BIOMETRIC SYSTEM: IRIS BASED RECOGNITION AND AUTHENTICATION USING COMPUTER VISION

A Major Project Report

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

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

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

AMIREDDY VINUTHNA (Roll No: 18BT1A0501)

Under the Guidance of
Mrs. M. Sangeetha
Assistant Professor (CSE)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VISVESVARAYA COLLEGE OF ENGINEERING & TECHNOLOGY
(Recognized by A.I.C.T.E Affiliated to JNTU, Hyderabad),
(MP Patelguda, Ibrahimpatnam, Telangana, INDIA - 501510)

2021 – 2022

Ph : 08414 - 252677
(O) : 77299 89434
Website : www.vcethyd.ac.in
e-mail : vcethr@gmail.com

Visvesvaraya

College of Engineering & Technology

(Approved by AICTE, New Delhi & Govt. of T.S., Affiliated to JNTUH, Hyderabad.)
(Sponsored by : Jawahar Educational Society) (An ISO 9001 - 2018 Certified Institution)
M.P. Patelguda (V), Ibrahimpatnam (M), Ranga Reddy Dist - 501510, T.S.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that this project report entitled **BIOMETRIC SYSTEM: IRIS BASED RECOGNITION AND AUTHENTICATION USING COMPUTER VISION** being submitted by

AMIREDDY VINUTHNA (18BT1A0501)

In partial fulfillment of the requirements for the award of the degree **Bachelor of Technology in Computer Science & Engineering** to the Jawaharlal Nehru Technological University, Hyderabad, during the academic year 2021- 2022, is a bonafide record of work carried out under our guidance and supervision.

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

Mrs. M. Sangeetha	**Dr. L. Kiran Kumar Reddy**
Assistant Professor	Head of the Department
Internal Guide	Dept. of CSE

Internal Examiner	External Examiner

Visvesvaraya
College of Engineering & Technology
M.P.Patelguda (V), Ibrahimpanam (M), Ranga Reddy Dist- 501510

DECLARATION

I here by declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name : AMIREDDY VINUTHNA
Roll No. : 18BT1A0501

ACKNOWLEDGEMENT

It gives me a great sense of pleasure to present the report of the project undertaken during B.Tech. I would like to express my special thanks to the Principal & Professor **Dr. D. Ramesh** for moral support and **College Management** of Visvesvaraya College of Engineering & Technology, Hyderabad for providing me infrastructure to complete the project.

I owe special debt of gratitude to **Dr. N Raji Reddy,** Dean Academics of Visvesvaraya College of Engineering & Technology, Hyderabad for his constant support and guidance throughout the course of my work.

I thank **Dr. L. Kiran Kumar Reddy** Head of the Department of Computer Science & Engineering for his constant support and cooperation.

I take this opportunity to acknowledge the contribution of the Major Project Coordinator **Mr. V. Hirish Reddy,** Assistant Professor, Department of Computer Science & Engineering, for his full support and assistance during the development of the project.

I also convey thanks to my internal Project Guide **Mrs .M. Sangeetha,** Assistant Professor (CSE) Visvesvaraya College of Engineering & Technology, Hyderabad for her guidance throughout the course of my work. It is only her cognizant efforts that my endeavours have seen light of the day.

I would also like to express my gratitude towards my Parents for their kind co-operation and encouragement which helped me in completion of this project.

I do not want to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of my project. Last but not the least, I acknowledge my friends for their contribution in the completion of the project.

AMIREDDY VINUTHNA **Roll No.:** 18BT1A0501

TABLE OF CONTENTS

CONTENT	PAGE NO.
ABSTRACT	**i**
LIST OF FIGURES	**ii**
LIST OF SCREENS	**iii**
1. INTRODUCTION	**1**
1.1 Domain Description	2
1.2 About Project	7
1.2.1 Problem Definition	7
1.2.2 Proposed Solution	7
1.3 Objective	8
2. BIOMETRIC SYSTEMS SURVEY	**9**
2.1 Theoretical Background	9
2.2 Existing System	11
2.3 Proposed System	11
2.4 Advantages of proposed system	11
2.5 Feasibility Study	12
2.5.1 Operational Feasibility	12
2.5.2 Technical Feasibility	12
2.5.2.1 Survey of Technology	12
2.5.2.2 Feasibility of Technology	12
2.5.3 Economic Feasibility	12

3. SYSTEM ANALYSIS 13

3.1 Specifications 13

3.1.1 Functional Requirements 13

3.1.2 Non-Functional Requirements 13

3.2 Software Requirements 14

3.3 Hardware Requirements 14

3.4 Module Description 15

4. SOFTWARE DESIGN 20

4.1 Introduction 20

4.2 Block Diagram 21

4.3 Data Flow Diagrams 21

4.3.1 DFD Symbols 22

4.3.2 Level-0 DFD 23

4.3.4 Level-1 DFD 23

4.3.5 Level-2 DFD 24

4.4 Unified Modelling Language 24

4.4.1 Class Diagram 25

4.4.2 Object Diagrams 25

4.4.3 Use Case Diagram 25

4.4.4 Interaction Diagrams 26

4.4.4.1Sequence Diagram 26

4.4.4.2 Collaboration Diagram 26

4.4.5 Activity Diagram 26

4.5 Control Flow Diagrams 27

4.6 Database Design 28

4.6.1 ER Diagrams 28

5. IMPLEMENTATION 29

5.1 Technology used 29

5.2 Code Templates 33

6. TESTING 36

6.1 Introduction 36

6.2 Design of Test cases and Scenarios 36

6.2.1 Black Box Testing 36

6.2.2 White Box Testing 37

6.2.3 Test Cases 37

7. OUTPUT SCREENS 39

8. CONCLUSION 44

9. FUTURE ENHANCEMENT 45

10. BIBLIOGRAPHY 46

ABSTRACT

One of the most important modules of computer systems is the one that is responsible for user safety. It was proven that simple passwords and logins cannot guarantee high efficiency and are easy to obtain by the hackers. The well-known alternative is identity recognition based on biometrics. Biometrics can be described as the technology used in identifying or authenticating an individual based on their unique biological features such as iris, signature, and voice. In a biometric system a person is identified automatically by processing the unique features that are posed by the individual.

Iris Recognition is regarded as the most reliable and accurate biometric identification systems available. In Iris Recognition a person is identified by the iris which is the part of eye using pattern matching or image processing. The aim is to identify a person in real time, with high efficiency and accuracy by analysing the random patters visible within the iris. In this project I have designed an IRIS BIOMETRIC SYSTEM, that uses computer vision technology to do facial recognition and to capture the IRIS of an individual, and store their information in a database for biometric authentication. The DLIB library is used to detect and recognize the face structure to pinpoint the eyes, and then capture the eyes through openCV's video capturing functions. The captured IRIS is stored into an image along with input from the user. During verification an associated ID inputted by the user is entered when verifying the IRIS. The IRIS is captured through the video feed, and sent to a pattern feature-like matching system setup using ORB's detector. The detector does pattern matching of both the user's IRIS through the feed, and the stored image of the user's IRIS from the keypoints and the authentication is done only if the match is found.

LIST OF FIGURES

Figure Number:	**Name of Figures:**	**Page Number:**
1.1	Iris Recognition	1
1.1.1	Computer Vision	2
1.1.2	Working of Computer Vision	3
1.1.3	CV in Self-Driving Cars	3
1.1.4	CV in Face Recognition	4
1.1.5	CV in AR&MR	4
1.1.6	CV in Healthcare	5
1.1.7	CV Tasks	5
2.1.1	Basic Criteria of biometric Systems	10
2.1.2	Types of Biometric Systems	10
3.4.1	Facial Landmarks using DLib	17
4.2.1	Block Diagram	21
4.3.2.1	Level-0 DFD	23
4.3.3.1	Level-1 DFD	23
4.3.4.1	Level-2 DFD	24
4.4.3.1	Use Case Diagram	26
4.5.1	Control Flow Diagram	27
4.6.1	Registered users Database Design	28
4.6.2	E-R Diagram	28

LIST OF SCREENS

Figure Number:	Name of Figures:	Page Number:
5.2.1	Register Iris Template	34
5.2.2	Iris Matching using ORB detection	34
5.2.3	Verify Users Template	35
5.2.4	DLIB Face detection Template	35
5.2.5	Database Template	35
7.1	User Interface for Iris Biometric System	39
7.2	On clicking Register button	39
7.3	Entering users Information	40
7.4	On clicking Verify button	40
7.5	Verification through user id	41
7.6	Iris match found case	41
7.7	User Information	41
7.8	Iris match not found case	42
7.9	Registered users Database	42
7.10	Registered users info in database	43

CHAPTER 1

INTRODUCTION

A biometric system provides automatic recognition of an individual based on some sort of unique feature or characteristic possessed by the individual. Biometric systems have been developed based on fingerprints, hand geometry, voice, facial features, handwriting, the retina, and the IRIS. Iris recognition is a type of biometric technology that enables a person to be authenticated automatically via his criteria through machines. Compared to other biometric traits, iris possess lesser false acceptance and rejection rate due to its high in universality, uniqueness, permanence, permanency, and stability. It is an automated method of biometric identification that uses mathematical pattern-recognition techniques on video images of one or both of the irises of an individual's eyes, whose complex patterns are unique, stable, and can be seen from some distance.

IRIS RECOGNITION TECHNOLOGY

The iris is an externally visible, yet protected organ whose unique epigenetic pattern remains stable throughout adult life.These characteristics make it very attractive for use as a biometric for identifying individuals. Image processing techniques can be employed to extract the unique iris pattern from a digitized image of the eye, and encode it into a biometric template, which can be stored in a repository database.

The biometric template contains an objective mathematical representation of the unique information stored in the iris, and it allows comparisons to be made between templates. When a subject wishes to be identified by iris recognition system, their eye is first photographed, and then a template created for their iris region. This template is then compared with the other templates stored in a database until either a matching template is found and the subject is identified,or no match is found and the subject remains unidentified.Although prototype systems had been proposed earlier,it was not until the early nineties that Cambridge researcher, John Daugman, implemented a working automated iris recognition systems. The Daugman system is patented and the rights are now owned by the company Iridian Technologies. It is the most successful and most well-known, and also many other systems have been developed. Other notable ones include the systems of Wilde.

Figure 1.1: Iris Recognition

1.1 DOMAIN DESCRIPTION

What is Computer Vision ?

Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Figure 1.1.1: Computer Vision

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image. Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive and the market is continuing to grow. It is expected to reach USD 48.6 billion by 2022.

How does computer vision work?

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.
Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN).

Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will “look” at the data and teach itself to tell one image from another. Algorithms enable the machine to learn by itself, rather than someone programming it to recognize an image.

A CNN helps a machine learning or deep learning model “look” by breaking images down into pixels that are given tags or labels. It uses the labels to perform convolutions (a mathematical operation on two functions to produce a third function) and makes predictions about what it is “seeing.” The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions start to come true. It is then recognizing or seeing images in a way similar to humans.

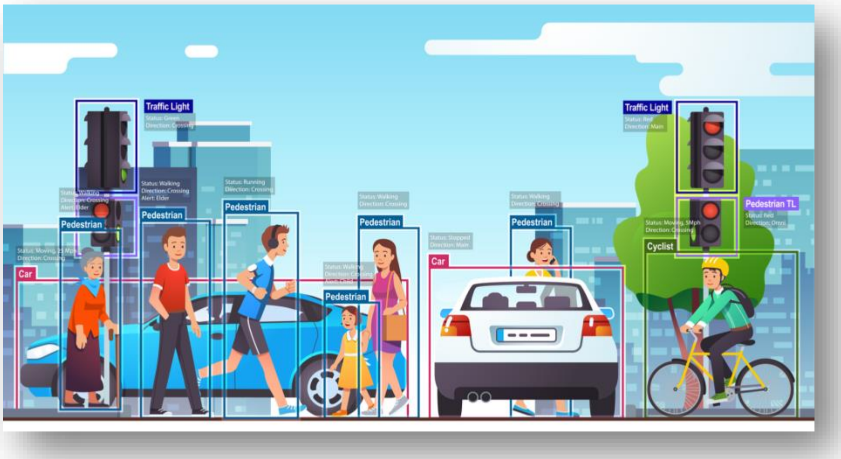


Figure 1.1.2: Working of Computer Vision

Much like a human making out an image at a distance, a CNN first discerns hard edges and simple shapes, then fills in information as it runs iterations of its predictions. A CNN is used to understand single images. A recurrent neural network (RNN) is used in a similar way for video applications to help computers understand how pictures in a series of frames are related to one another.

Applications Of Computer Vision

Computer vision is one of the areas in Machine Learning where core concepts are already being integrated into major products that we use every day.

- **CV In Self-Driving Cars**

But it’s not just tech companies that are leverage Machine Learning for image applications.

Figure 1.1.3: CV in Self-Driving Cars

Computer vision enables self-driving cars to make sense of their surroundings. Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians. The self-driving car can then steer its way on streets and highways, avoid hitting obstacles, and (hopefully) safely drive its passengers to their destination.

- **CV In Facial Recognition**

Computer vision also plays an important role in facial recognition applications, the technology that enables computers to match images of people's faces to their identities. Computer vision algorithms detect facial features in images and compare them with databases of face profiles. Consumer devices use facial recognition to authenticate the identities of their owners. Social media apps use facial recognition to detect and tag users. Law enforcement agencies also rely on facial recognition technology to identify criminals in video feeds.

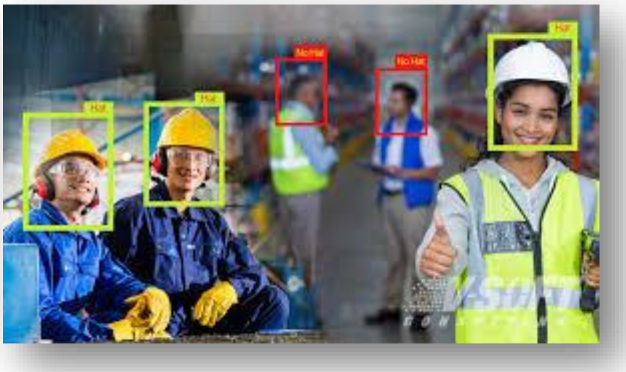


Figure 1.1.4: CV in Face Recognition

- **CV In Augmented Reality & Mixed Reality**

Computer vision also plays an important role in augmented and mixed reality, the technology that enables computing devices such as smartphones, tablets and smart glasses to overlay and embed virtual objects on real world imagery. Using computer vision, AR gear detect objects in real world in order to determine the locations on a device's display to place a virtual object. For instance, computer vision algorithms can help AR applications detect planes such as tabletops, walls and floors, a very important part of establishing depth and dimensions and placing virtual objects in physical world.

