CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

In this newly complicated world of terrorism, identity theft, and rampant consumer fraud, biometrics has been heralded as a key technology for identity management, and hence security. As never before has identity management been so important. Governments and enterprises of all sizes have become much more vigilant regarding security. There is always a need to re-examine and potentially improve security, and biometrics is attracting growing interest as fraud increases and the conventional authentication methods PINs, passwords, and identity cards prove inadequate to counter the growing threats.

Biometric tools have become prominent differentiators for multiple applications in a variety of markets. The use of biometrics offers no panacea to completely remedy society's threats, and it provides no guarantee against terrorist activities. However, biometric technologies remain a critically important component of the total solution. The biometric authentication market has emerged and is expanding at an increasing rate. Biometric systems are proliferating. The diversity of the various modalities and the many false claims of their promoters and detractors alike have somewhat clouded the market with at best some misinformation and at worst a public concern that this new technology is somehow menacing and will restrict freedoms. Unfortunately, many of the key benefits of biometrics have become obfuscated due to unfortunate sensationalism and myths that have surrounded biometric solutions.

1.2 MOTIVATION

The main motivation behind this choice of fingerprint and eye characteristics for a multi-biometric authentication system is that fingerprint is the oldest and most widely adopted biometric technology and, as a result, is the most mature of all biometric technologies, eye recognition is proofed that it is most accurate and hygienic biometric technology among others, this is reported in Biometric Product Testing Final Report.

1.3 PROBLEM STATEMENT

The majority of deployed biometric systems today use information from a single biometric technology for verification or identification. Large-scale biometric systems have to address additional demands such as larger population coverage and demographic diversity, varied deployment environment, and more demanding performance requirements. Today's single modality biometric systems are finding it difficult to meet these demands, and a solution is to integrate additional sources of information to strengthen the decision process. A multi biometric system combines information from multiple biometric traits, algorithms, sensors, and other components to make a recognition decision.

1.4 PROJECT OBJECTIVES

- To obtain the authentication accuracy of 90% using more than one biometric trait.
- To improve the quality of feature extraction of fingerprint and eye so as to increase the performance of the system and make it reliable.

CHAPTER 2

LITERATURE SURVEY

2.1 SURVEY PAPERS

[1]. Kamer Vishi, S¸ule Yildirim Yayilgan, "Multimodal Biometric Authentication using Fingerprint and eye Recognition in Identity Management", IEEE Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2013.The proposed method is evaluated using two fingerprint databases and two eye databases. The fused score is used to classify an unknown user with genuine or imposter.

[2]. Debanjan Sadhya, Parth Pahariya, Rishi Yadav, Apoorv Rastogi, Ayush Kumar, Lakshya Sharma-" A Multimodal Biometric Database" Biometrics is an automated authentication mechanism that allows the identification or verification of individuals based on unique physiological and behavioral characteristics. In addition to novel biometric recognition frameworks and protocols, standard databases containing sample biometric traits are essential for validating the obtained results. In this paper, we introduce a new multimodal database named BioSoft which consists of biometric data collected from 75 individuals. In comparison to the already existing databases, BioSoft contains a set of 23 soft biometric traits corresponding to each enrolled individual. This property makes our database very useful due to the unavailability of any other manually extracted multimodal database incorporating soft biometric characteristics. Additionally, the primary biometric modalities of face, ear, eye, voice, handwriting and fingerprints (obtained from two different sensors) are present in this database. Thus our database contains both physiological and behavioral characteristics of individuals, thus making it applicable for validating a wide variety of approaches.

[3]. A. Jaya Lakshmi, I. Ramesh babu-"PKI Key Generation using Multimodal Biometrics Fusion of Fingerprint and Eye" In this paper we introduce an efficient approach for the secure PKI key generation on the idea of multiple modalities Eye and fingerprint. Contributions include a general approach for trivialities purpose generation, distinguishable eye feature generation and a PKI key generation mechanism. The image processing techniques are used to extract a biometric measurement from the fingerprint and eye. Within the proposed technique, GPOF (Generation of PKI key while not Fusion) minutiae points are extracted from the Finger print and key are generated from them. Within the same manner options are extracted from the eye and key are generated from them. Currently generate one prime variety from the key generated using finger print using Probabilistic primality algorithm. Within the same manner generate another prime variety from the key generated using eye using Probabilistic Primality algorithm. Currently generate PKI keys using RSA algorithm that uses these two prime numbers.

[4]. A. Jagadeesan, T. Thillaikkarasi, K. Duraiswamy-"Protected Bio-Cryptography Key Invention from Multimodal Modalities" Human users find hard to remember lengthy cryptographic keys. Therefore, researchers, for a long time period, have been investigating ways to use biometric features of the user rather than memorable password or passphrase, in an attempt to produce tough and repeatable cryptographic keys. Our goal is to integrate the volatility of the user's biometric features into the generated key, so as to construct the key unpredictable to a hacker who is deficient of important knowledge about the user's biometrics. In our earlier research, we have incorporated multiple biometric modalities into the cryptographic key generation to provide better security. In this paper, we propose an efficient approach based on multimodal biometrics (Eye and fingerprint) for generating a secure cryptographic key, where the security is further enhanced with the difficulty of factoring large numbers. This paper deals with two approaches of extracting minutiae points and texture properties from fingerprint and eye and gives the optimal solution. At first, the features, minutiae points and texture properties are extracted from the fingerprint and eye images respectively. Then, the extracted features are fused at the feature level to obtain the multi-biometric template.

[5]. Bir Bhanu, V. Govindaraju-"Multi biometrics for Human Identification" In today's security-conscious society, real-world applications for authentication or identification require a highly accurate system for recognizing individual humans. The required level of performance cannot be achieved through the use of a single biometric such as face, fingerprint, ear, eye, palm, gait, or speech. Fusing multiple biometrics enables the indexing of large databases, more robust performance, and enhanced coverage of populations. Multiple biometrics is also naturally more robust against attacks than single biometrics. This book addresses a broad spectrum of research issues on multi-biometrics for human identification, ranging from sensing modes and modalities to fusion of biometric samples and combination of algorithms. It covers publicly available multi-biometrics databases, theoretical and empirical studies on sensor fusion techniques in the context of biometrics authentication, identification, and performance evaluation and prediction.

[6]. AnilJain,KarthikNandakumar,ArunRos-"Score normalization in multimodal biometric systems" Multimodal biometric systems consolidate the evidence presented by multiple biometric sources and typically provide better recognition performance compared to systems based on a single biometric modality. Although information fusion in a multimodal system can be performed at various levels, integration at the matching score level is the most common approach due to the ease in accessing and combining the scores generated by different matchers. Since the matching scores output by the various modalities are heterogeneous, score normalization is needed to transform these scores into a common domain, prior to combining them. In this paper, we have studied the performance of different normalization techniques and fusion rules in the context of a multimodal biometric system based on the face, fingerprint and hand-geometry traits of a user. Experiments conducted on a database of 100 users indicate that the application of min–max, *z*-score, and tanh normalization schemes followed by a simple sum of scores fusion method results in better recognition performance compared to other methods. However, experiments also reveal that the min–max and z-score normalization techniques are sensitive to outliers in the data, highlighting the need for a robust and efficient normalization procedure like the tanh normalization. It was also observed that multimodal systems utilizing user-specific weights perform better compared to systems that assign the same set of weights to the multiple biometric traits of all users.

CHAPTER 3

REQUIREMENT SPECIFICATION

The system requirement specification (SRS) describes all data, functional, and behavioral requirements of the software and hardware under development or production. It includes functional and non-functional requirements for the software and hardware to be developed.

The functional requirements include what the software and hardware should do and the non-functional requirements include the constraint on the design and implementation. Requirements must be measurable, testable, related to identified needs or opportunities, and defined to a detail level sufficient to system design.

3.1 FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs (see also software). Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases.

The various methods used in this project are as follows: -

- Ability to detect the eye and fingerprint.
- Match the eye and fingerprint.
- Turn on the motor if both are matched.

3.2 NON-FUNCTIONAL REQUIREMENTS

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process and standards. Non-functional requirements often apply to the system as a whole.

3.2.1 Dependability

The dependability of a computer system is a property of the system that equates to its trustworthiness. Trustworthiness essentially means the degree of user confidence that the system will operate as they expect and that the system will not ‘fail’ in normal use.

3.2.2 Availability

The ability of the system is to deliver services when requested. There is no error in the program while executing the program.

3.2.3 Reliability

The ability of the system to deliver services as specified. The program is compatible with all types of operating system without any failure.

3.2.4 Safety

The ability of the system to operate without catastrophic failure. This program is user friendly and it will never affect the system.

3.2.5 Security

The ability of the system to protect itself against accidental or deliberate intrusion. Our system is highly secure.

3.3 SYSTEM REQUIREMENTS

Software Requirement

- Raspbian Jessie
- Python
- Open CV

Hardware Requirements

- Raspberry Pi
- Pi Camera
- Relay Driver
- Gear Motor
- USB Cable

3.3.1 Raspberry Pi

Raspberry Pi is a credit-card sized computer manufactured and designed in the United Kingdom by the Raspberry Pi foundation with the intention of teaching basic computer science to school students and every other person interested in computer hardware, programming and DIY-Do-it Yourself projects.

The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU and was originally shipped with 256 megabytes of RAM, later upgraded (Model B & Model B+) to 512 MB. It does not include a built-in hard disk or solid-state drive, but it uses an SD card for booting and persistent storage, with the Model B+ using a MicroSD.

