/download/