Figure 1.1.5: CV in AR&MR

- **CV In Healthcare**

Computer vision has also been an important part of advances in health-tech. Computer vision algorithms can help automate tasks such as detecting cancerous moles in skin images or finding symptoms in x-ray and MRI scans.

Figure 1.1.6: CV in Healthcare

Challenges of Computer Vision

Helping computers to see turns out to be very hard. Inventing a machine that sees like we do is a deceptively difficult task, not just because it’s hard to make computers do it, but because we’re not entirely sure how human vision works in the first place.

Studying biological vision requires an understanding of the perception organs like the eyes, as well as the interpretation of the perception within the brain. Much progress has been made, both in charting the process and in terms of discovering the tricks and shortcuts used by the system, although like any study that involves the brain, there is a long way to go.

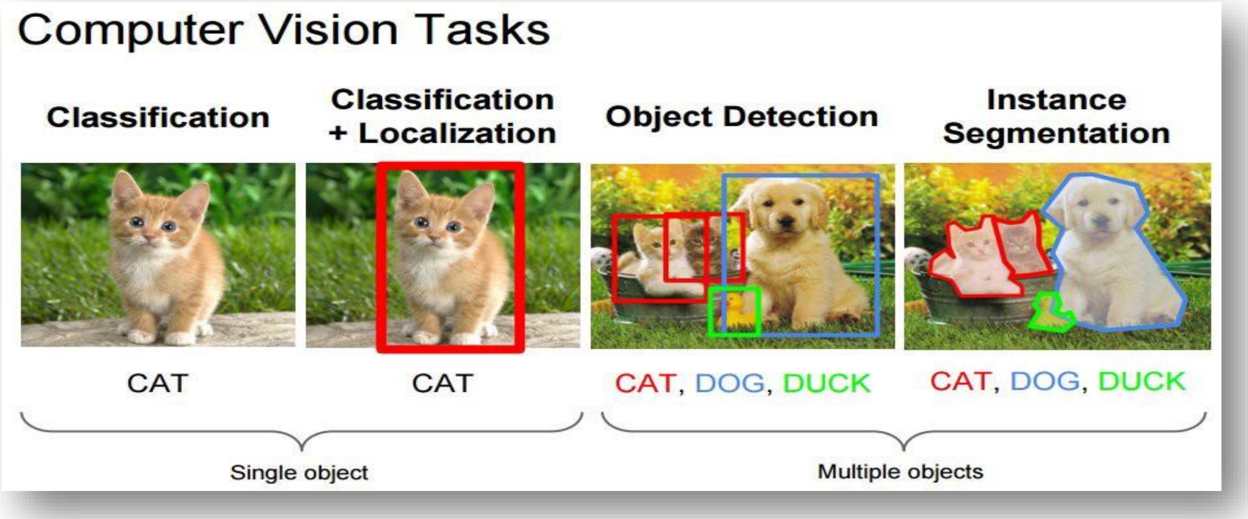


Figure 1.1.7: CV Tasks

Many popular computer vision applications involve trying to recognize things in photographs; for example:

- **Object Classification**: What broad category of object is in this photograph?
- **Object Identification**: Which type of a given object is in this photograph?
- **Object Verification**: Is the object in the photograph?
- **Object Detection**: Where are the objects in the photograph?
- **Object Landmark Detection**: What are the key points for the object in the photograph?
- **Object Segmentation**: What pixels belong to the object in the image?
- **Object Recognition**: What objects are in this photograph and where are they?

Outside of just recognition, other methods of analysis include:

- **Video motion analysis** uses computer vision to estimate the velocity of objects in a video, or the camera itself.
- In **image segmentation**, algorithms partition images into multiple sets of views.
- **Scene reconstruction** creates a 3D model of a scene inputted through images or video.
- In **image restoration**, noise such as blurring is removed from photos using Machine Learning based filters.

Any other application that involves understanding pixels through software can safely be labeled as computer vision.

METHODOLOGY

The system is to be composed of a number of sub-systems,which correspond to each stage of iris recognition. These stages are:

- **Image acquisition** - capturing eye image
- Quality testing(Image Pre-prosessing)[skipped in our case]
- **Segmentation** - locating the iris region in an eye image
- **Normalization** - creating a dimensionally consistent representation of the iris region
- **Feature encoding/Feature extraction** - creating a template containing only the most discriminating features of the iris.[As we are using ORB detection technique this need not to be done manually]
- **Matching** - This phase consists of two steps, namely matching and identification. In the matching process, the extracted features of the iris are compared with the iris images in the database.
- **Authentication**

1.2 ABOUT PROJECT

Computer vision is a significant research field, which provides efficient solutions to many problems. Pattern recognition is mainly used to automatically recognize different entities from an image. The security sector has given computer vision much attention, particularly for identification. Every human has unique, particular properties such as shape and size. Modern security sciences use these unique features to control access to restricted places, which, in the field of security, is a fundamental problem.

1.2.1 PROBLEM DEFINITION

The increasing demand for efficient authentication systems in the security field has sparked the development of authentication systems that are more secure and efficient. Traditional approaches to identification such as the use of a key or password are unsatisfactory in several application areasas these methods can easily be forgotten, stolen, or cracked. To overcome these weaknesses, modern science is interested in automating identification systems using biometric techniques.The need forreliable and secure systems has led to the emergence of the physiological and behavioral models in biometric systems. Both these models work effectively for security measures.

1.2.2 PROPOSED SOLUTION

Physiological biometrics include iris recognition, fingerprint recognition, face recognition, retina recognition, and hand geometry recognition. Biometric techniques in behavioural models comprise signature recognition, voice recognition, and gait recognition. Recently, iris recognition techniques have achieved great performance in identification. Among authentication techniques, iris recognition systems have received attention very much due to their rich iris texture which gives robust standardsfor identifying individuals. Among all biometric recognition systems, the iris recognition system (IRS) is the system with higher efficiency and is the more reliable system for checking authenticity.

This is due to the stability of the human iris, its invariance over time (i.e., it remains stable in spite of the aging process), and its uniqueness for every person, even between siblings or twins. It is covered via a structure that, if changed, could affect a person’s health and can be accessed using a non-invasive device. Thus, several leading companies, particularly in the sector of security, are looking forward to the future of the IRS due to different applications and potential of this technology.

Furthermore, the iris region is also deemed as one of the very stable biometric features which hardly change over time. Hence, the structure of the iris region can effectively be utilized for recognition or identification. With large-scale-range national identification programs being increasingly deployed all over the world, the demand for correct and reliable biometric recognition systems has increased. National identification programs are increasingly utilizing the IRS for better accuracy and reliability to register citizens (in addition to the utilization of different biometric recognition techniques such as two-dimensional face and fingerprint recognition system).The IRS is a high-accuracy verification technology and has a high ability for personal identification. It is increasingly being applied in automated systems (i.e., without human operator supervision)and is used especially in the security field. Thus, many countries use the IRS to improve security such as at the gates of smart airports and borders, in mobile devices, and at government buildings like hospitals.

In my project I have designed an IRIS BIOMETRIC SYSTEM, that uses computer vision technology to do facial recognition and to capture the IRIS of an individual, and store their information in a database for biometric authentication. The DLIB library is used to detect and recognize the face structure to pinpoint the eyes, and then capture the eyes through openCV's video capturing functions. The captured IRIS is stored into an image along with input from the user. During verification an associated ID inputted by the user is entered when verifying the IRIS. The IRIS is captured through the video feed, and sent to a pattern feature-like matching system setup using ORB's detector. The detector does pattern matching of both the user's IRIS through the feed, and the stored image of the user's IRIS from the keypoints and the authentication is done only if the match is found.

1.3 OBJECTIVE

The main aim of this Biometric system is to first Identify the face of the person and locate his Iris and capture it while Registration and store that Iris as a image file along with that registered users information into a Database. Now, when that person need to be authenticated that is, in verification process after a thorough pattern matching between both the Iris that is, the one which came through the video feed and the one already stored in the database, It verifies the user and only if the Iris and the ID inputted by the user matches the one that is in the Database it authenticates saying Match found, or else it does not authenticates.

CHAPTER 2

BIOMETRIC SYSTEMS SURVEY

2.1 THEORITICAL BACKGROUND

Biometrics is the branch of science that deals with the identification and verification of an individual based on the physiological and behavioral traits. These traits or identifiers are permanent, unique and can separate one individual from another. Biometric recognition systems integrate complex definitional, technological and operational selection under various contexts. The systems are not going to replace the authentication tools and technologies, but the combination of biometric approaches and authentication methods to help in improving the security aspects of the applications where user cooperation can be inferred.

Biometric based recognition methods and tools have become popular for the development of many useful, challenging and widely accepted applications such as security issues, surveillance, forensic investigations, fraudulent technologies, identity access management and access control. These systems also help to identify an individual in group of industrial networks, home/office building and control system. For the successful implementation of the biometric systems, deep artificial neural networks are in great demand. These systems can be built up either on the single modality or multiple modalities.

Basic Criteria for Biometrics Security Systems

There are seven basic criteria for biometric security system:
uniqueness, universality, permanence, collectability, performance, acceptability and circumvention. As mentioned above, **uniqueness** is considered as the priority one requirement for biometric data. It will indicate how differently and uniquely the biometric system will be able to recognize each user among groups of users. For instance, the DNA of each person is unique and it is impossible to replicate. **Universality** is the secondary criteria for the biometric security. This parameter indicates requirements for unique characteristics of each person in the world, which cannot be replicated. For example, retinal and iris are characteristics will satisfy this requirement.

Thirdly, a **permanence** parameter is required for every single characteristic or trait which is recorded in the database of the system and needs to be constant for a certain period of time period. This parameter will mostly be affected by the age of the user. Following the permanence parameter is the collectability. The **collectability** parameter requires the collection of each characteristic and trait by the system in order to verify their identification. Then, **performance** is the next parameter for the system which outlines how well the security system works. The accuracy and robustness are main factors for the biometric security system. These factors will decide the performance of the biometric security system. The **acceptability** parameter will choose fields in which biometric technologies are acceptable.

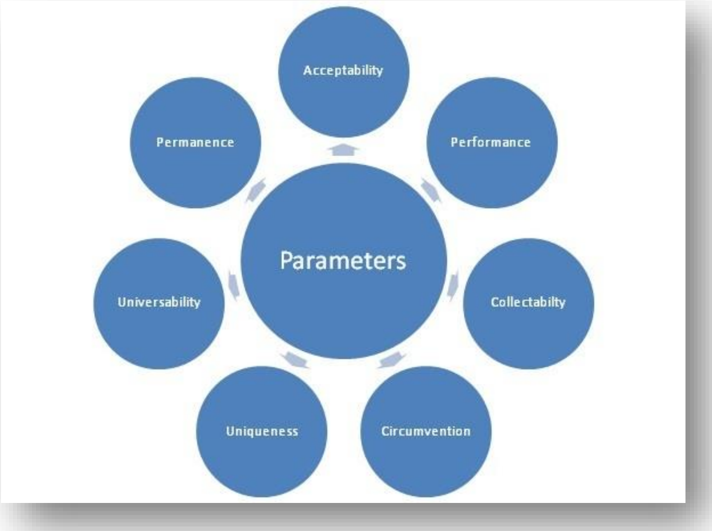


Figure 2.1.1: Basic Criteria of Biometric systems

Finally, **circumvention** will decide how easily each characteristic and trait provided by the user can lead to failure during the verification process. DNA is believed to be the most difficult characteristic leading to the failure of the verification process.

TYPES OF BIOMETRIC SYSTEMS

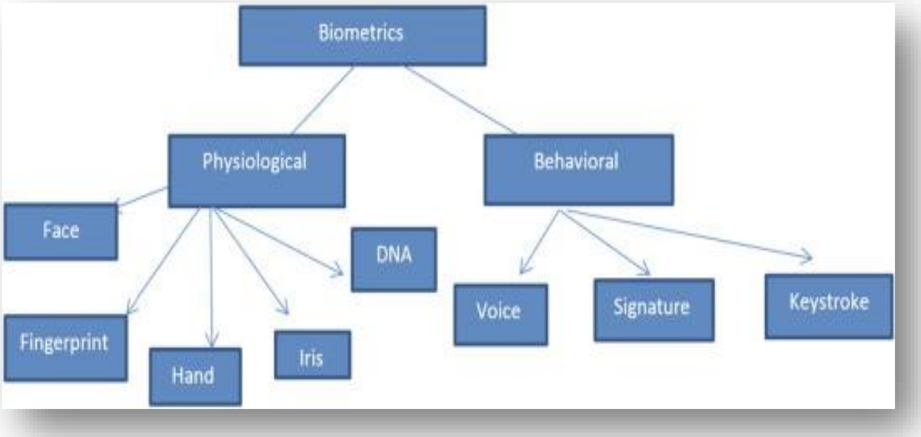


Figure 2.1.2: Types of Biometric systems

Verifying through Biometrics can generally be 2 types:

Physiological biometrics include iris recognition, fingerprint recognition, face recognition, retina recognition, and hand geometry and DNA recognition. Biometric techniques in behavioural models comprise signature recognition, voice recognition and Keystroks. All of these include their own advantages and disadvantages, where Iris being more reliable and accurate.

2.2 EXISTING SYSTEM

Many research works carried out since John Daugman developed and patented the first actual algorithms to perform iris recognition, published the first papers about it and gave the first live demonstrations, the concept behind this invention has a much longer history and today it benefits from many other active scientific contributors. He invented iris recognition, biometric algorithms for identifying persons reliably and rapidly using the random texture visible in the iris of an eye. Encoded iris pattern phase sequences yield much entropy, enabling recognition with confidence levels high enough to search national-sized databases without identity collisions, and at speeds of millions per second per CPU core.

2.3 PROPOSED SYSTEM

In future numerous business enterprises will be relied on biometrics technologies since it is the only way to provide guarantee about the presence of an individual. The proposed system provides iris recognition based biometric system in where the web-cam(Laptop camera) takes images of persons and sends to the computer for further processing and storing for future usages. The images are stored in data bases that will be used for comparing with the next images of person for authentication. If the next image gets matched with the stored one then the message goes to the administrative computer and if not matched, match not found will be displayed. Computer/Laptop can be used for data storing, processing and analyzing for iris recognition process.