The Foundation provides Debian and Arch Linux ARM distributions for download. Tools are available for Python as the main programming language, with support for BBC BASIC (via the RISC OS image or the Brandy Basic clone for Linux), C, Java and Perl.

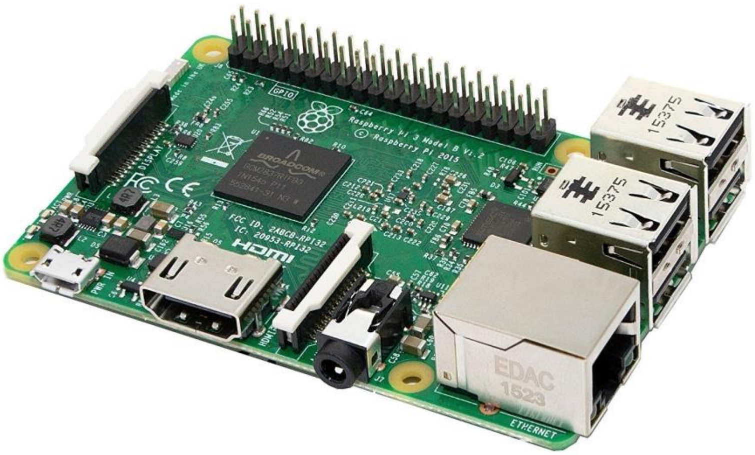


Fig 3.1: Raspberry Pi Model.

PORTS, PINS AND THEIR USES

Fig 3.1 shows the Raspberry Pi model and the ports. The following are the ports on the Raspberry Pi board and some of their employments. The ports may likewise be utilized for different purposes than recorded beneath.

USB	Mainly utilized for peripherals like Keyboard, mouse and a Wi-Fi Adapter. A

	controlled USB center point can be associated and be extended
HDMI	This is the High Definition Multimedia Interface [HDMI] and is use to associate with a Display unit like TV or Monitor or some of the time a projector
Stereo Audio	Audio associations utilizing a 3.5 mm jack
SD Card	SD card is utilized as a boot gadget and furthermore relentless capacity. More stockpiling can be connected to the USB
Micro USB	The miniaturized scale USB port is utilized for providing energy to the unit
CSI Connector	CSI [Camera serial Interface] is utilized for associating a camera to the unit
Ethernet	Used for interfacing with a system utilizing a system link
DSI Connector	DSI [Digital serial Interface] is utilized for associating aLC

One other essential stick is the GPIO.GPIO remains for General Purpose Input and Output.

There are 40 Pins on a Model B altogether.

- There are three power supply pins [3.3v, 5.0v and 0v].

- 26 GPIO pins

3.3.2 Raspberry Pi with Raspbian OS

Below are some of the Operating systems that a Pi can run but in this project we use only Raspbian.

Linux	There are three official Linux flavors available for download namely
	Debian [Raspbian] *Recommended

	ArchLinux
	Pidora [Based on Fedora]
RISC OS	A retro looking 1080p GUI designed by the ARM designers. RISC was more common during the 90's
Firefox OS	A new OS by the Firefox team. Pretty much a combination of Firefox and PTXdist-built Linux
Plan 9	Unix like OS by the by the Bell Labs, created by the UNIX creators
Android	No explanation necessary, but this hasn't gone beyond a 2.3 build and a bit too slow.

Raspbian OS is one of the authority working frameworks accessible for nothing to download and utilize. The framework depends on Debian Linux and is streamlined to work effectively with the Raspberry Pi PC. As we definitely know an OS is an arrangement of essential projects and utilities that keeps running on a predefined equipment, for this situation the Pi. Debian is extremely lightweight and settles on an incredible decision for the Pi. The Raspbian incorporates devices for perusing, python programming and a GUI desktop.

The Raspbian desktop condition is known as the "Lightweight X11 Desktop Environment" or in short LXDE. This has a genuinely appealing UI that is constructed utilizing the X Window System programming and is a natural point and snap interface. We might look more into how to introduce and utilize this OS in the following segment.

3.3.3 Operating System

The Raspberry Pi primarily uses Linux kernel-based Operating systems. The ARM11 is based on version 6 of the ARM which is no longer supported by several popular versions of Linux, including Ubuntu. The install manager for Raspberry Pi is NOOBS. The OSs included with NOOBS is:

- ✓ Archlinux ARM

- ✓ OpenELEC
- ✓ Pidora (Fedora Remix)
- ✓ Raspbmc and the XBMC open source digital media center
- ✓ RISC OS – The operating system of the first ARM-based computer

Raspbian (recommended) – Maintained independently of the Foundation, based on ARM hard-float (armhf)-Debian 7 'Wheezy' architecture port, that was designed for a newer ARMv7 processor whose binaries would not work on the Raspberry Pi, but Raspbian is compiled for the ARMv6 instruction set of the Raspberry Pi making it work but with slower performance. It provides some available dev software packages, pre-compiled software bundles. A minimum size of 2 GB SD card is required, but a 4 GB SD card or above is recommended. There is a Pi Store for exchanging programs. The ‘Raspbian Server Edition (RSEv2.4)', is a stripped version with other software packages bundled as compared to the usual desktop computer oriented Raspbian.

Applications of the Raspberry Pi can be given as follows:

- ➢ Teaching programming concepts.
- ➢ Teaching hardware interfacing.
- ➢ Raspberry Pi being very cost effective can be deployed in large numbers in underdeveloped and developing countries like Africa, India, China, Brazil etc. To schools and colleges and to everyone who is interested in computers and electronics.
- ➢ It can be used in robotics for controlling motors, sensors, etc.
- ➢ It can be used as a downloading machine replacing desktop computers. It consumes very low power and can also be accessed remotely.
- ➢ It can be used as a media centre at home. Any television can be converted to a smart TV with internet capabilities with the Pi.
- ➢ It can be used for designing prototypes of DIY projects and certain embedded devices. It becomes very cheap option for testing and evaluation purpose.
- ➢ Can be used in creating and handling small servers.

3.3.4 Fingerprint sensor

The R307 Reader is a fingerprint reader featuring an elegant, sleek design with a soft, cool blue glow and, of course, the unsurpassed performance Digital Persona is known for. Made for power-users and shared environments, the R307 is the natural choice for those that want and need the very best. Here’s a look at just some of its features and benefits:

Blue LED Soft, cool blue glow fits into any environment. Provides a pleasing presence; doesn't compete in low light environments, such as restaurants, or conflict with alarm condition colors, such as in healthcare.

Small form factor Conserves valuable desk space.

Rugged construction High-quality metal casing weighted to resist unintentional movement.

Special undercoating Stays where you put it because of a special undercoating.

Rotation invariant Touch it from any direction, it still provides a high quality image and matching performance, perfect for shared environments.

Excellent-image quality High-quality optics ensure best image every time.

3.3.5 Pi Camera

In order to meet the increasing need of Raspberry Pi compatible camera modules. The ArduCAM team now released a revision C add-on camera module for Raspberry Pi which is fully compatible with official one. It optimizes the optical performance than the previous Pi cameras, and gives user a much clear and sharp image. Also it provides the FREX and STROBE signals which can be used for multi-camera synchronize capture with proper camera driver firmware.

Features

- High-Definition video camera for Raspberry Pi Model A/B/B+ and Raspberry Pi 2
- Omni vision OV5647 sensor in a fixed-focus module with replaceable Lens
- Lens holder: M12x0.5 , CS mount or C mount
- 5MPixel sensor
- Integral IR filter
- Still picture resolution: 2592 x 1944
- Max video resolution: 1080p
- Max frame rate: 30fps
- Support FREX/ STROBE feature
- Size: 36 x 36 mm
- 15 cm flat ribbon cable to 15-pin MIPI Camera Serial Interface (CSI) connector

3.3.6 Gear motor

Fig 3.2 shows the picture of gear motor. A small motor (ac induction, permanent magnet dc, or brushless dc) designed specifically with an integral (not separable) gear reducer (gear head). The end shield on the drive end of the motor is designed to provide a dual function. The side facing the motor provides the armature/rotor bearing support and a sealing provision through which the integral rotor or armature shaft pinion passes. The other side of the end shield provides multiple bearing supports for the gearing itself, and a sealing and fastening provision for the gear housing. This construction provides many benefits for a user and eliminates the guesswork of sizing a motor and gear reducer on your own.

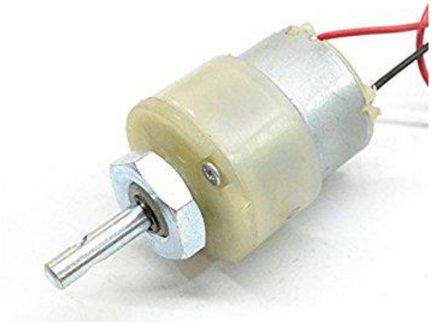


Fig 3.2: Gear motor

3.3.7 Relay drivers

Relays are components which allow a low-power circuit to switch a relatively high current on and off, or to control signals that must be electrically isolated from the controlling circuit itself.

To make a relay operate, you have to pass a suitable pull-in and holding current (DC) through its energizing coil. The circuit diagram is shown in Fig 3.3 and generally relay coils are designed to operate from a particular supply voltage - often 12V or 5V, in the case of many of the small relays used for electronics work. In each case the coil has a resistance which will draw the right pull-in and holding currents when it is connected to that supply voltage. So the basic idea is to choose a relay with a coil designed to operate from the supply voltage you're using for your control circuit (and with contacts capable of switching the currents you want to control), and then provide a suitable relay driver circuit so that your low-power circuitry can control the current through the relay coil. Typically this will be somewhere between 25mA and 70mA.