Here, I have designed an IRIS BIOMETRIC SYSTEM, that uses computer vision technology to do facial recognition and to capture the IRIS of an individual, and store their information in a database for biometric authentication. The DLIB library is used to detect and recognize the face structure to pinpoint the eyes, and then capture the eyes through openCV's video capturing functions. The captured IRIS is stored into an image along with input from the user. During verification an associated ID inputted by the user is entered when verifying the IRIS. The IRIS is captured through the video feed, and sent to a pattern feature-like matching system setup using ORB's detector. The detector does pattern matching of both the user's IRIS through the feed, and the stored image of the user's IRIS from the keypoints and the authentication is done only if the match is found.

2.4 ADVANTAGES OF PROPOSED SYSTEM

Iris Biometric System has many advantages over other biometric systems present namely:

- It has a very Low False Acceptance Rate.
- It is very difficult to Replicate.
- As it takes both Iris image and ID as input for verifying it has an extra step of authentication.
- Contactless Identification is possible through Iris based Authentication.
- It is very Stable and Versatile.
- Irises are different for even identical twins

2.5 FEASIBILITY STUDY

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn't profitable.

2.5.1 OPERATIONAL FEASIBILITY

The number of people working on this project are 3 to 4. These persons should have knowledge on the technologies from the domain of Artificial Intelligence (A.I.), those are understanding of Machine Learning (M.L.) and its types. Working of Natural Language Processing (N.L.P), Computer Vision(CV) and how Iris Recognition and Authentication works.

2.5.2 TECHNICAL FEASIBILITY

This assessment is based on an outline design of system requirements, to determine whether the company has the technical expertise to handle completion of the project. When writing a feasibility report, the following should be taken to consideration:

- A brief description of the business to assess more possible factors which could affect the study
- The part of the business being examined
- The human and economic factor
- The possible solutions to the problem

At this level, the concern is whether the proposal is both *technically* and *legally* feasible (assuming moderate cost).

The technical feasibility assessment is focused on gaining an understanding of the present technical resources of the organization and their applicability to the expected needs of the proposed system. It is an evaluation of the hardware and software and how it meets the need of the proposed system.

2.5.2.1 Survey of Technology

For our project we have chosen the Computer Vision aka C.V. Technology as we found that by using this technology, we can complete our project and get out desired result.

2.5.2.2 Feasibility of Technology

For our project from Computer Vision aka C.V., we have chosen necessary functions to capture the photo of iris through video feed while registering and verification.

2.5.3 ECONOMIC FEASIBILITY

Our project is economically supportive as it's required small or medium amount of resources which will cost up-to medium amount for those resources.

CHAPTER 3

SYSTEM ANALYSIS

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

3.1 SPECIFICATIONS

3.1.1 Functional requirements

The following are the functional requirements of our project:

- When web cam opens up a Human face is required to locate and capture Iris, So first functional requirement could be an Iris that is captured from video feed for both registration and verification(authentication).

3.1.2 Non Functional Requirements

- **Maintainability:** Maintainability is used to make future maintenance easier, meet new requirements.
- **Robustness:** Robustness is the quality of being able to withstand stress, pressures or changes in procedure or circumstance.
- **Reliability:** Reliability is an ability of a person or system to perform and maintain its functions in circumstances.
- **Size:** The size of a particular application play a major role, if the size is less then efficiency will be high.
- **Speed:** If the speed is high then it is good. When the number of lines in code are less then the speed of the program will be comparatively less but not always.

3.2 SOFTWARE REQUIREMENTS

One of the most difficult tasks is that, the selection of the software, once system requirement is known that is determining whether a particular software package fits the requirements.

PROGRAMMING LANGUAGE	**PYTHON**
VERSION	**3.6.4/above**
TECHNOLOGY	**COMPUTER VISION**
TOOL	**ANACONDA**
IDE	**JUPYTER NOTEBOOK**
OPERATING SYSTEM	**WINDOWS 10**
BROWSER	**GOOGLECHROME(version 88.0.4324.150)**

Table3.2.1SoftwareRequirements

3.3 HARDWARE REQUIREMENTS

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and the capacity requirements are also important.

PROCESSOR	**INTEL CORE I3 OR ABOVE**
RAM CAPACITY	**4GB AND ABOVE**
HARDDISK	**500 GB AND ABOVE**
I/O	**WEB CAMERA, KEYBOARD, MONITER, MOUSE**

Table 3.3.1 Hardware Requirements

3.4 MODULE DESCRIPTION

The DLIB library is used to detect and recognize the face structure to pinpoint the eyes, and then capture the eyes through openCV's video capturing functions.
The IRIS is captured through the video feed, and sent to a pattern feature-like matching system setup using ORB's detector. The detector does pattern matching of both the user's IRIS throughthe feed, and the stored image of the user's IRIS from the keypoints.

The following Python Libraries need to be installed for implementation of this project :

- OpenCV
- Numpy
- Pillow
- PyQT5
- CMAKE (this is required to install dlib)
- DLIB
- DateTime
- Tkinter
- (pip install opencv-contrib-python) - REMOVED, the library is patented so its no longer used for Iris Matching but opencv-contrib-python has other useful libraries so its worth installing.

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these feature. So to install it execute the following command in the command-line:

```
pip install opencv-python
```

Numpy

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. Numpy is the core library for scientific computing in Python. It provides a high- performance multidimensional array object, and tools for working with these arrays.It is used for Numerical Calculations
So to install it execute the following command in the command-line:

```
pip install numpy
```

In iris recognition numpy is used to calculate the distance between pupil and iris for accurate identification of iris.

Pillow

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of image file formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders. This module is not preloaded with Python. So to install it execute the following command in the command-line:

```
pip install pillow
```

PyQt5

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for **Qt**, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library. So to install it execute the following command in the command-line:

```
pip install pyqt5
```

Qt is used for developing graphical user interfaces (GUIs) and multi-platform applications that run on all major desktop platforms and most mobile or embedded platforms. Most GUI programs created with Qt have a native-looking interface, in which case Qt is classified as a *widget toolkit*. Non-GUI programs can also be developed, such as command-line tools and consoles for servers. An example of such a non-GUI program using Qt is the Cutelyst web framework.

DLIB(Digital Library)

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge. So to install it execute the following command in the command-line:

```
pip install dlib
```

But for installing dlib we first need to install **cmake** using command :

```
pip install cmake
```

This is because dlib was developed in C based programming language, so it needs this program to use it. CMake is an open-source, cross-platform family of tools designed to build, test and package software. CMake is used to control the software compilation process using simple platform and compiler independent configuration files, and generate native makefiles and workspaces that can be used in the compiler environment of your choice. The suite of CMake tools were created by Kitware in response to the need for a powerful, cross-platform build environment for open-source projects such as ITK and VTK.In our project the DLIB library is used to detect and recognize the face structure to pinpoint the eyes.

What is Facial Landmark?

Detecting facial landmarks is a *subset* of the *shape prediction* problem. Given an input image (and normally an ROI that specifies the object of interest), a shape predictor attempts to localize key points of interest along the shape.

In the context of facial landmarks, our goal is detect important facial structures on the face using shape prediction methods.

Detecting facial landmarks is therefore a two step process:

- Step #1: Localize the face in the image.
- Step #2: Detect the key facial structures on the face ROI(here in our case ROI is Eyes(Iris))

Understanding dlib's facial landmark detector

The pre-trained facial landmark detector inside the dlib library is used to estimate the location of ***68 (x, y)-coordinates*** that map to facial structures on the face.The indexes of the 68 coordinates can be visualized on the image below:

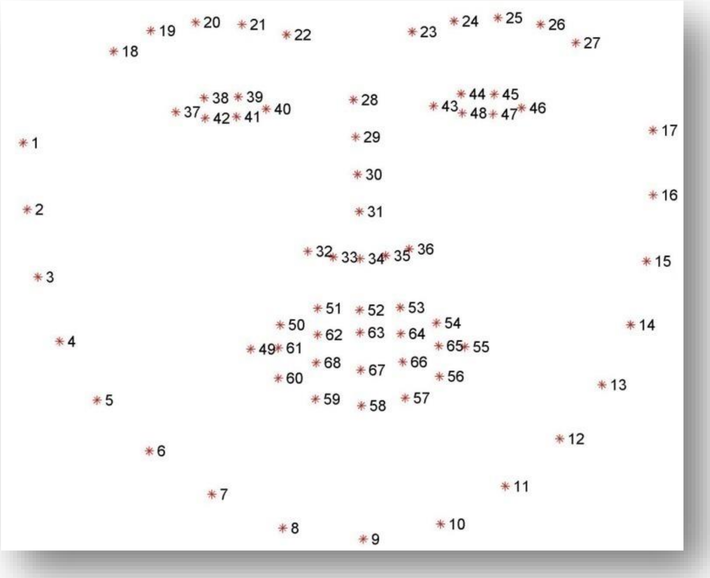


Figure 3.4.1: Facial Landmarks using dlib

These annotations are part of the 68 point **iBUG 300-W dataset** which the dlib facial landmark predictor was trained on.

It's important to note that other flavors of facial landmark detectors exist, including the 194 point model that can be trained on the HELEN dataset.

Regardless of which dataset is used, the same dlib framework can be leveraged to train a shape predictor on the input training data — this is useful if we would like to train facial landmark detectors or custom shape predictors of our own.

DateTime

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

ORB(Oriented FAST and Rotated BRIEF)Detector

ORB is a fusion of FAST keypoint detector and BRIEF descriptor with some added features to improve the performance. FAST is **Features from Accelerated Segment Test** used to detect features from the provided image. It also uses a pyramid to produce multiscale-features. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch. Now it doesn't compute the orientation and descriptors for the features, so this is where BRIEF comes in the role.

ORB uses BRIEF descriptors but as the BRIEF performs poorly with rotation. So what ORB does is to "steer" BRIEF according to the orientation of keypoints. For any feature set of n binary tests at location (xi,yi), define a 2×n matrix, S which contains the coordinates of these pixels. Then using the orientation of patch, θ, its rotation matrix is found and rotates the S to get steered(rotated) version Sθ.

ORB discretize the angle to increments of 2π/30 (12 degrees), and construct a lookup table of precomputed BRIEF patterns. As long as the keypoint orientation θ is consistent across views, the correct set of points Sθ will be used to compute its descriptor.

BRIEF has an important property that each bit feature has a large variance and a mean near 0.5. But once it is oriented along keypoint direction, it loses this property and become more distributed. High variance makes a feature more discriminative, since it responds differentially to inputs. Another desirable property is to have the tests uncorrelated, since then each test will contribute to the result. To resolve all these, ORB runs a greedy search among all possible binary tests to find the ones that have both high variance and means close to 0.5, as well as being uncorrelated. The result is called **rBRIEF**.

For descriptor matching, multi-probe LSH which improves on the traditional LSH, is used. The paper says ORB is much faster than SURF and SIFT and ORB descriptor works better than SURF. ORB is a good choice in low-power devices for panorama stitching etc.

ORB in OpenCV

As usual, we have to create an ORB object with the function, **cv.ORB()** or using feature2d common interface. It has a number of optional parameters. Most useful ones are nFeatures which denotes maximum number of features to be retained (by default 500), scoreType which denotes whether Harris score or FAST score to rank the features (by default, Harris score)etc. It find the features in an image and match them with the other images in a continuous Video.

Algorithm

- Take the query image and convert it to grayscale.
- Now Initialize the ORB detector and detect the keypoints in query image and scene.

- Compute the descriptors belonging to both the images.
- Match the keypoints using Brute Force Matcher.
- Show the matched images.

TKINTER

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

CHAPTER 4

SOFTWARE DESIGN

4.1 INTRODUCTION

Software design is a process to transform user requirements into some suitable form, which helps the programmer in software coding and implementation.

For assessing user requirements, an SRS (Software Requirement Specification) document is created whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages.

Software design is the first step in SDLC (Software Design Life Cycle), which moves the concentration from problem domain to solution domain. It tries to specify how to fulfill the requirements mentioned in SRS.

Design Verification

The output of software design process is design documentation, pseudo codes, detailed logic diagrams, process diagrams, and detailed description of all functional or non-functional requirements.

The next phase, which is the implementation of software, depends on all outputs in each phase of software design. It is then becomes necessary to verify the output before proceeding to the next phase. The early any mistake is detected, the better it is or it might not be detected until testing of the product. If the outputs of design phase are in formal notation form, then their associated tools for verification should be used otherwise a thorough design review can be used for verification and validation.

By structured verification approach, reviewers can detect defects that might be caused by overlooking some conditions. A good design review is important for good software design, accuracy and quality.

4.2 BLOCK DIAGRAM

The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation. It is a visual representation of a system that uses simple, labeled blocks that represent single or multiple items, entities or concepts, connected by lines to show relationships between them.

Block diagrams are a generalized representation of a concept and are not intended to display complete information in regards to design or manufacture. Unlike schematics, blueprints and layout diagrams, block diagrams do not portray the necessary detail for physical construction. Block diagrams are made simple so as not to cloud concepts.

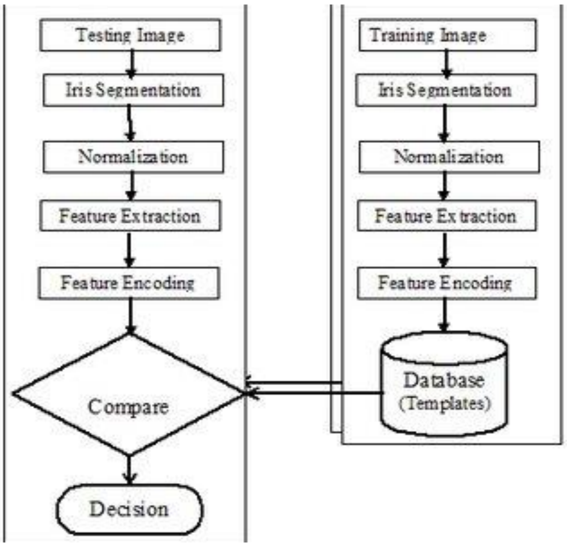


Figure 4.2.1: Block Diagram for Iris based Recognition and Authentication biometric system

4.3 DETAILED LEVEL DIAGRAMS(DFD)

Data flow diagram (DFD) is a graphical representation of "flow" of data through an information system, modelling its process concepts. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFD's can also be used for the visualization of data processing (structured design).