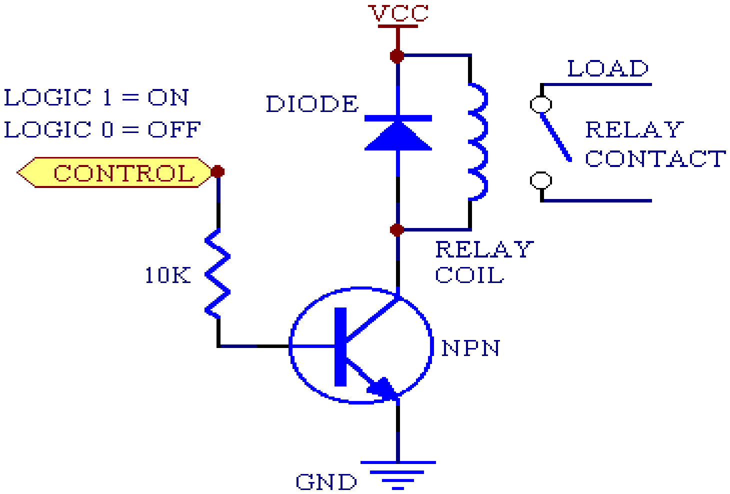


Fig 3.3: Circuit of the relay

3.3.8 Arduino

Arduino is an open source microcontroller which can be easily programmed, erased and reprogrammed at any instant of time. Introduced in 2005 the Arduino platform was designed to provide an inexpensive and easy way for hobbyists, students and professionals to create devices that interact with their environment using sensors and actuators.

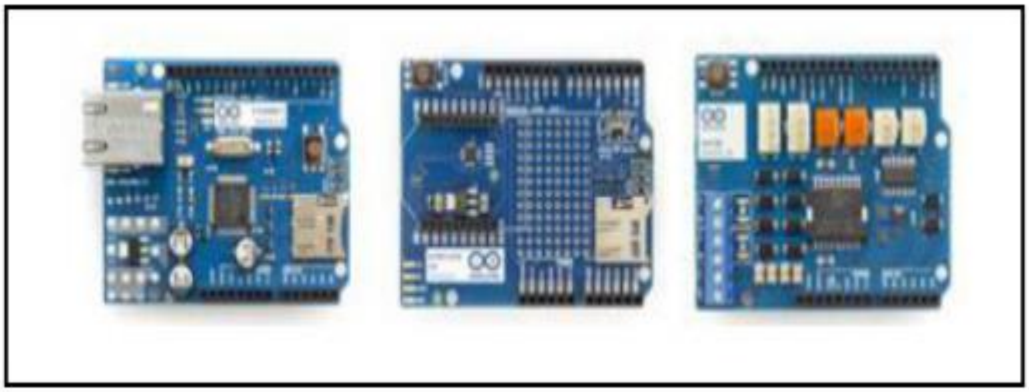


Fig 3.4: Arduino shields- Ethernet, Wireless and Motor Driver

Some of the most commonly used Shields are shown in Fig 3.4 and explained below:

- Arduino Ethernet shield: It that allows an Arduino board to connect to the internet using the Ethernet library and to read and write an SD card using the SD library.

- Arduino Wireless shield: It allows your Arduino board to communicate wirelessly using Zigbee.

- Arduino Motor Driver Shield: It allows your Arduino boards to interface with driver of a motor etc.

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Biometric technologies vary in capability, performance, and reliability. The success of a given biometric modality depends not only on the effectiveness of the technology and its implementation, but also on the total security solution for which any biometric system comprises only a part. The next several years will be exciting for the biometric market. We can expect increased user acceptance and demand as biometrics continue to become more user friendly and more reliable. Improved technology and biometric need are converging. There should be significant growth in each of the various biometric modalities, as well as in multimodal biometrics.

Because of their security, speed, efficiency, and convenience, biometric authentication systems have the potential to become the new standard for access control. Biometrics replaces or supplements knowledge and possession authentication with a person’s physical or behavioral characteristics. Biometrics can be used in any situation where identity badges, PINs/passwords, or keys are needed. Biometrics offers some clear advantages over traditional identity methods:

- Biometric traits cannot be lost, stolen, or borrowed.
- Generally, physical human characteristics are much more difficult to forge than security codes, passwords, badges, or even some encryption keys.

4.2 PROPOSED SYSTEM

We propose a new multi-modal biometric authentication approach using eye and fingerprint images as biometric traits. We fuse these two modalities at score-level by fusing different comparison scores from fingerprint and eye traits into a single score by combination approach. Since comparison scores that are generated from these uncorrelated and independent modalities are not homogeneous, score normalization step is essential to transform comparison scores into a common scale before fusing them. The individual comparison scores obtained from the eye and fingerprints are combined at score-level using three normalization methods (Min-Max, Z-Score, Hyperbolic Tangent) and four fusion approaches (Minimum Score, Maximum Score Simple Sum and User Weighting). The fused-score is utilized to classify an unknown user into the genuine or impostor. We demonstrate that fusion based at score level achieves high performance on different multimodal biometric databases involving fingerprint and eye modalities. In addition, we have analyzed the properties (performance, robustness and efficiency) of score normalization and fusion methods. Furthermore, we have analyzed the quality of fingerprint and eye databases. Finally, we show that fusion of uncorrelated modalities such as fingerprint and eye achieves better accuracy and security compared to unimodal biometric systems.

4.3 SCOPE

The main motivation behind this choice of fingerprint and eye characteristics for a multi-biometric authentication system is that fingerprint is the oldest and most widely adopted biometric technology and, as a result, is the most mature of all biometric technologies, eye recognition is proofed that it is most accurate and hygienic biometric technology among others, this is reported in Biometric Product Testing Final Report.

4.4 PROGRAMMING LANGAUAGE

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages. Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive −You can actually sit at a Python prompt and interact with the interpreter directly to write your programs. Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python uses automatic memory allocation and garbage collection where else C++ requires the programming to allocate memory and to collect garbage. Python is designed to make distributed computing easy with the networking capability that is inherently integrated into it. Python is one of the first programming languages to consider security as part of its design.

4.5 HAAR FEATURE-BASED CASCADE CLASSIFIERS

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.

Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, Haar features shown in the Fig 4.1 are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under the white rectangle from sum of pixels under the black rectangle.

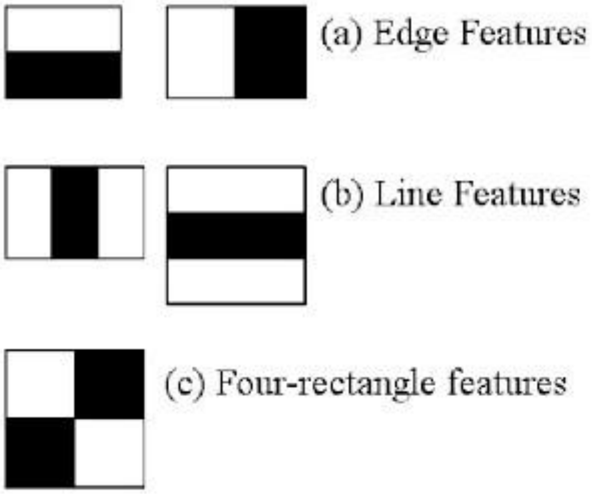


Fig 4.1: Haar features

Now, all possible sizes and locations of each kernel are used to calculate lots of features. (Just imagine how much computation it needs? Even a 24x24 window results over 160000 features). For each feature calculation, we need to find the sum of the pixels under white and black rectangles. To solve this, they introduced the integral image. However large your image, it reduces the calculations for a given pixel to an operation involving just four pixels.

OpenCV already contains many pre-trained classifiers for face, eyes, smiles, etc. Those XML files are stored in the **opencv/data/haarcascades/** folder. First we need to load the required XML classifiers. Then load our input image (or video) in **grayscale** mode. Code is show below:

```
import numpy as np

import cv2 as cv

face_cascade = cv.CascadeClassifier('haarcascade_frontalface_default.xml')

eye_cascade = cv.CascadeClassifier('haarcascade_eye.xml')

img = cv.imread('sachin.jpg')

gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
```

Now we find the faces in the image. If faces are found, it returns the positions of detected faces as Rect(x,y,w,h). Once we get these locations, we can create a ROI for the face and apply eye detection on this ROI (since eyes are always on the face !!!).

```
faces = face_cascade.detectMultiScale(gray, 1.3, 5)
for (x,y,w,h) in faces:
cv.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
roi_gray = gray[y:y+h, x:x+w]
roi_color = img[y:y+h, x:x+w]
eyes = eye_cascade.detectMultiScale(roi_gray)
for (ex,ey,ew,eh) in eyes:
```

```
cv.rectangle(roi_color,(ex,ey),(ex+ew,ey+eh),(0,255,0),2)
cv.imshow('img',img)
cv.waitKey(0)
cv.destroyAllWindows()
```

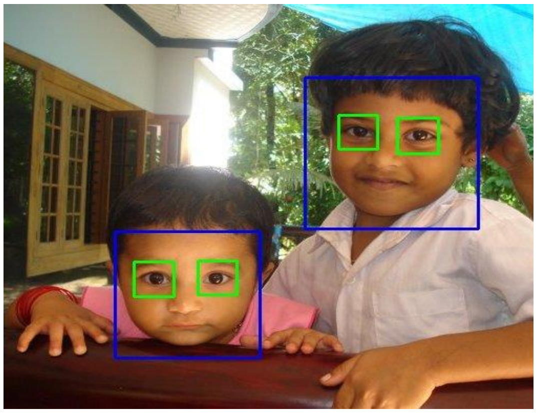


Fig 4.2: Results of Haar cascade algorithm

4.6 FINGERPRINT WITH ARDUINO

The software side of the project and how to enrol our fingerprints to the module's embedded memory in order to recognize them. We need to download some libraries. First of all we need the Adafruit Fingerprint library, the Adafruit GFX library and the Sumotoy's library for the display.

https://github.com/adafruit/Adafruit-Fingerprint-Sensor-Library

https://github.com/adafruit/Adafruit-GFX-Library

https://github.com/sumotoy/TFT_ILI9163C

First of all we have to upload the enroll example to our Arduino board. We go to File -> Examples -> Adafruit Fingerprint Sensor Library -> Enroll. With this example program we can store fingerprints in the FLASH memory of the module. We upload the sketch and we open the Serial Monitor. The program asks us to enter the ID to enroll. Then we place the finger on the sensor twice as we are instructed and the fingerprint is stored! You can store as many as 1000 fingerprints this way.