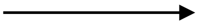
A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It doesn't show information about timing of processes, or information about whether processes will operate in sequence or parallel. A DFD is also called as "bubble chart".

4.3.1 DFD Symbols

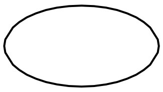
In the DFD, there are four symbols:

- A square define a source or destination of system data.
- An arrow indicates dataflow. It is the pipeline through which the information flows.
- A circle or a bubble represents transforms dataflow into outgoing dataflow.
- An open rectangle is a store, data at reset or at temporary repository of data.

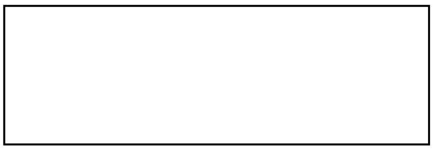
Dataflow: Data move in a specific direction from an origin to a destination.



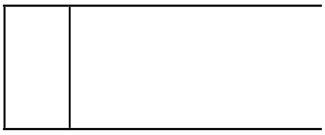
Process: People, procedures or devices that use or produce (Transform) data. The physical component is not identified.



Sources: External sources or destination of data, which may be programs, organizations or other entity.



Data store: Here data is stored or referenced by a process in the system.



In our project, we had built the data flow diagrams at the very beginning of business process modelling in order to model the functions that our project has to carry out and the interaction between those functions together with focusing on data exchanges between processes.In Software engineering DFD(data flow diagram) can be drawn to represent the system of different levels of abstraction. Higher-level DFDs are partitioned into low levels-hacking more information and functional elements. Levels in DFD are numbered 0, 1, 2 or beyond.

4.3.2 Level 0/Context Level DFD

It is also known as a context diagram. It’s designed to be an abstraction view, showing the system as a single process with its relationship to external entities. It represents the entire system as a single bubble with input and output data indicated by incoming/outgoing arrows. A Context level Data flow diagram created using select structured systems analysis and design method (SSADM). This level shows the overall context of the system and its operating environment and shows the whole system as just one process. It does not usually show data stores, unless they are “owned” by external systems, e.g. are accessed by but not maintained by this system, however, these are often shown as external entities. The Context level DFD is shown in fig.4.3.2.1

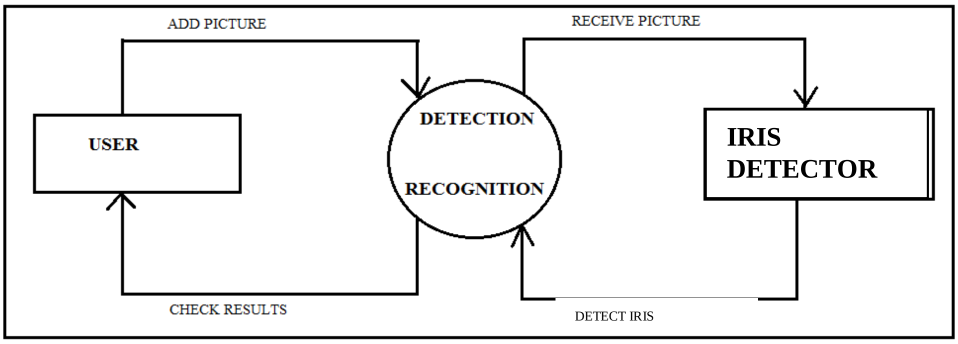


Figure 4.3.2.1: Level-0 DFD for Iris based Recognition and Authentication biometric system

The Context Level Data Flow Diagram shows the data flow from the application to the database and to the system.

4.3.3 Level-1/Top Level DFD

In 1-level DFD, the context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main functions of the system and breakdown the high-level process of 0-level DFD into subprocesses**.**

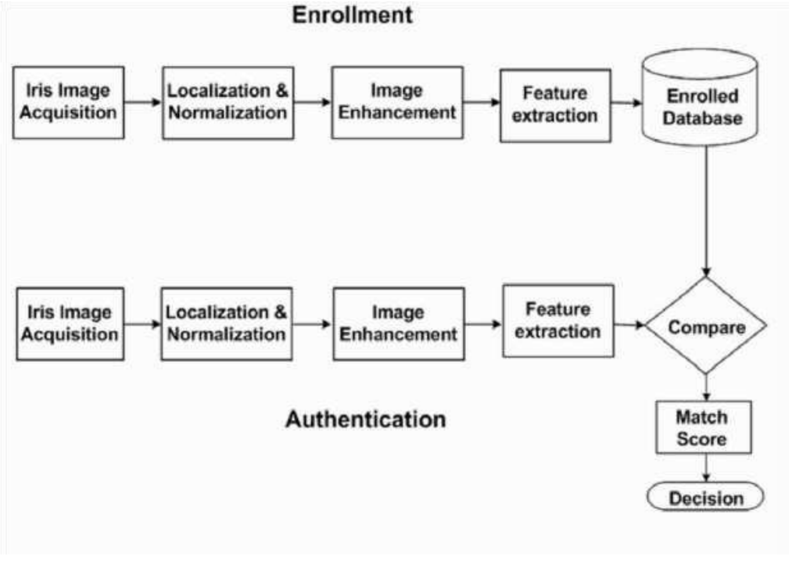


Figure 4.3.3.1: Level-1 DFD for Iris based Recognition and Authentication biometric system

4.3.4 Level-2/Detailed Level DFD

2-level DFD goes one step deeper into parts of 1-level DFD. It can be used to plan or record the specific/necessary detail about the system's functioning.

This level explains each process of the system in a detailed manner. In first detailed level DFD (Generation of individual fields): how data flows through individual process/fields. In second detailed level DFD (generation of detailed process of the individual fields) Detailed description of individual levels are given here.

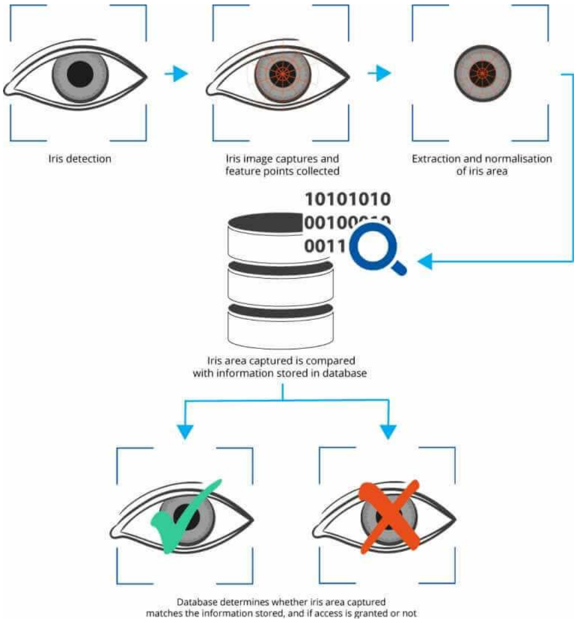

Iris detection
Iris image captures and feature points collected
Extraction and normalisation of iris area
10101010
Iris area captured is compared with information stored in database
Database determines whether iris area captured matches the information stored, and if access is granted or not

Figure 4.3.4.1: Level-2 DFD for Iris based Recognition and Authentication biometric system

4.4 UNIFIED MODELLING LANGUAGE DIAGRAMS

The Unified Modelling Language (UML) is a Standard language for specifying, visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagrams.

- **User Model View**

This view represents the system from the user's perspective. The analysis representation describes a usage scenario from the end- users perspective.

- **Structural Model View**

In this model the data and functionality are arrived from inside the system.This model view models the static structures.

- **Behavioral Model View**

It represents the dynamic of behavioral as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

- **Implementation model View**

In this the structural and behavioral as parts of the system are represented as they are to be built.

- **Environmental Model View**

In this the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

The different types of UML diagrams include:

1. Class diagram
2. Object diagram
3. Use case diagram
4. Interaction diagram
5. Activity diagram

4.4.1 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

4.4.2 Object Diagram

Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams. Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams. Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment. Object diagrams are used to render a set of objects and their relationships as an instance.

4.4.3 Use Case Diagram

Use case diagrams are one of the five diagrams in the UML for modeling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modeling the dynamic aspects of systems).Use case diagrams are central to modeling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.

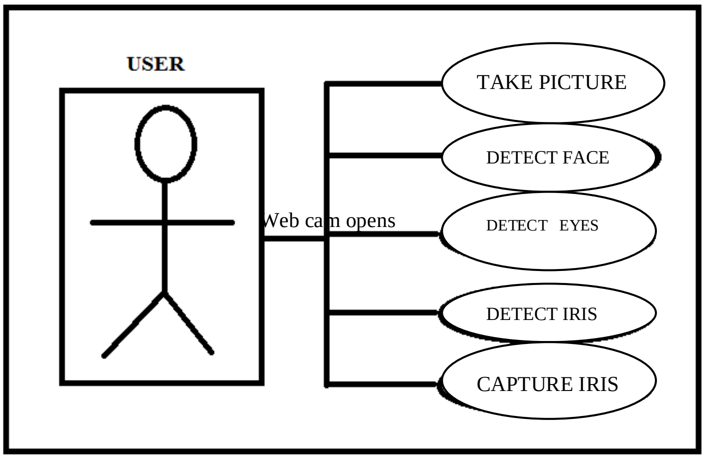


Figure 4.4.3.1: Use Case Diagram Iris based Recognition and Authentication biometric system

4.4.4 Interaction Diagrams

From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.

This interactive behavior is represented in UML by two diagrams known **as Sequence diagram and Collaboration diagram**. The basic purpose of both the diagrams are similar.

Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

4.4.4.1 Sequence Diagram

Sequence diagram is the diagram in which main representation is of the sequence of messages flowing from one object to another; also main emphasis is on representing that how the messages/events are exchanged between objects and in what time order. Sequence diagrams are a popular dynamic modelling solution. Dynamic modeling focuses on the interactions occurring within the system. Sequence diagrams specifically focus on the "lifelines" of an object and how they communicate with other objects to perform a function before the lifeline ends.

4.4.4.2 Collaborative Diagram

The collaboration diagram is used to show the relationship between the objects in a system.
Both the sequence and the collaboration diagrams represent the same information but differently.
Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features.
Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

4.4.5 Activity Diagram

In UML, the activity diagram is used to demonstrate the flow of control within the system rather than the implementation. It models the concurrent and sequential activities.

The activity diagram helps in envisioning the workflow from one activity to another. It put emphasis on the condition of flow and the order in which it occurs. The flow can be sequential, branched, or concurrent, and to deal with such kinds of flows, the activity diagram has come up with a fork, join, etc. It is also termed as an object-oriented flowchart. It encompasses activities composed of a set of actions or operations that are applied to model the behavioral diagram.

4.5 CONTROL FLOW DIAGRAM

A control flow diagram helps us understand the detail of a process. It shows us where control starts and ends and where it may branch off in another direction, given certain situations. Let's say you are working on software to start a machine. What happens if the engine is flooded, or a spark plug is broken? Control then changes the flow to other parts of the software.

We can represent these branches with a diagram. The flow diagram is helpful because it can be understood by both stakeholders and systems professionals. Although some of the symbols might not be fully understood by the layperson, they can still grasp the general concept.

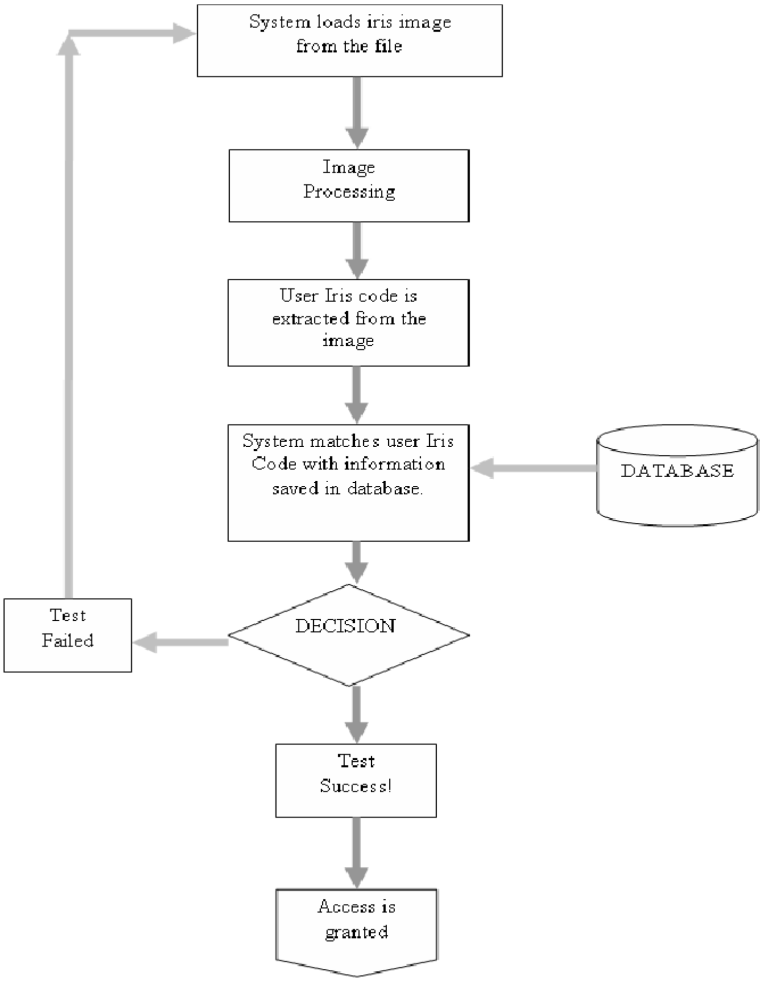
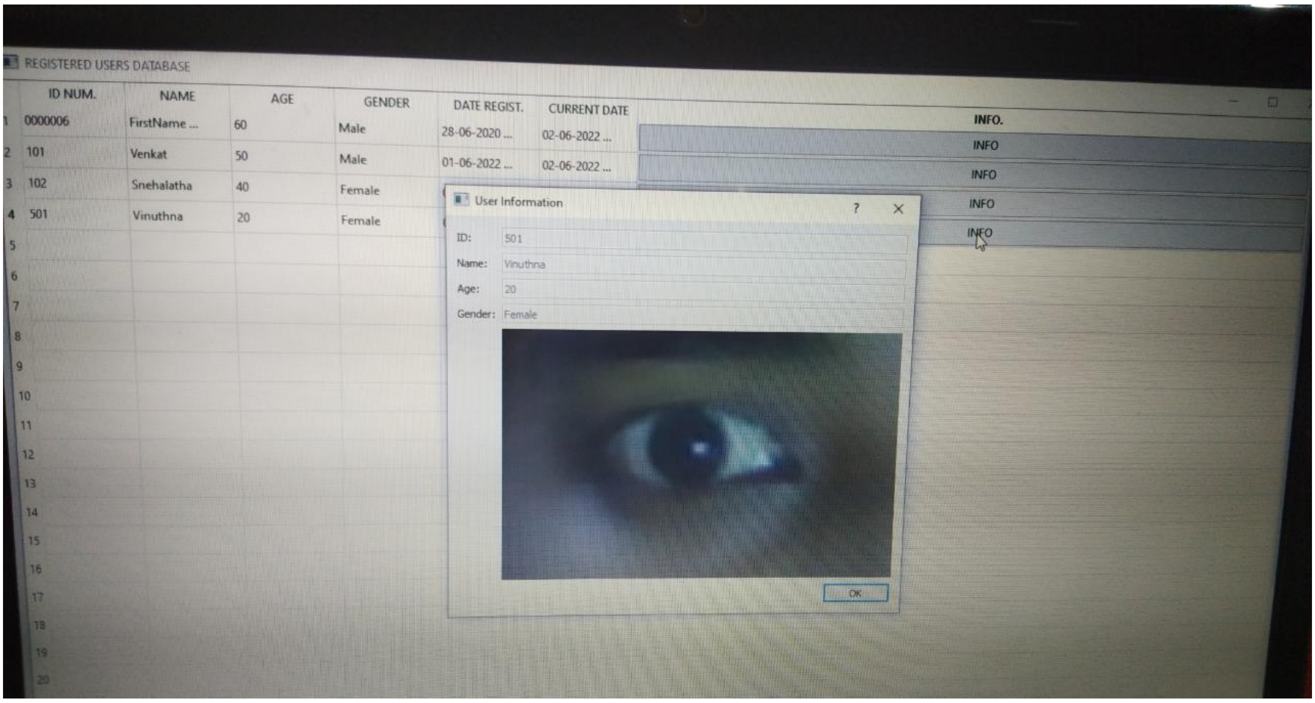