Code is as follows:

```
void loop() {  fingerprintID = getFingerprintID(); //We scan the fingerprint here
  delay(50);
  if(fingerprintID == 1) //We have found a valid fingerprint with the id 1
  {
    display.drawBitmap(30,35,icon,60,60,GREEN);
    delay(2000);
    displayUnlockedScreen();
    displayIoanna();
    delay(5000);
    display.fillScreen(BLACK);
    displayLockScreen();
  }   if(fingerprintID == 2) //We have found a valid fingerprint with the id 2  {
    display.drawBitmap(30,35,icon,60,60,GREEN);
    delay(2000);
    displayUnlockedScreen();
    displayNick();
    delay(5000);
    display.fillScreen(BLACK);
    displayLockScreen();
  }
```

We start the sensor and the display, and we check for a finger on the sensor every 50ms. If there is a finger on the sensor we request the module to search if that finger is enrolled in its memory. If it finds the fingerprint in the memory it returns that fingerprints' ID. Next it displays a welcome message and locks the screen again after a few seconds.

4.7 DATASET

A data set (or dataset) is a collection of data. Most commonly a data set corresponds to the contents of a single database table, or a single statistical data matrix, where every column of the table represents a particular variable, and each row corresponds to a given member of the data set in question. The data set lists values for each of the variables, such as height and weight of an object, for each member of the data set. Each value is known as a datum. The data set may comprise data for one or more members, corresponding to the number of rows. The term data set may also be used more loosely, to refer to the data in a collection of closely related tables, corresponding to a particular experiment or event. An example of this type is the data sets collected by space agencies performing experiments with instruments aboard space probes.

In our project we use a database for both eye and fingerprint, for eye the database is stored in the raspberry pi and it can be refreshed from time to time. For fingerprint the database is stored in the device itself and can be changed when required.

4.8 OVERALL PROCESS OF THE PROJECT

The overall process of the project is to identify and authenticate the eye and fingerprint and turn the gear motor on and off based on the case of match or mismatch. The major action to be performed here is to correctly match the eye to the one present in the database; the other is matching the fingerprint to its database.

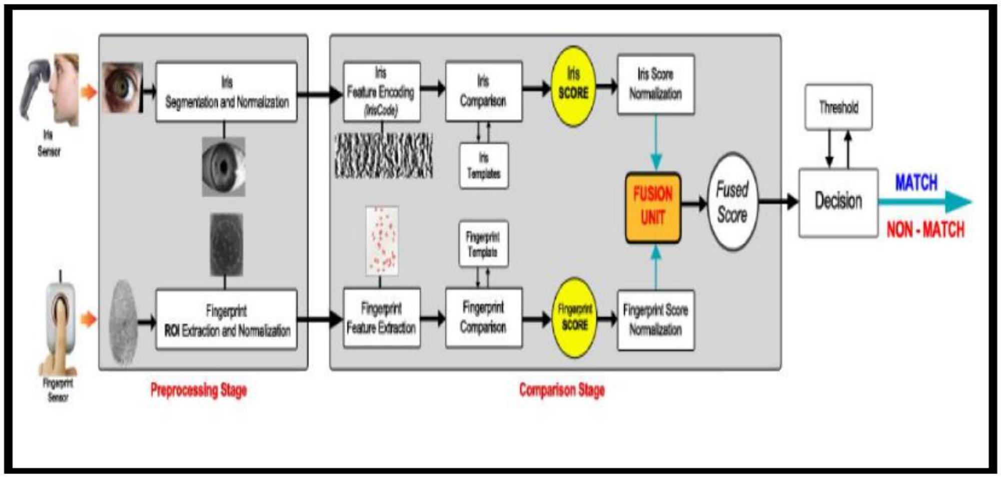


Fig 4.3: Overall process of project

The system basically crosschecks the databases with the data given by the input devices that is the pi camera and the fingerprint scanner and authenticates whether the user is valid or not. Fig 4.3 shows the above described process.

CHAPTER 5

SYSTEM DESIGN

Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

This chapter also illustrates the constraints under consideration and the parameters to be checked in order to select the most appropriate approach to follow.

A Data FLOW diagram (DFD) is a graphical representation of the “flow” of data through an information system modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system which can be later be elaborated. DFDs can also be used for the visualization of data processing (structural design).

5.1 DESIGN CONSTRAINTS

The constraint(s) under consideration are:

- Care should be taken that there is nobody else in the line of sight of the pi camera when identification is happening but the user.
- The eyes of the user must be brightly visible so as the program to be able to validate the user.

5.2 ARCHITECTURAL DESIGN

Biometric technologies vary in capability, performance, and reliability. The success of a given biometric modality depends not only on the effectiveness of the technology and its implementation, but also on the total security solution for which any biometric system comprises only a part.

Because of their security, speed, efficiency, and convenience, biometric authentication systems have the potential to become the new standard for access control. Biometrics replaces or supplements knowledge and possession authentication with a person's physical or behavioral characteristics. Biometrics can be used in any situation where identity badges, PINs/passwords, or keys are needed. Biometrics offers some clear advantages over traditional identity methods:

- Biometric traits cannot be lost, stolen, or borrowed.
- Generally, physical human characteristics are much more difficult to forge than security codes, passwords, badges, or even some encryption keys.

We propose a new multi-modal biometric authentication approach using eye and fingerprint images as biometric traits. We first validate a user based on the eye identification. Then, if the eye has been validated, we proceed to validate the fingerprint of the same user. If both the traits are validated successfully, the user is granted access. Else the user is termed as an "invalid user" and the process stops at that point.

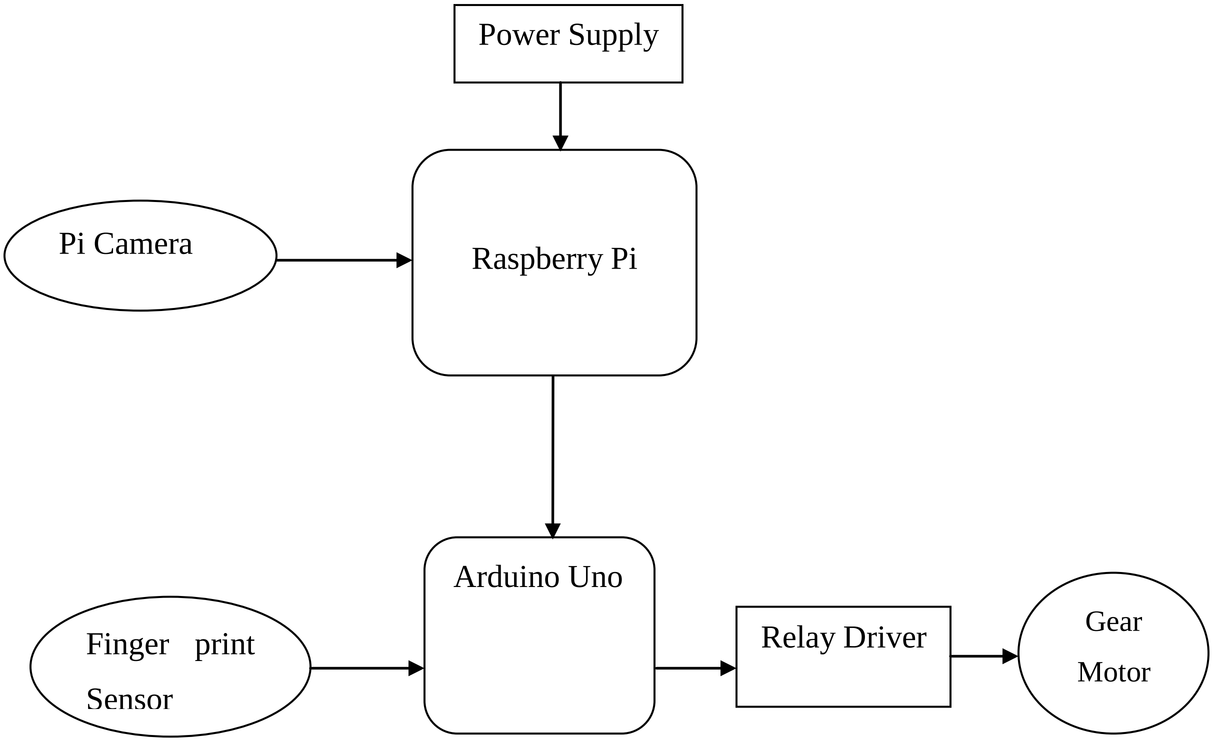


Fig 5.1: System Architecture

As shown in the Fig 5.1, the power supply is provided to the Raspberry Pi, which in turn powers up the Arduino uno device. The Pi Camera is connected to the Raspberry Pi. The Raspberry Pi also in turn houses the Relay, which provides more power for the gear motor. The fingerprint sensor is connected to the Arduino board. The Raspberry Pi and Arduino Uno are connected via high pins.