System loads iris image from the file
Image Processing
User Iris code is extracted from the image
System matches user Iris Code with information saved in database.
DATABASE
DECISION
Test Failed
Test Success!
Access is granted

Figure 4.5.1: Control Flow Diagram for Iris based Recognition and Authentication biometric system

4.6 DATABASE DESIGN

In order to store the registered users information we along with the captured Iris in a image format we have designed a database.



REGISTERED USERS DATABASE

	ID NUM.	NAME	AGE	GENDER	DATE REGIST.	CURRENT DATE	INFO.
1	0000006	FirstName ...	60	Male	28-06-2020 ...	02-06-2022 ...	INFO
2	101	Venkat	50	Male	01-06-2022 ...	02-06-2022 ...	INFO
3	102	Snehalatha	40	Female			INFO
4	501	Vinuthna	20	Female			INFO

User Information

ID: 501

Name: Vinuthna

Age: 20

Gender: Female

Figure 4.6.1: Registered users Database Design

4.6.1 Entity Relationship(ER)Diagram

ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system. It develops a conceptual design for the database. It also develops a very simple and easy to design view of data. In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

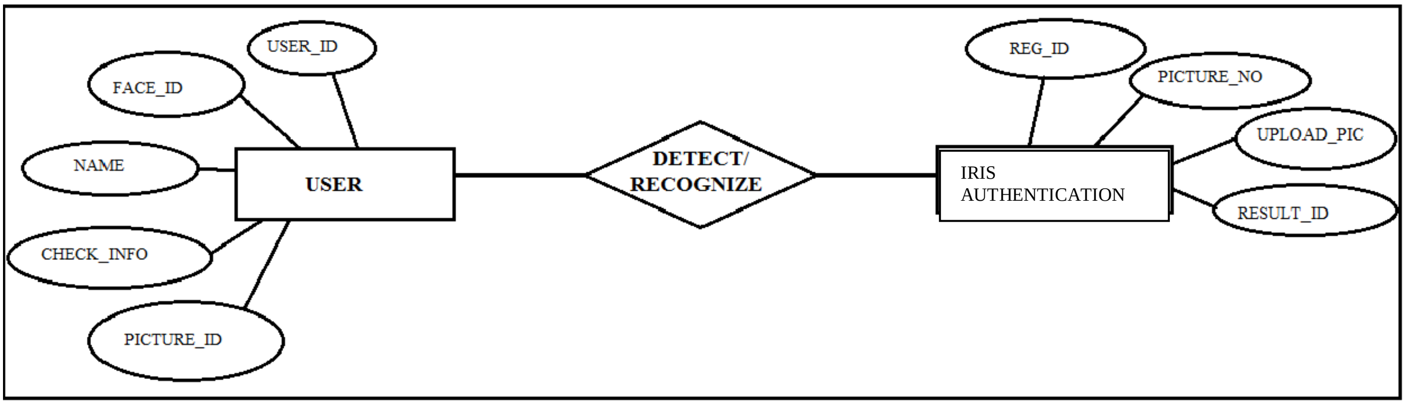


Figure 4.6.1: ER Diagram for Iris based Recognition and Authentication biometric system

CHAPTER 5

IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work andbe effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of change over methods.

The project is implemented by accessing simultaneously from more than one system and more than one window in one system. The application is implemented in the Internet Information Services 5.0 web server under the Windows XP and accessed from variousclients.

5.1 TECHNOLOGIES USED

For easy implementation of Iris based Biometric System using Computer Vision Technology we have chosen Python as our Programming Language.

What is Python?

Python is an interpreter, high-level programming language for general-purpose programming by “Guido van Rossum” and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional, procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non-profit Python Software Foundation. Python is a general purpose, dynamic, high level and interpreted programming language. It supports object-oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high level data structures.

Python Versions

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting, garbage collector, and support for Unicode. With this release, the development process became more transparent and community-backed.

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward-compatible with previous versions. However, many of its major features have been back ported to the Python 2.6.xand 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3.In January 2017, Google announced work on a Python 2.7 to go Trans compiler to improve performance under concurrent workloads.

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and Python 3.7.0b1 (PEP 540) adds a new "UTF-8 Mode" (and overrides POSIX locale). For implementation of our Project we atleast need a python version of 3.6.4

Why Python?

- Python is a scripting language like PHP, Perl, and Ruby.
- No licensing, distribution, or development fees. It is an Open-Source Language.
- It is a Desktop application.
- Linux, windows
- Excellent documentation
- Thriving developer community

Libraries Of python:

Python's large standard library, commonly cited as one of its greatest strengths,provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 33), but most modules are not.

They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for thirdparty Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping
- Easily available Documentation
- System Administration

Modules in python

Module

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

1. **NumPy**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.
Using NumPy, mathematical and logical operations on arrays can be performed. In iris recognition numpy is used to calculate the distance between pupil and iris for accurate identification of iris, and also for the purpose of Face Tracking and Iris Tracking.

Operations using NumPy

Using NumPy, a developer can perform the following operations :

- Mathematical and logical operations on arrays.
- Fourier transforms and routines for shape manipulation.
- Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MATLAB

NumPy is often used along with packages like SciPy (Scientific Python) and Mat−plotid (plotting library). This combination is widely used as a replacement for MATLAB, a popular platform for technical computing. However, Python alternative to MATLAB is now seen as a more modern and complete programming language. It is open source, which is an added advantage of NumPy.

2. CV2

cv2 is the module import name for opencv-python, which has many different functions for video capturing and taking inputs from an image or live video.

3. OS

The OS module in Python provides functions for interacting with the operating system. OS comes under Python’s standard utility modules. This module provides a portable way of using operating system-dependent functionality. The *os* and *os.path* modules include many functions to interact with the file system**.**

4. PIL

Python Imaging Library (expansion of PIL) is the de facto image processing package for Python language. It incorporates lightweight image processing tools that aids in editing, creating and saving images. Support for Python Imaging Library got discontinued in 2011, but a project named pillow forked the original PIL project and added Python3.x support to it. Pillow was announced as a replacement for PIL for future usage. Pillow supports a large number of imagefile formats including BMP, PNG, JPEG, and TIFF. The library encourages adding support for newer formats in the library by creating new file decoders.This module is not preloaded with Pythonand requires installation.

5. PyQt5

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for **Qt**, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library.

6. DLIB(Digital Library)

Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems. It is used in both industry and academia in a wide range of domains including robotics, embedded devices, mobile phones, and large high performance computing environments. Dlib's open source licensing allows you to use it in any application, free of charge.

7. Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

8.Functools(Partial)

The functools module is for higher-order functions: functions that act on or return other functions. In general, any callable object can be treated as a function for the purposes of this module. Partial functions allows us to derive a function with x parameters to a function with fewer parameters and constant values set for the more limited function.We can write partial functional application in python through functools library.Partial is a higher order function which takes a function as input (like map and filter) but it also returns a function that can be used in the same way as any other function in your program. Partial function application is a very useful tools especially where you need to apply a range of different input to a single object or need to bind one of the arguments to a function to be constant.

COMPUTER VISION

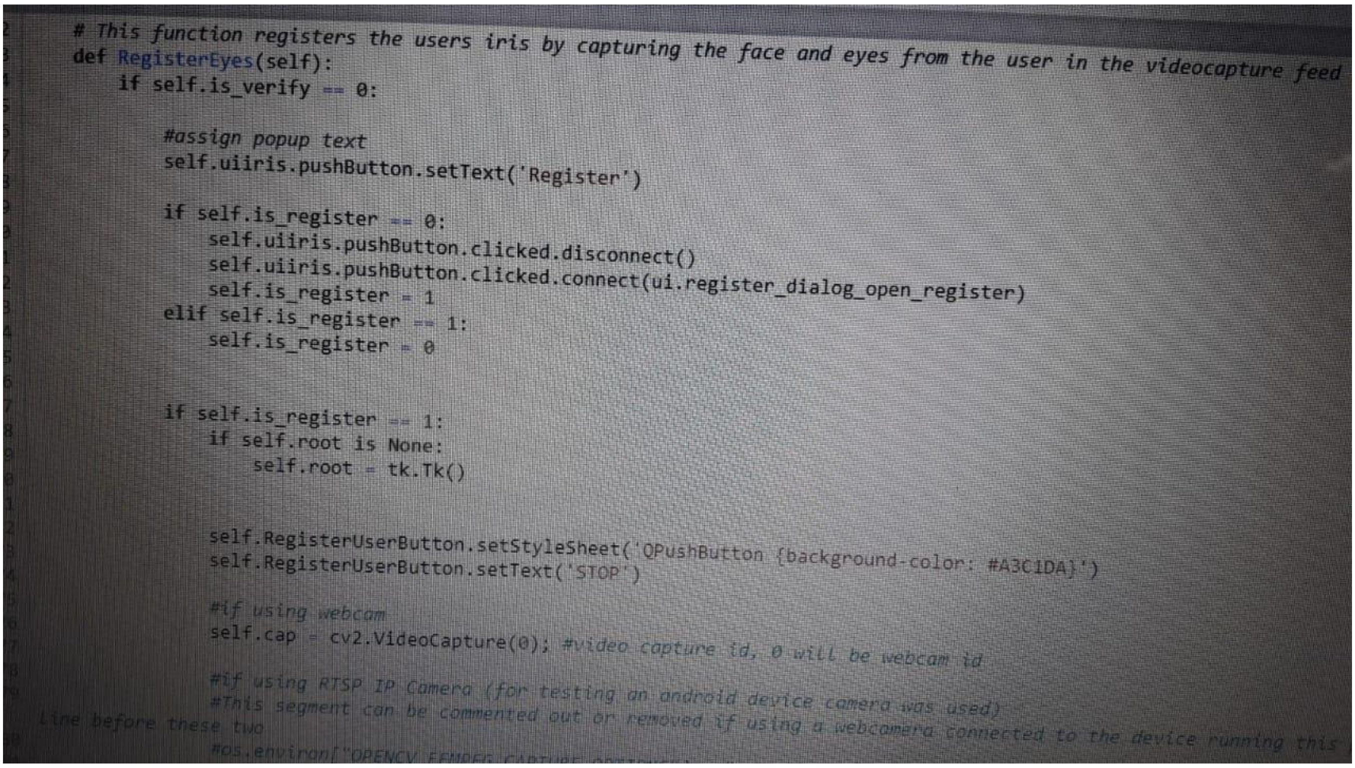
Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

We used Computer Vision Technology to implement this project as it is convenient with CV ti take a live image of iris of the person for registration and verification purposes.

5.2 CODE TEMPLATES

In our Project we have included four main features those are: Register, Verify, Show Reistered Users(this is the database where all registered users information is stored)and lastly to exit the system Close option. The main code templates of these functionalities are:

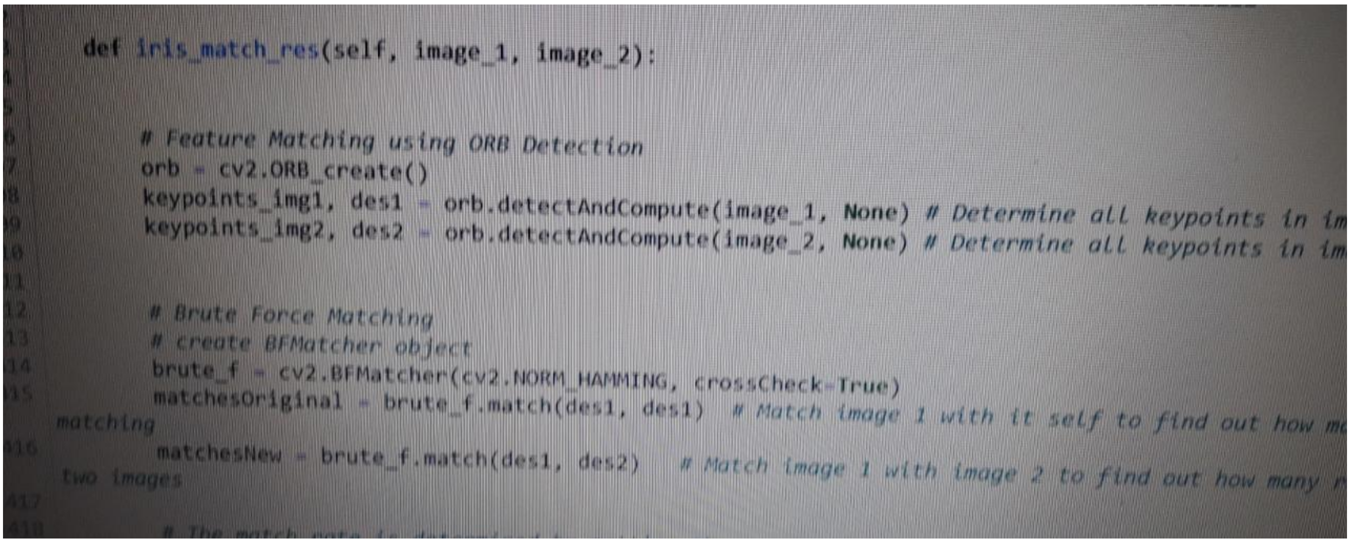
- Code Template to Register User trough videocapture:



```
# This function registers the users iris by capturing the face and eyes from the user in the videocapture feed
def RegisterEyes(self):
    if self.is_verify == 0:

        #assign popup text
        self.uiiris.pushButton.setText('Register')

        if self.is_register == 0:
            self.uiiris.pushButton.clicked.disconnect()
            self.uiiris.pushButton.clicked.connect(ui.register_dialog_open_register)
            self.is_register = 1
        elif self.is_register == 1:
            self.is_register = 0


        if self.is_register == 1:
            if self.root is None:
                self.root = tk.Tk()


            self.RegisterUserButton.setStyleSheet('QPushButton {background-color: #A3C1DA}')
            self.RegisterUserButton.setText('STOP')

            #if using webcam
            self.cap = cv2.VideoCapture(0); #video capture id, 0 will be webcam id

            #if using RTSP IP Camera (for testing an android device camera was used)
            #This segment can be commented out or removed if using a webcamera connected to the device running this
Line before these two
            #os.environ["OPENCV_FFMPEG_CAPTURE_OPTIONS"
```

Figure 5.2.1: Register Iris Template

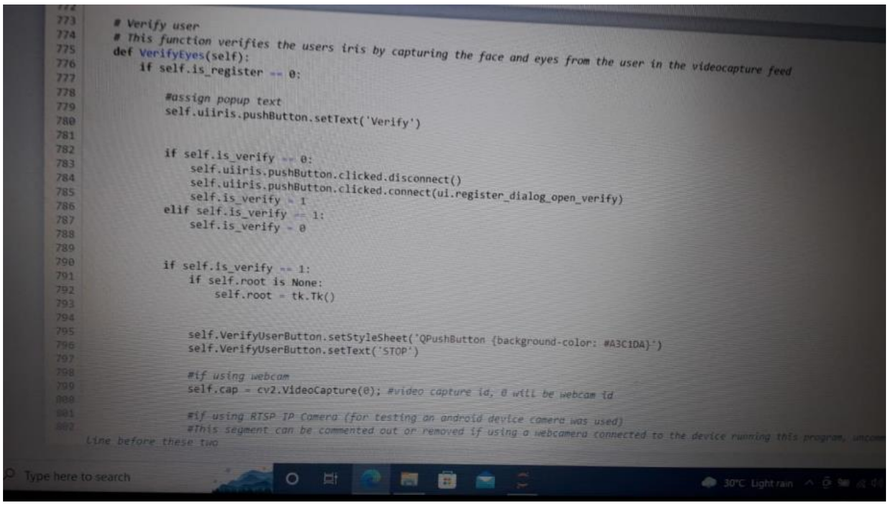
- Code Template for Iris matching using ORB detection



```
    def iris_match_res(self, image_1, image_2):


        # Feature Matching using ORB Detection
        orb = cv2.ORB_create()
        keypoints_img1, des1 = orb.detectAndCompute(image_1, None) # Determine all keypoints in im
        keypoints_img2, des2 = orb.detectAndCompute(image_2, None) # Determine all keypoints in im


        # Brute Force Matching
        # create BFMatcher object
        brute_f = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
        matchesOriginal = brute_f.match(des1, des1)  # Match image 1 with it self to find out how m
matching
        matchesNew = brute_f.match(des1, des2)   # Match image 1 with image 2 to find out how many m
two images


        # The match
```

Figure 5.2.2: Iris Matching using ORB detection

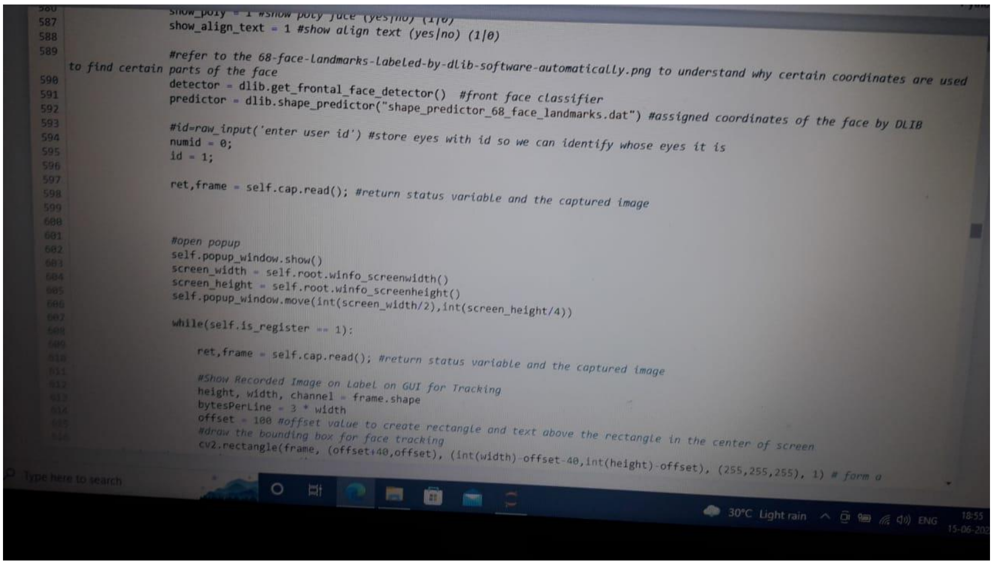
- Code Template to verify Eyes



```
    # Verify user
    # This function verifies the users iris by capturing the face and eyes from the user in the videocapture feed
    def VerifyEyes(self):
        if self.is_register == 0:

            #assign popup text
            self.uiiris.pushButton.setText('Verify')


            if self.is_verify == 0:
                self.uiiris.pushButton.clicked.disconnect()
                self.uiiris.pushButton.clicked.connect(ui.register_dialog_open_verify)
                self.is_verify = 1
            elif self.is_verify == 1:
                self.is_verify = 0


            if self.is_verify == 1:
                if self.root is None:
                    self.root = tk.Tk()


                self.VerifyUserButton.setStyleSheet('QPushButton {background-color: #A3C1DA}')
                self.VerifyUserButton.setText('STOP')

                #if using webcam
                self.cap = cv2.VideoCapture(0); #video capture id, 0 will be webcam id

                #if using RTSP IP Camera (for testing on android device camera was used)
                #This segment can be commented out or removed if using a webcamera connected to the device running this program, uncom
Line before these two
```

Figure 5.2.3: Verify User Template

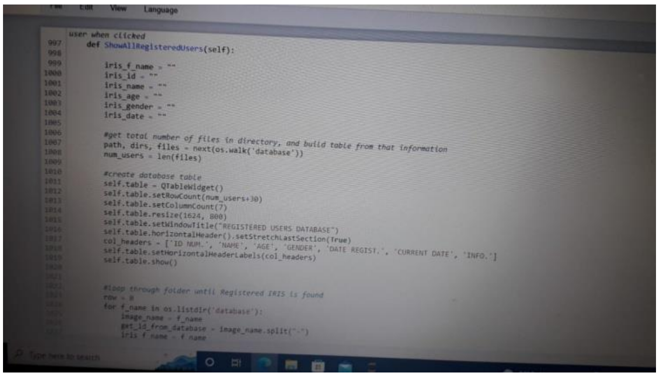
- Code Template to use DLIB in detecting Face



```
            show_poly = 1 #show poly face (yes|no) (1|0)
            show_align_text = 1 #show align text (yes|no) (1|0)

            #refer to the 68-face-landmarks-labeled-by-dlib-software-automatically.png to understand why certain coordinates are used
to find certain parts of the face
            detector = dlib.get_frontal_face_detector()  #front face classifier
            predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.dat") #assigned coordinates of the face by DLIB

            #id=raw_input('enter user id') #store eyes with id so we can identify whose eyes it is
            numid = 0;
            id = 1;

            ret,frame = self.cap.read(); #return status variable and the captured image



            #open popup
            self.popup_window.show()
            screen_width = self.root.winfo_screenwidth()
            screen_height = self.root.winfo_screenheight()
            self.popup_window.move(int(screen_width/2),int(screen_height/4))

            while(self.is_register == 1):

                ret,frame = self.cap.read(); #return status variable and the captured image

                #Show Recorded Image on Label on GUI for Tracking
                height, width, channel = frame.shape
                bytesPerLine = 3 * width
                offset = 100 #offset value to create rectangle and text above the rectangle in the center of screen
                #draw the bounding box for face tracking
                cv2.rectangle(frame, (offset+40,offset), (int(width)-offset-40,int(height)-offset), (255,255,255), 1) # form a
```

Figure 5.2.4: DLIB face detection Template

- Code Template for creating database for registered users



```
user when clicked
    def ShowAllRegisteredUsers(self):

        iris_f_name = ""
        iris_id = ""
        iris_name = ""
        iris_age = ""
        iris_gender = ""
        iris_date = ""

        #get total number of files in directory, and build table from that information
        path, dirs, files = next(os.walk('database'))
        num_users = len(files)

        #create database table
        self.table = QTableWidget()
        self.table.setRowCount(num_users+30)
        self.table.setColumnCount(7)
        self.table.resize(1624, 800)
        self.table.setWindowTitle("REGISTERED USERS DATABASE")
        self.table.horizontalHeader().setStretchLastSection(True)
        col_headers = ['ID NUM.', 'NAME', 'AGE', 'GENDER', 'DATE REGIST.', 'CURRENT DATE', 'INFO.']
        self.table.setHorizontalHeaderLabels(col_headers)
        self.table.show()


        #loop through folder until Registered IRIS is found
        row = 0
        for f_name in os.listdir('database'):
            image_name = f_name
            get_id_from_database = image_name.split("-")
            iris_f_name = f_name
```

Figure 5.2.5: Database Template

CHAPTER 6

TESTING

6.1 INTRODUCTION

Software Testing is a method to check whether the actual software product matches expected requirements and to ensure that software product is Defect free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

6.2 DESIGN OF TEST CASES AND SCENARIOS

It is the process of testing the functionality and it is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as at undiscovered error. A successful test is one that uncovers an as at undiscovered error. Software testing is usually performed for one of tworeasons:

- Defect Detection
- Reliability estimation

6.2.1 BLACK BOX TESTING:

The base of the black box testing strategy lies in the selection of appropriate dataas per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. Now a days, it is becoming to routethe testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system, which makes it unfit to test application by the developer. The following are different types of techniques involved in black box testing.They are:

- Decision Table Testing
- All pairs testing
- State transition tables testing
- Equivalence Partitioning

Software testing is used in association with Verification and Validation. Verification is the checking of or testing of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques as reviews, inspections, walk-through.Validation is the process of checking what has been specified is what the user actually wanted.

- Validation: Are we doing the right job?
- Verification: Are we doing the job right?

In order to achieve consistency in the Testing style, it is imperative to have and follow a set of testing principles. This enhances the efficiency of testing within SQA team members and thus contributes to increased productivity.

6.2.2 WHITE BOX TESTING

White box testing requires access to source code. Though white box testing can be performed any time in the life cycle after the code is developed, it is a good practice to perform white box testing during unit testing phase.

In designing of database the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

At SDEI, 3 levels of software testing is done at various SDLC phases

UNIT TESTING: In which each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented

INTEGRATION TESTING: In which progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.

SYSTEM TESTING: In which the software is integrated to the overall product and tested to show that all requirements are met. A further level of testing is also done, in accordance with requirements:

REGRESSION TESTING: Is used to refer the repetition of the earlier successful tests to ensure that changes made in the software have not introduced new bugs/side effects.

ACCEPTANCE TESTING: Testing to verify a product meets customer specified requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtain.

6.2.3 Test Cases

- To make sure if the system meets user requirements or not.
- To make sure that during the operation, incorrect input, processing and output will be detected.
- To see that when correct inputs are fed to the system the inputs are correct.
- To also verify that the controls that are incorporated in the system are as intended.

Test Case 1: Registering

Test Case: Clicking Register button

Test Description: On clicking it should open front camera and detect face along with tracking eyes and iris.

Expected Result: Opening frontal camera and detecting face and iris for registration

Result: Success.

Test Case 2: Verification

Test Case: Clicking Verify button

Test Description: On clicking it should take iris from the video feed by opening front camera and verify it with the one stored in database and return if match is found or not.

Expected Result: Verification of Iris

Result: Success

Test Case 3: Viewing Database of Registered users

Test Case: Clicking Show Registered Users button

Test Description: On clicking it should open the database of all registered userd along with their information given at the time of registration.

Expected Result: Database of Registered Users

Result: Success.