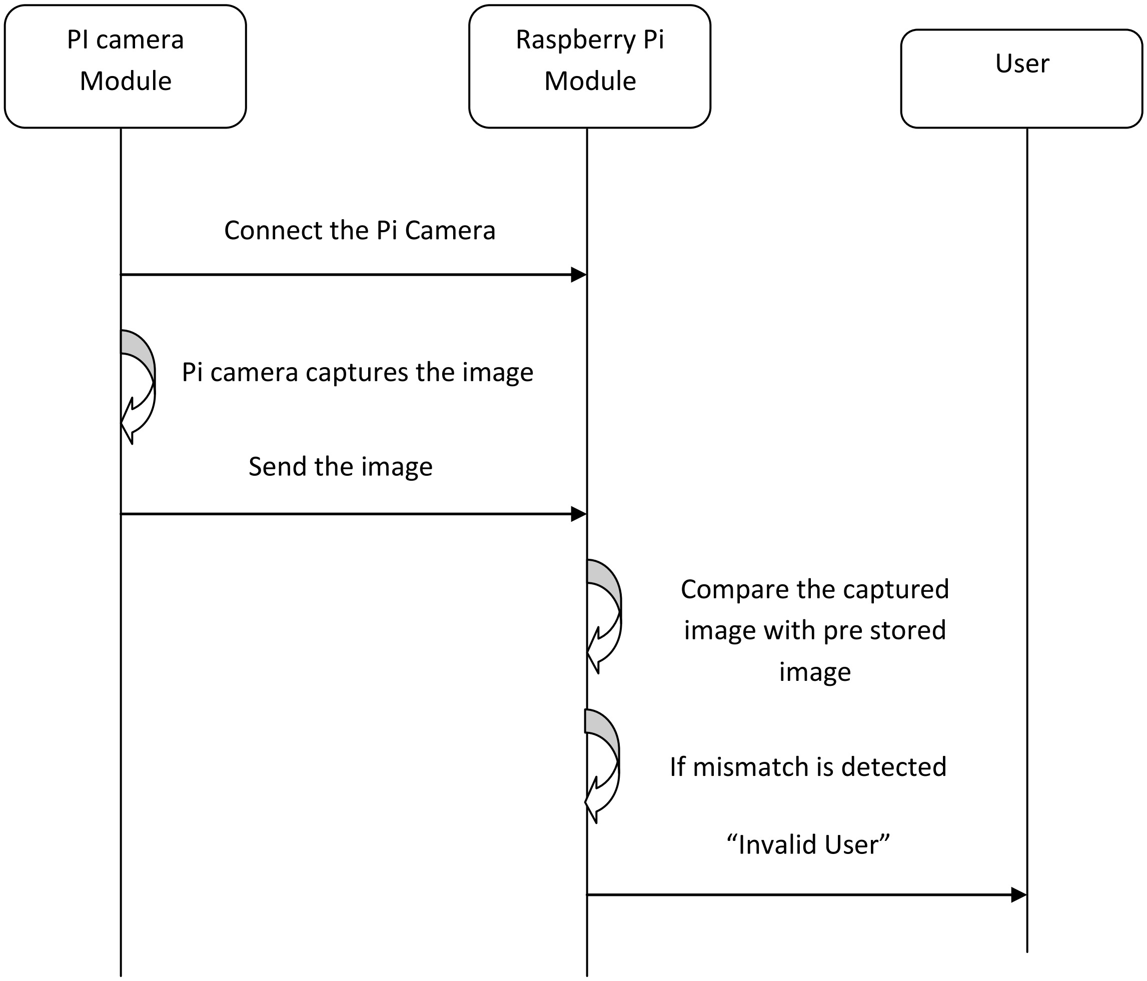


Fig 5.2: Pi Camera Module Sequence Diagram

The figure 5.2 shows the sequence diagram for Pi Camera module. The Pi Camera is connected to the Raspberry Pi. The Pi Camera captures the image of the user and compares it with the pre stored images. If no mismatch is detected, the process proceeds to validate the fingerprint of the user. Else the output “Invalid User” is displayed.

5.3 DATA FLOW DIAGRAM

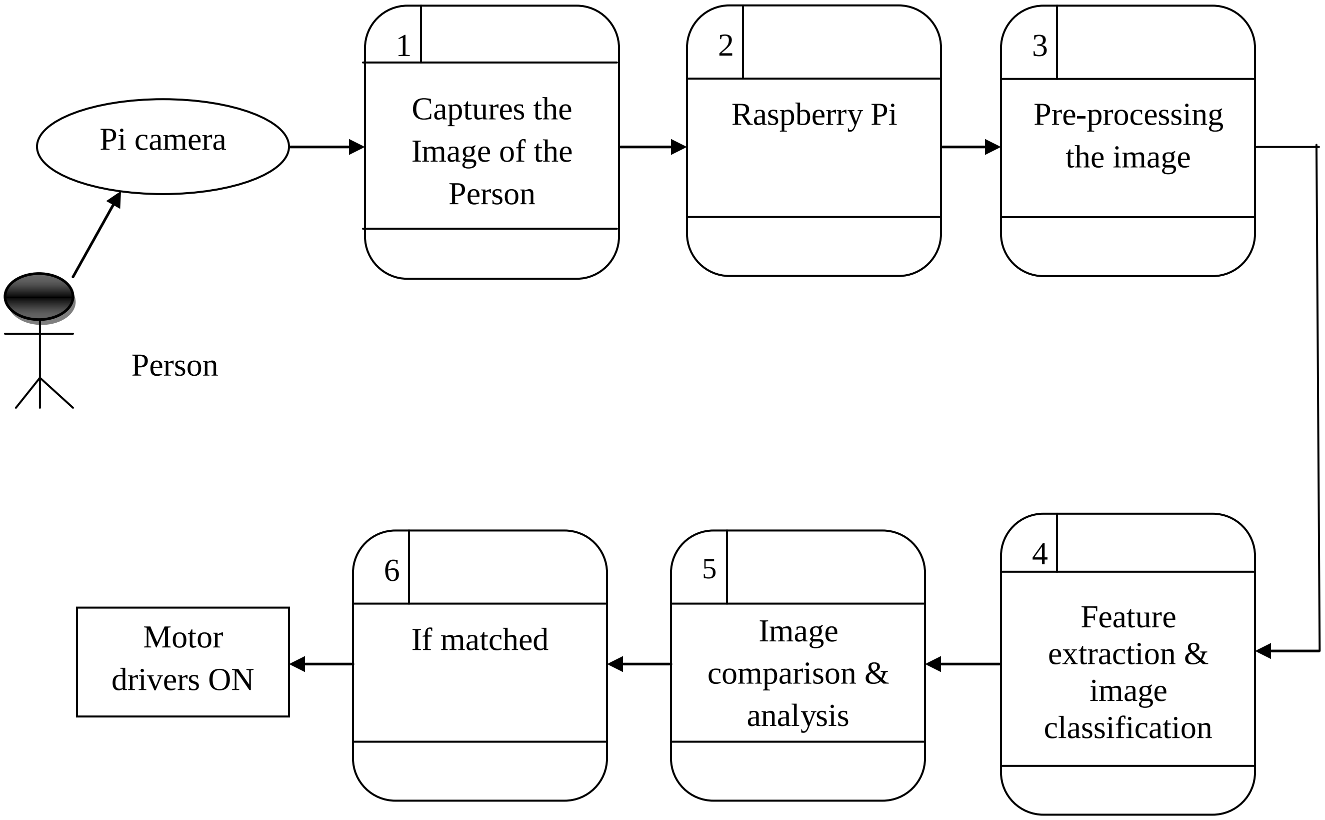

Pi camera
1
Captures the Image of the Person
2
Raspberry Pi
3
Pre-processing the image
Person
4
Feature extraction & image classification
5
Image comparison & analysis
6
If matched
Motor drivers ON

Fig 5.3: Data Flow Diagram

Fig 5.3 shows the data flow diagram of our project involving 6 steps, an image is captured from Pi camera and input is sent to Raspberry Pi. In Raspberry Pi the grayscale and feature extraction of image takes place, then the comparison of image takes place from stored database, if matched then the motor turns ON else the “invalid user” is displayed on screen.

5.4 USE CASE DIAGRAM

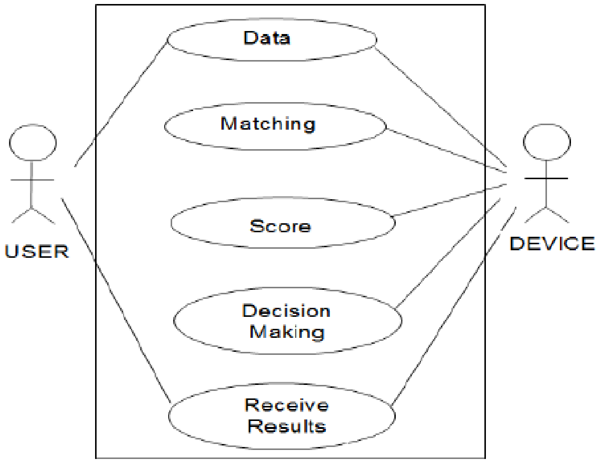


Fig 5.4: Use Case Diagram

From Fig 5.4 we come to know about the use case diagram where user has many operations conducted on the system. The results are also seen.