Test Case 4: Closing Application

Test Case: Clicking Close button

Test Description: On clicking it should quit the application

Expected Result: Closed application

Result: Success

CHAPTER 7

OUTPUT SCREENS

1. Initially when the application is started this is the Interface:

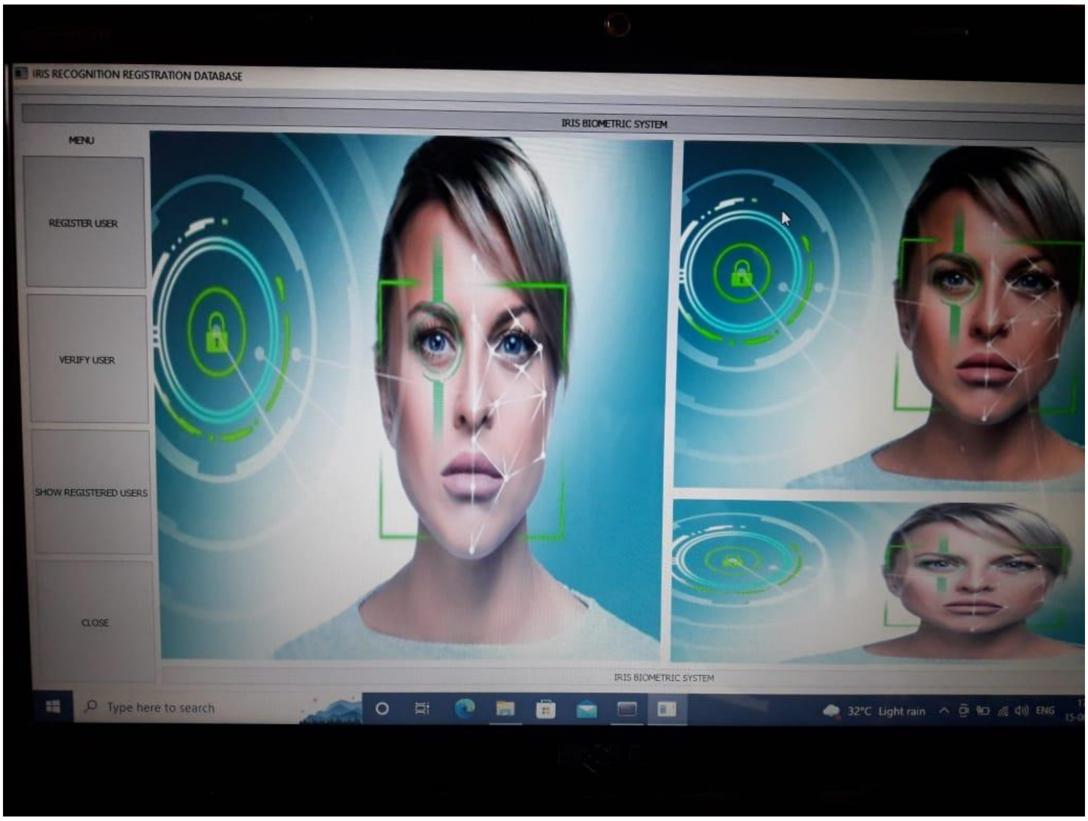


Figure 7.1: User Interface of Iris Biometric System

2. When Register Button is clicked it tries to detect the face

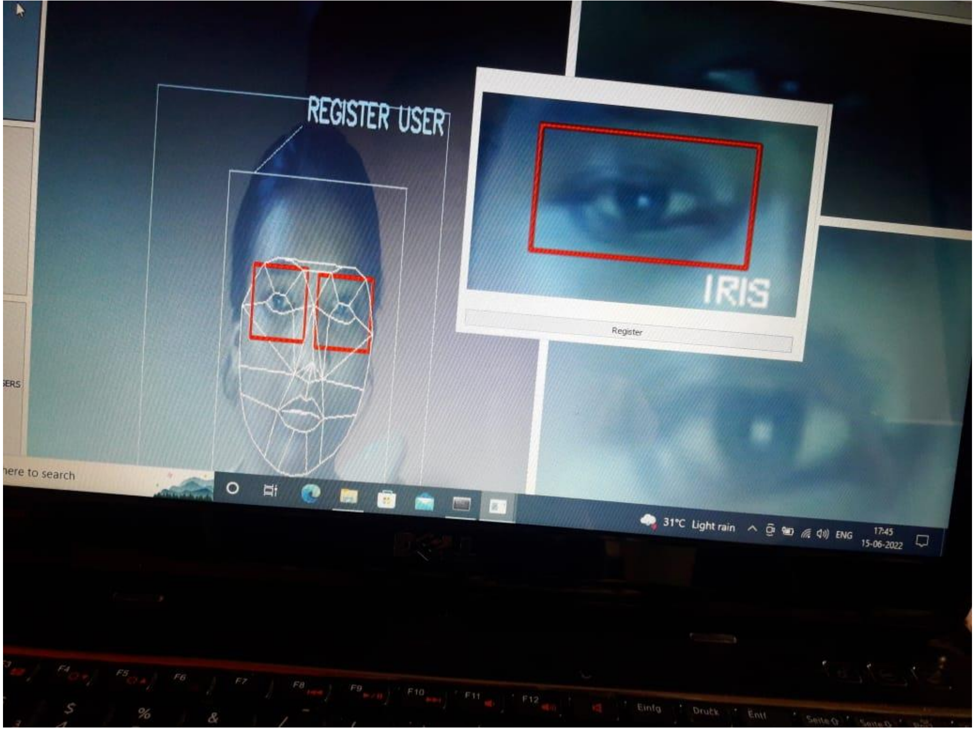
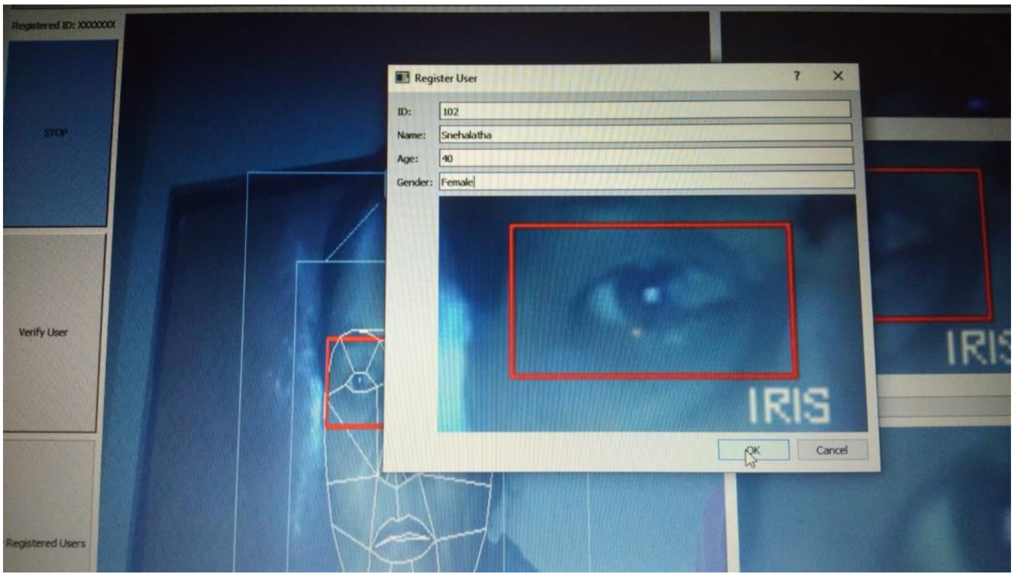


Figure 7.2 On clicking Register button

3. Pop up to enter users information for Registration



ID: 102
Name: Snehalatha
Age: 40
Gender: Female

Figure 7.3: Entering users information

4. When Verify button is clicked it takes users iris for verification through video feed

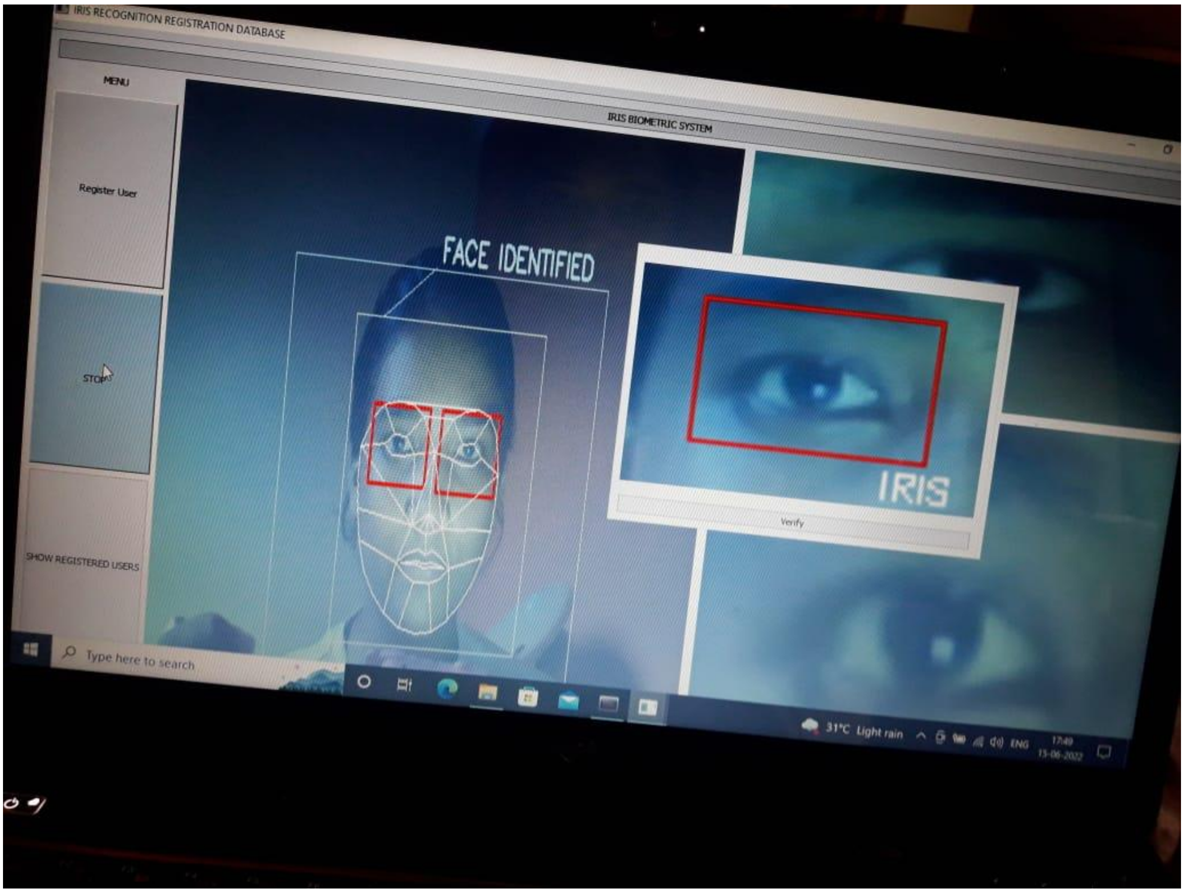
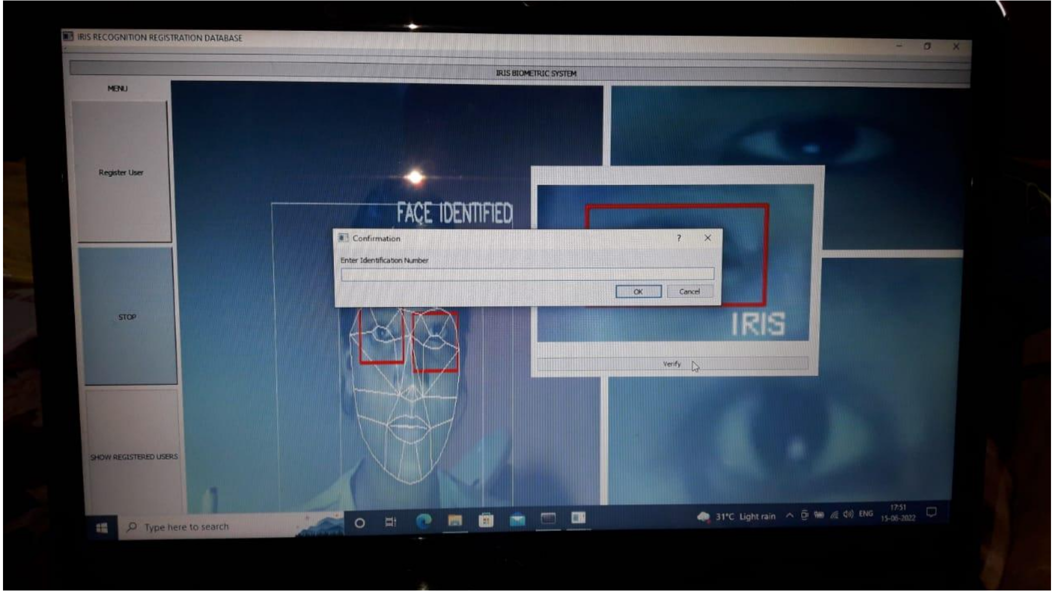


Figure 7.4: On clicking Verify button

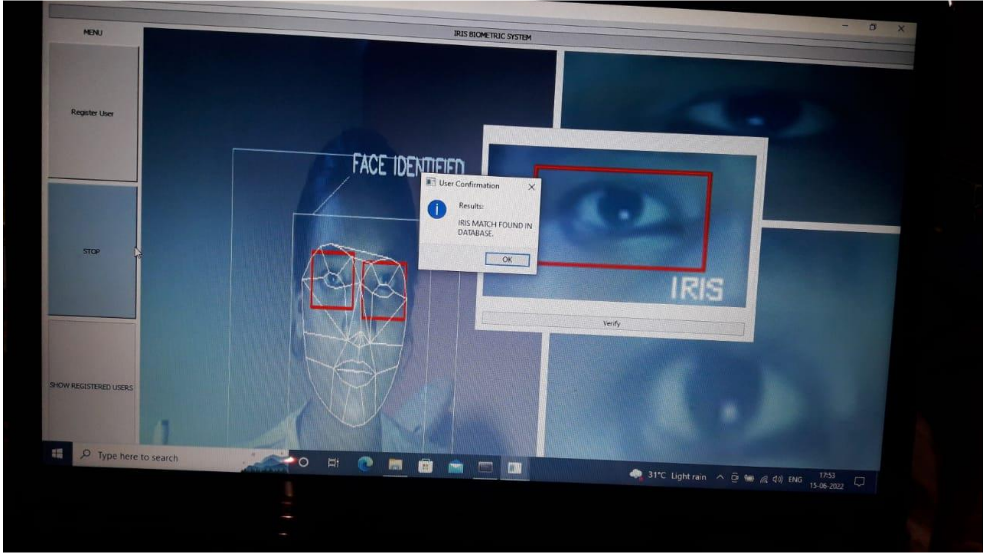
5. Pop up asking for user id for verification(for extra authentication)



Enter Identification Number

Figure 7.5: Verification through user id

6. If iris match is found message is displayed

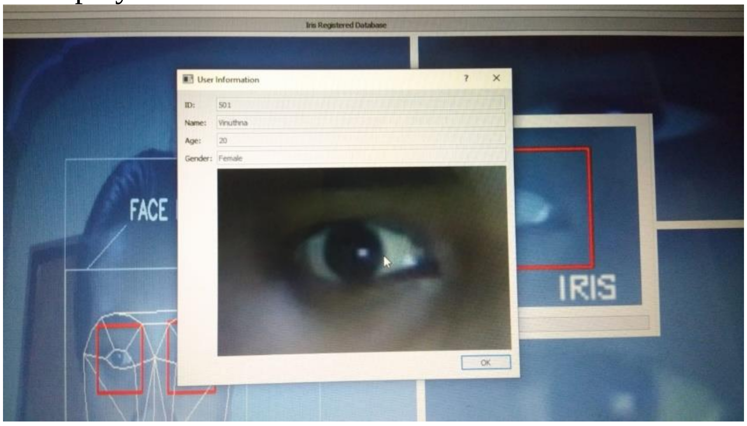


Results:

IRIS MATCH FOUND IN DATABASE.