CHAPTER 6

IMPLEMENTATION

The implementation phase of any project development is the most important phase as it yields the final solution, which solves the problem at hand. The implementation phase involves the actual materialization of the ideas which are expressed in the analysis document and developed in the design phase. Implementation should be perfect mapping of the design document in suitable programming language in order to achieve the necessary final product.

Implementation of software is always preceded by important decisions such as selection of platform, programming language used, etc. These decisions are often influenced by several factors such as real environment in which the system works, the speed that is required, the security concerns and other implementation specific details.

The major parts of implementation in this project are:

- Eye Matching.
- Fingerprint Matching.
- Final Result.

6.1 EYE MATCHING

The eye matching part is implemented in Python language as shown ahead. The code ahead validates the eye for four registered users. Pi Camera is used for capturing image here.

```
#!/usr/bin/python

# Import the required modules

import cv2, os

import picamera
import numpy as np

from PIL import Image

import RPi.GPIO as GPIO

from PIL import Image, ImageEnhance

import requests

#from ser import dat,dat1,dat2,dat3

import time

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(3,GPIO.OUT)

GPIO.setup(11,GPIO.OUT)

GPIO.setup(12,GPIO.OUT)

GPIO.setup(13,GPIO.OUT)

GPIO.setup(15,GPIO.OUT)

GPIO.setup(16,GPIO.OUT)

GPIO.setup(18,GPIO.OUT)

GPIO.output(11,False)

#i=GPIO.output(3)

# For face detection we will use the Haar Cascade provided by OpenCV.

cascadePath = "haarcascade_frontalface_default.xml"

faceCascade = cv2.CascadeClassifier(cascadePath)
```

```
cascadePath1= "haarcascade_righteye_2splits.xml"
faceCascade1 = cv2.CascadeClassifier(cascadePath1)
# For face recognition we will the the LBPH Face Recognizer
recognizer = cv2.createLBPHFaceRecognizer()
def get_images_and_labels(path):
# Append all the absolute image paths in a list image_paths
# We will not read the image with the .sad extension in the training set
# Rather, we will use them to test our accuracy of the training
image_paths = [os.path.join(path, f) for f in os.listdir(path) if not f.endswith('.')]
# images will contains face images
images = []
# labels will contains the label that is assigned to the image
labels = []
var = 0
i=1
for image_path in image_paths:
# Read the image and convert to grayscale
image_path=('/home/pi/db/picture%s.jpg' % i)
i=i+1
#print(image_path)
image_pil = cv2.imread(image_path)
```

```
gray = cv2.cvtColor(image_pil,cv2.COLOR_BGR2GRAY)

# Convert the image format into numpy array

# image_pil = ImageEnhance.Brightness(image_pil)

image = np.array(gray, 'uint8')

#image = np.array(image_pil)

img=cv2.resize(image,(128,128))

# Get the label of the image

#nbr = int(os.path.split(image_path)[1].split(".")[0].replace("subject", ""))


# Detect the face in the image

faces = faceCascade.detectMultiScale(img)

var=var+1


# If face is detected, append the face to images and the label to labels

for (x, y, w, h) in faces:

images.append(img[y:y+h,x:x+w])

labels.append(var)

#cv2.imshow("Adding faces to traning set...", img[y:y+h,x:x+w])

#cv2.waitKey(50)

# return the images list and labels list
```

```
#print(images)

return images, labels



# Path to the Yale Dataset

path = './db'

images, labels = get_images_and_labels(path)

cv2.destroyAllWindows()

#print(labels)

#print(np.array(labels))

recognizer.train(images,np.array(labels))

while True:

print('starting....')

m=1

predicted=0

conf=0.0

#i=GPIO.output(3)

if m==1:

with picamera.PiCamera() as cam:

cam.start_preview()

time.sleep(5)

cam.capture('image.jpg')
cam.stop_preview()

predict_image_pil = cv2.imread('image.jpg')

gray = cv2.cvtColor(predict_image_pil,cv2.COLOR_BGR2GRAY)

predict_image = np.array(gray, 'uint8')

#predict_image = np.array(predict_image_pil)

predict_image =cv2.resize(predict_image,(128,128))

# predict_image = ImageEnhance.Sharpness(predict_image)

faces = faceCascade.detectMultiScale(predict_image)

for (x, y, w, h) in faces:

predicted,conf= recognizer.predict(predict_image[y:y+h,x:x+w])

#cv2.imshow("Adding faces to traning set...", predict_image[y:y+h,x:x+w])

#cv2.waitKey(50)

print(predicted, conf)



if(predicted>=1 and predicted<=10):

print('ID1')

print('User-1 EYE Validated')

time.sleep(2)

GPIO.output(11,True)

print('put your finger')
```

```
break


if (predicted>=11 and predicted<=20):
print('ID2')
print('User-2 EYE Validated')
time.sleep(2)
print(put your finger')
time.sleep(10)
GPIO.output(12,True)
break


if (predicted>=21 and predicted<=30):
print('ID3')
print('User-3 EYE Validated')
time.sleep(2)
print(‘put your finger')
time.sleep(10)
GPIO.output(13,True)
break


if (predicted>=31 and predicted<=40):
```

```
print('ID4')
print('User-4 EYE Validated')
time.sleep(2)
print(‘put your finger')
time.sleep(10)
GPIO.output(15,True)
break

if(predicted==0):
print('Invalid User')
GPIO.output(3,False)
time.sleep(10)
break
```

6.2 FINGERPRINT MATCHING

The device r307 is used for fingerprint capturing and validation. It is connected to Arduino board. The code is written in ino language.

```
#include <Adafruit_Fingerprint.h>
#include <SoftwareSerial.h>
int getFingerprintIDez();
SoftwareSerial mySerial(2, 3);// tx, rx
int in= 0;
const int mot =  13;
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
 void doorOpen()
{

  if(finger.fingerID==0)
  {
  in= digitalRead(4);

  Serial.println("Welcome aaaaa");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line
to display corresponding name
  delay(1000);
  if(in)
  {
      digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to
display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }


  }
  if(finger.fingerID==1)
  {
     in= digitalRead(5);
    Serial.println("Welcome bbbbbb");// i enroled ID no 1 as Cinla's fingerprint, so used this
line to display corresponding name
  delay(1000);
   if(in)
  {
     digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }
  }

   if(finger.fingerID==2)
  {
     in= digitalRead(6);
  Serial.println("Welcome ccccccc");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
   if(in)
  {
    digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }
  }



 if(finger.fingerID==3)
  {
     in= digitalRead(7);
  Serial.println("Welcome dddddddd");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
   if(in)
  {
    digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }
  }




 if(finger.fingerID==4)
  {
     in= digitalRead(8);
  Serial.println("Welcome EEEEEE");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
   if(in)
  {
    digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }
  }

 if(finger.fingerID==5)
  {
     in= digitalRead(9);
  Serial.println("Welcome EEEEEE");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
   if(in)
  {
    digitalWrite(mot, HIGH);
    Serial.println("aaa vv");//i enroled ID no 1 as Nidhi'sfingerprint, so used this line to display corresponding name
  delay(1000);
  digitalWrite(mot, LOW);
  }
  }

}

void doorClose()
{
}
void setup()
{
 pinMode(4, INPUT);
 pinMode(5, INPUT);
 pinMode(6, INPUT);
 pinMode(7, INPUT);
 pinMode(8, INPUT);
 pinMode(9, INPUT);
  Serial.begin(9600);
  Serial.println("fingertest");
  finger.begin(57600);
  pinMode(mot, OUTPUT);
  if (finger.verifyPassword())
  {
    Serial.println("Found fingerprint sensor!");
  } else {
    Serial.println("Did not find fingerprint sensor :(");

    while (1);
  }
  Serial.println("No valid finger found,waiting for valid finger...");

  }

  void loop()                     // run over and over again
{

  if(getFingerprintIDez()>=0)
  {
        doorOpen();
  }

}
uint8_t getFingerprintID()
{
  uint8_t p = finger.getImage();
  switch (p)
  {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
        default:
      Serial.println("Unknown error");
      return p;
  }

  // OK success!

  p = finger.image2Tz();
  switch (p)
  {
    case FINGERPRINT_OK:
      Serial.println("Image converted");
      break;
    case FINGERPRINT_IMAGEMESS:
      Serial.println("Image too messy");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_FEATUREFAIL:
      Serial.println("Could not find fingerprint features");
      return p;
    case FINGERPRINT_INVALIDIMAGE:
      Serial.println("Could not find fingerprint features");
      return p;
    default:
      Serial.println("Unknown error");
      return p;
  }

  // OK converted!
  p = finger.fingerFastSearch();
  if (p == FINGERPRINT_OK)
  {
    Serial.println("Found a print match!");
  } else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
  } else if (p == FINGERPRINT_NOTFOUND) {
    Serial.println("Did not find a match");
    return p;
  } else {
    Serial.println("Unknown error");
    return p;
  }

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
}

// returns -1 if failed, otherwise returns ID #
int getFingerprintIDez() {
  uint8_t p = finger.getImage();
  if (p != FINGERPRINT_OK)  return -1;
  p = finger.image2Tz();
  if (p != FINGERPRINT_OK)  return -1;

  p = finger.fingerFastSearch();
  if (p != FINGERPRINT_OK)  return -1;

  // found a match!
  Serial.print("Found ID #"); Serial.print(finger.fingerID);
  Serial.print(" with confidence of "); Serial.println(finger.confidence);
  return finger.fingerID;
}
```

6.3 FINAL RESULT

The final result is represented through the motor. If both the traits were successfully matched, the motor runs, thus depiction that the user has been given access. If there has been a mismatch, the motor does not run. The Raspberry Pi is connected to Arduino board through the voltage and I/O pins where the output of the Raspberry Pi acts as an input for Arduino board. And similarly the output of Arduino acts as an input for the gear motor.

CHAPTER 7

TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to the process of executing a program or application with the intent of finding software bugs.

Testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not. This activity results in the actual, expected and difference between their results i.e..testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to actual device or requirements.

This chapter introduces the testing environment provided by individual and integrated testing of the modules mentioned in the previous chapters.

7.1UNIT TESTING

Unit testing is an approach successfully used to develop software, and to ease code refactoring for keeping bugs to minimum. It is also the insurance that the software is doing the right calculation (quality insurance).A test just checks if the code is running and is producing the correct answer/behavior in a given situation.

The unit tests are written to check the working of every module used in the program. This guarantees that the module performs the task which assigned to it .This also makes debugging easier in case of any errors which adds to the readability of the program. The goal of unit testing is to isolate each part of the program and show that individual parts are correct in terms of requirements and functionality.

7.2 INTEGRATION TESTING

The testing of combined parts of an application to determine if they function correctly together is integration testing .This testing can be done by using two different methods:

- TOP-DOWN INTEGRATION TESTING: In Top-Down integration testing, the highest-level modules are tested first and then progressively lower-level modules are tested.
- BOTTOM-UP INTEGRATION TESTING: Testing can be performed starting from smallest and lowest level modules and proceeding one at a time.

This test guarantees that the various modules which are discussed in the previous section and also which pass the unit test are working fine when they are integrated to form a system .The input is given in such a way that all the control paths in the codes are executed are performed effectively.

7.3SYSTEM TESTING

This is the next level in the testing and tests the system as a whole .Once all the components are integrated, the application as a whole is tested rigorously to see that it meets Quality Standards.

ACCEPTED TESTING: The main purpose of this testing is to find whether application meets the intended specifications and satisfies the requirements, we will follow two different methods:

- ALPHA TESTING: This test is the first stage of testing and will be performed amongst the teams. Unit testing, integration testing and system testing are together known as Alpha testing.
- BETA TESTING: In beta testing, a sample of the intended audience tests the application and send their feedback to the project team. Getting the feedback, the project team will fix the problems before releasing the software to actual user.

Following are the few test cases used in our project:

Test case for Both Samples Valid (Valid User):

Test Case	1
Name of Test	Valid Eye Sample and Valid Fingerprint Sample (Valid User).
Input	First, the Eyes of the user are scanned, Once the eye samples are matched, the user is asked for fingerprint sample.
Expected output	The motor should rotate, thus giving access to the user.
Actual output	Once the Eye and Fingerprint are validated, the motor rotates (Since both Eye and Fingerprint are valid).
Result	Successful

Fig 7.1 Test case for Both Samples Valid

Second Sample Invalid

Test Case	2
Name of Test	Valid Eye Sample and Invalid Fingerprint Sample
Input	First, the Eye of the user is scanned, Once the Eye is matched, the user is asked for fingerprint sample.
Expected output	The motor should not rotate (Imposter).
Actual output	The process stops after fingerprint verification and the motor does not rotate (Since only Eye is valid,

	the process stops when fingerprint is found to be invalid.)
Result	Successful

Fig 7.2 Test case for Invalid Fingerprint

First Sample Invalid (Invalid User):

Test Case	3
Name of Test	Invalid Iris Sample (Invalid User)
Input	The Iris of the User is scanned.
Expected output	The motor should not rotate (Invalid Person).
Actual output	The process stops after Iris is found to be invalid and Invalid User is displayed.
Result	Successful

Fig 7.3 Test case for Invalid Eye sample

7.3.1 TESTING METHODS

- WHITE BOX TESTING: White box testing is detailed investigation of internal logic and structure of the code .To perform White box testing on an application, the tester needs to possess knowledge of the internal working of the code .
- BLACK BOX TESTING: The techniques of testing without having any knowledge of the interior workings of the application is Black Box testing. Typically, when performing a black box test, a tester will interact with the system’s user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

CHAPTER 8

INTERPRETATION OF RESULTS

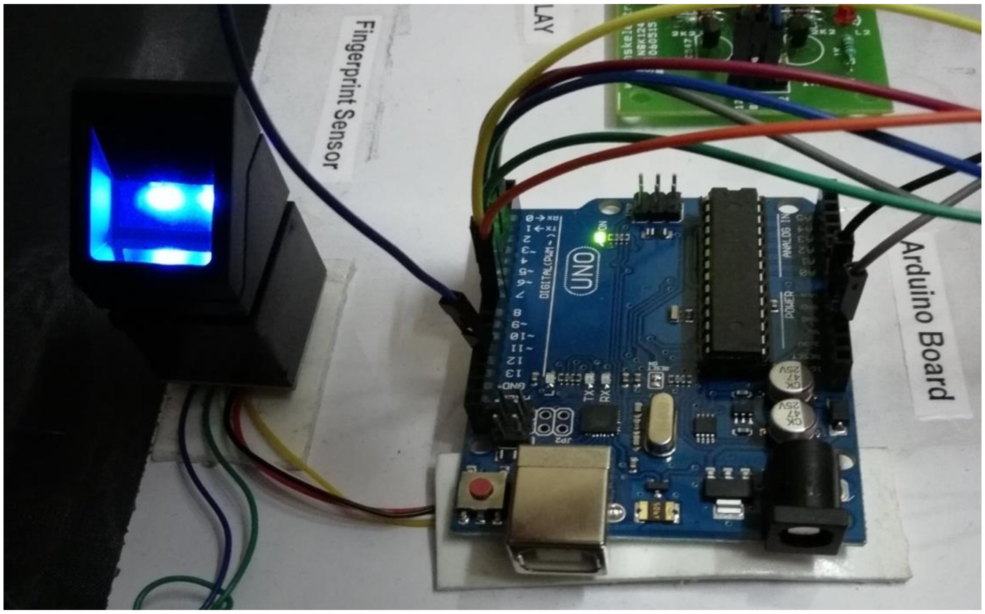


Fig 8.1 R307 Fingerprint Sensor Connected to Arduino board.

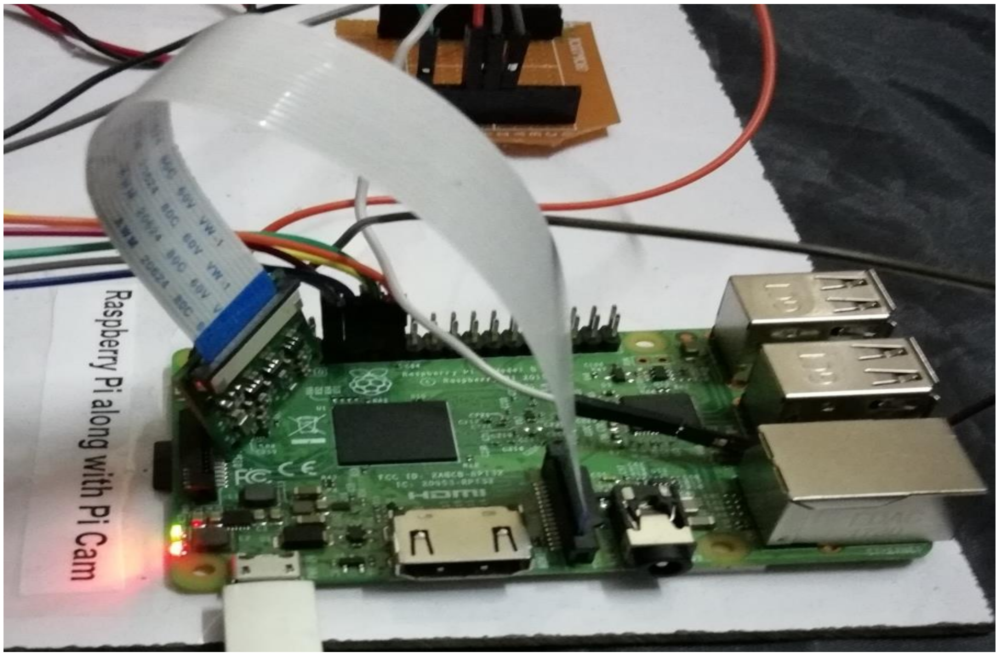


Fig 8.2 Pi Camera Connected to Raspberry Pi.