Figure 7.6: Iris match found case

7. Display user information



ID: 501

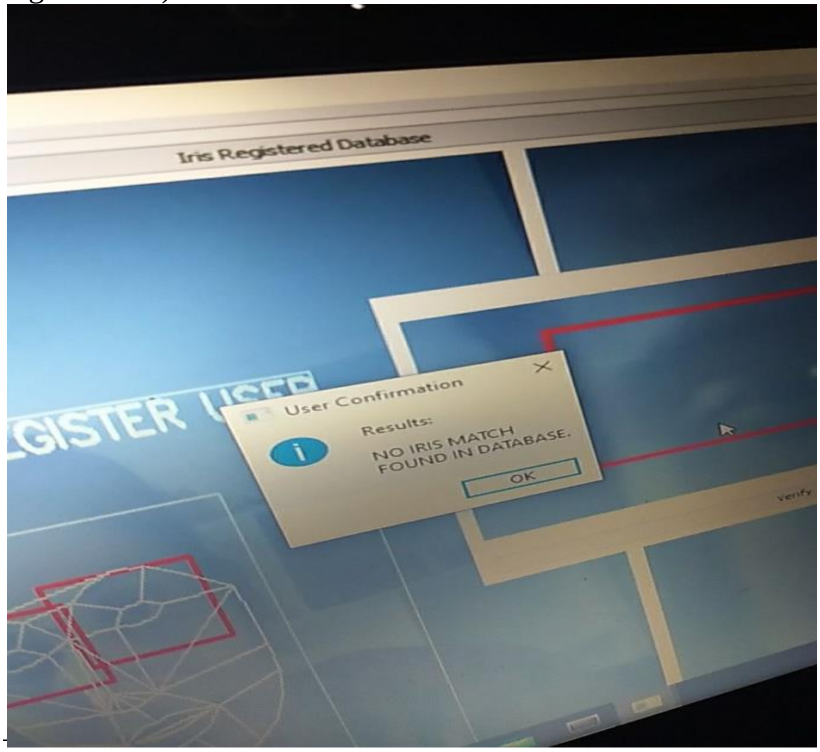
Name: Vinuthna

Age: 20

Gender: Female

Figure 7.7: User Information

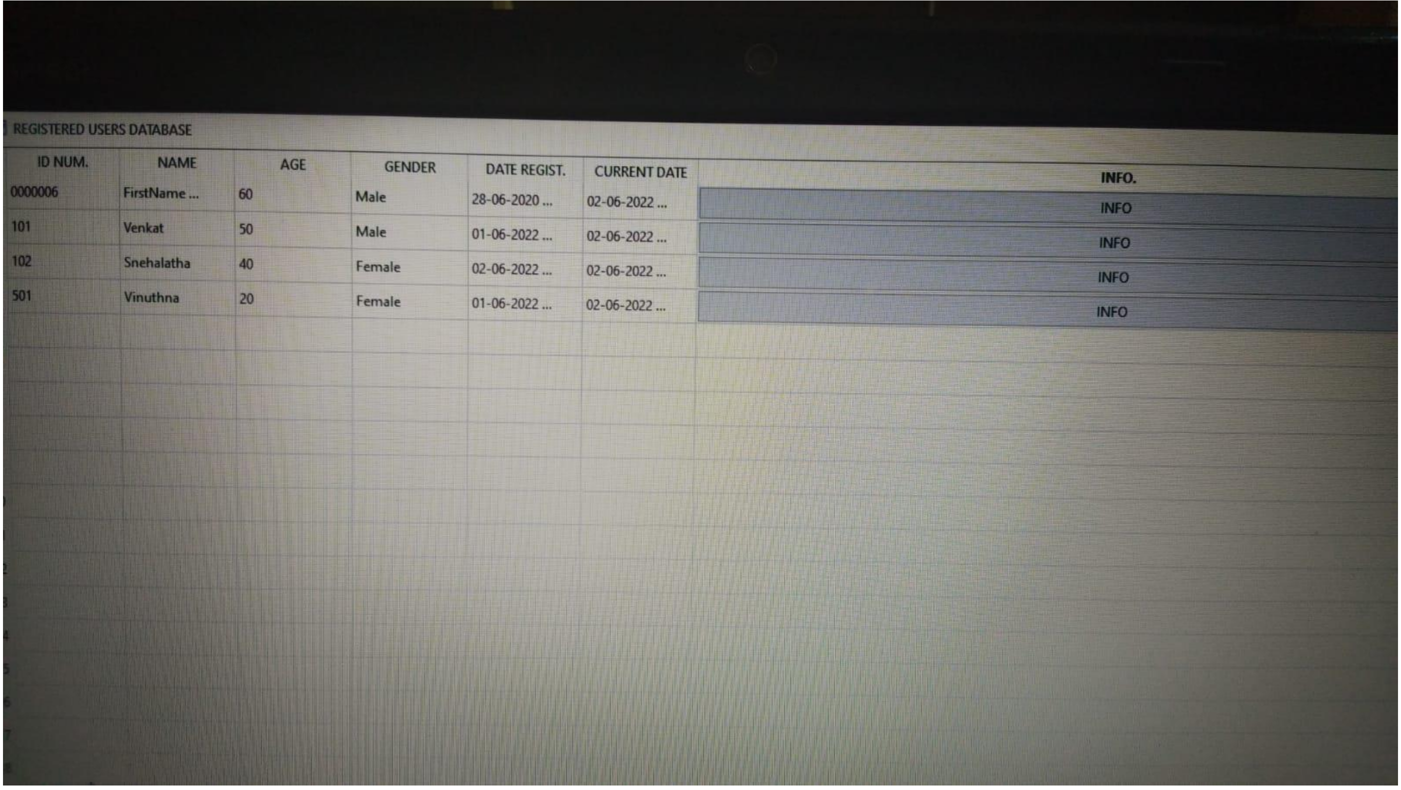
8. If iris match is not found(this also happens if users iris is registered but if id does not match with registered id)



Results:
NO IRIS MATCH
FOUND IN DATABASE.

Figure 7.8: Iris match not found case

9.When Show registered users button is clicked

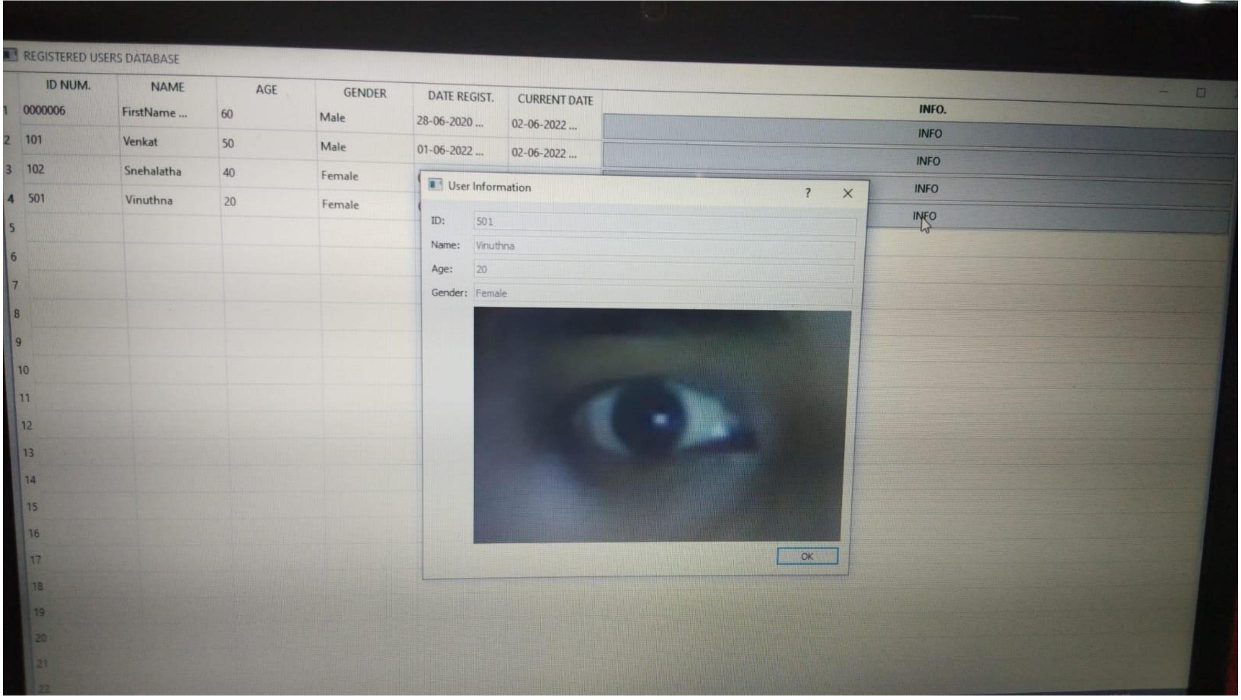


REGISTERED USERS DATABASE

ID NUM.	NAME	AGE	GENDER	DATE REGIST.	CURRENT DATE	INFO.
0000006	FirstName ...	60	Male	28-06-2020 ...	02-06-2022 ...	INFO
101	Venkat	50	Male	01-06-2022 ...	02-06-2022 ...	INFO
102	Snehalatha	40	Female	02-06-2022 ...	02-06-2022 ...	INFO
501	Vinuthna	20	Female	01-06-2022 ...	02-06-2022 ...	INFO

Figure 7.9: Registered users database

10.When Info button in Show registered users is clicked



REGISTERED USERS DATABASE

	ID NUM.	NAME	AGE	GENDER	DATE REGIST.	CURRENT DATE	INFO.
1	0000006	FirstName ...	60	Male	28-06-2020 ...	02-06-2022 ...	INFO
2	101	Venkat	50	Male	01-06-2022 ...	02-06-2022 ...	INFO
3	102	Snehalatha	40	Female			INFO
4	501	Vinuthna	20	Female			INFO

User Information

ID: 501

Name: Vinuthna

Age: 20

Gender: Female

Figure 7.10: Registered users info in database

CHAPTER 8

CONCLUSION

The use of iris recognition system has been seen in various areas of life such as airport, crime detection, business application, various research firm and industries, experts anticipate the growth of iris recognition system. The use of iris recognition system is expanding worldwide as thepublic has been oriented about the necessity of iris recognition system. For instance, iris recognition system is used in banks where it is incorporated into the Automated Teller Machines (ATMs). During the last few years, one can easily observe that human identity recognitionbased on biometrics was made one of the top technologies. In particular, it is observable in the case of mobile devices where face or fingerprint is used. However, in the next few years, iris can take their place. It is connected with the fact that novel devices have much more precise cameras as well as iris is really hard to spoof. On the basis of diversified sources, one can claim that iris can be described by more than 250 specific elements. It is much more than in the case of fingerprint or face.

In this project I have designed an IRIS BIOMETRIC SYSTEM, that uses computer vision technology to do facial recognition and to capture the IRIS of an individual, and store their information in a database for biometric authentication. The DLIB library is used to detect and recognize the face structure to pinpoint the eyes, and then capture the eyes through openCV's video capturing functions.The captured IRIS is stored into an image along with input from the user. During verification an associated ID inputted by the user is entered when verifying the IRIS. The IRIS is captured through the video feed, and sent to a pattern feature-like matching system setup using ORB's detector.

The detector does pattern matching of both the user's IRIS through the feed, and the stored image of the user's IRIS from the key points and the authentication is done only if the match is found.

CHAPTER 9

FUTURE SCOPE AND ENHANCEMENT

In future this application can be extended to identify if the iris being captured is of a human beings or of an animal or a toy as this is little bit practically complicated to implement it is the study under research.This could also now extended by implementing in any other confidential applications like crime detection or army verification systems to avoid crimes and alert as it works with Iris based authentication and has the least replication possible. We can also extend this implementation by using more complex deep neural networks for achieving better security. To make the study more useful and effective the following suggestion have been proposed for further improvements in this area. To develop improved algorithms and data capturing sensors to reduce the level of failure to enrol and failure to acquire rate. To work on optimization of the code, so that the segmentation software can run in real time applications.

CHAPTER 10

BIBLIOGRAPHY

[1]https://www.jetir.org/view?paper=JETIR2005086 **Iris recognition system"**, International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN:2349-5162, Vol.7, Issue 5, page no.578-584, May-2020

[2]Gajendra Singh Chandel, Ankesh Bhargava , "Identification of People by Iris Recognition", International Journal of Science and Modern Engineering (IJISME) ISSN: 2319- 6386, Volume-1, Issue-4, March 2013.

[3]https://www.researchgate.net/publication/358798672_Towards_more_accurate_iris_recognition_system_by_using_hybrid_approach_for_feature_extraction_along_with_classifier

[4]https://www.researchgate.net/publication/312490387_Iris_Recognition_Technology_Implementation_Application_and_Security_Consideration

[5] Richard Yew Fatt Ng, Yong Hour Tay, Kai Ming Mok, "AReview of Iris Recognition Algorithm", DOI: 978--4244- 2328-6/08.

[6] Ujwalla Gawande**,** Mukesh Zaveri, Avichal Kapur, Improving Iris Recognition Accuracy by Score Based Fusion Method", International Journal of Advancements in Technology (IJoAT) http://ijict.org/ ISSN 0976-4860.

[7] Daugman, J., Dowing, C. "Epigenetic randomness, complexity, and singularity of human iris patterns. In: Preceding soft the Royal Society, B, 268, Biological Sciences, pp.1737–1740(2001).

[8] Daugman, J." Statistical richness of visual phase information: update on recognizing Persons by iris patterns. Int. J. Comput.Vis.45 (1), 25–38(2001).

[9] John Daugman, "How Iris Recognition Works", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 14, no. 1, January 2004.

[10] Khattab M. Ali Alheeti, "Biometric Iris Recognition Based on Hybrid Technique", International Journal on Soft Computing (IJSC) Vol.2, No.4, November 2011 DOI : 10.5121/ijsc.2011.2401 1.

[11] Ouda O, Chaoui S, Tsumura N (2020) Security evaluation of negative iris recognition. IEICE Trans Inf Syst 103(5):1144–1152