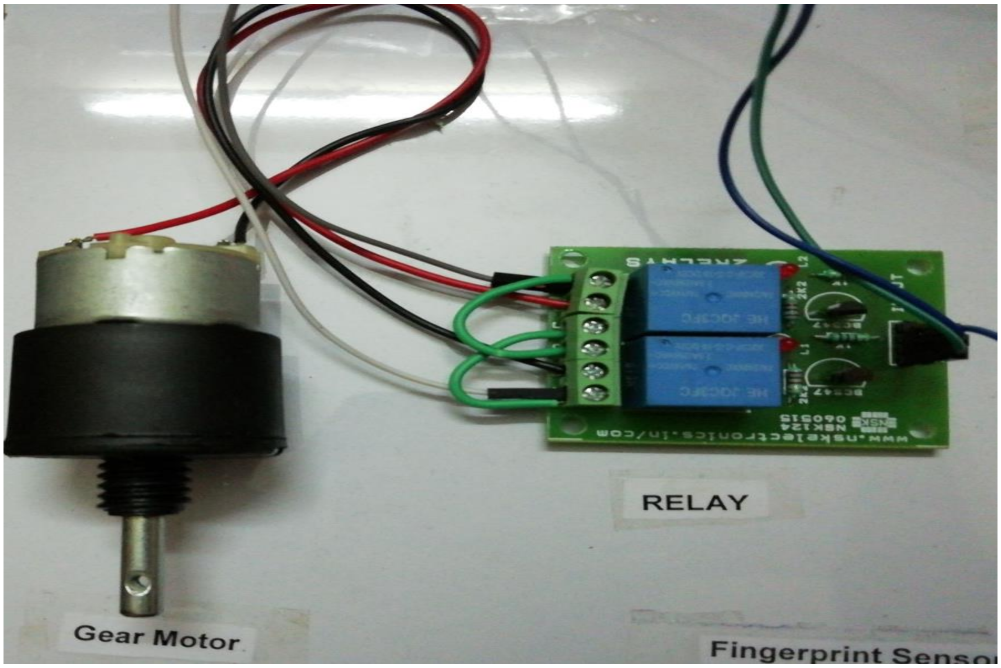


Fig 8.3 Gear Motor connected to Relay.

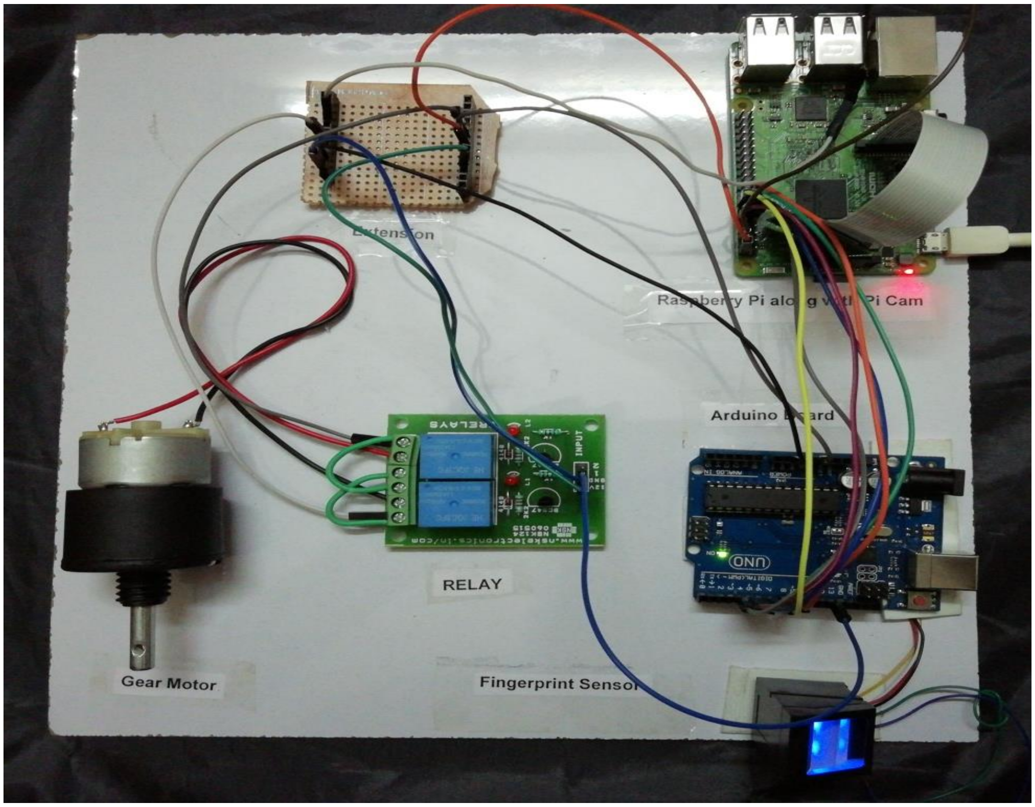
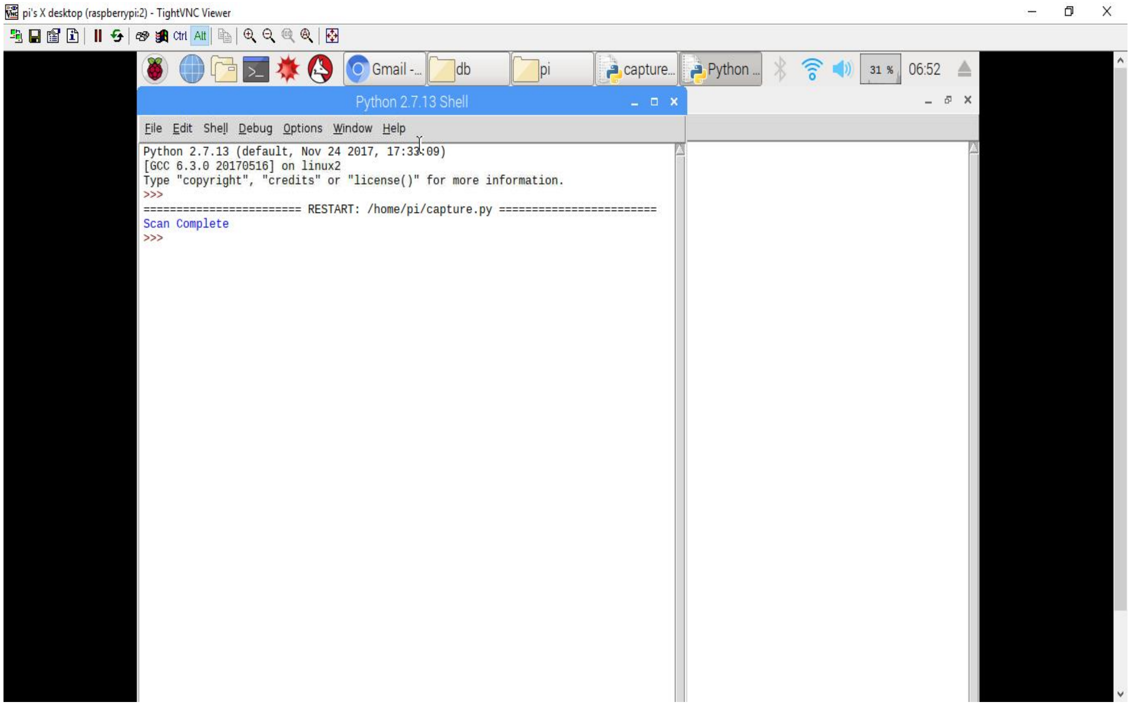
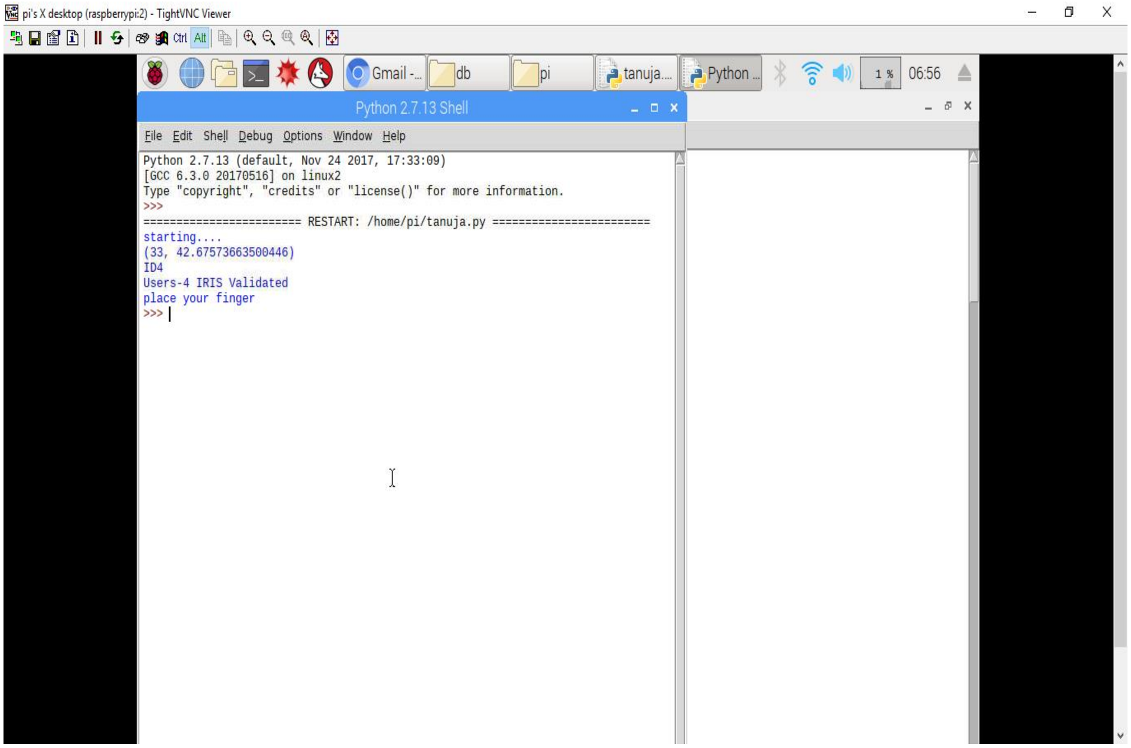


Fig 8.4 System integrated with all the components.


pi's X desktop (raspberrypi:2) - TightVNC Viewer
Ctrl Alt
Gmail -...
db
pi
capture...
Python ...
31 %
06:52
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help

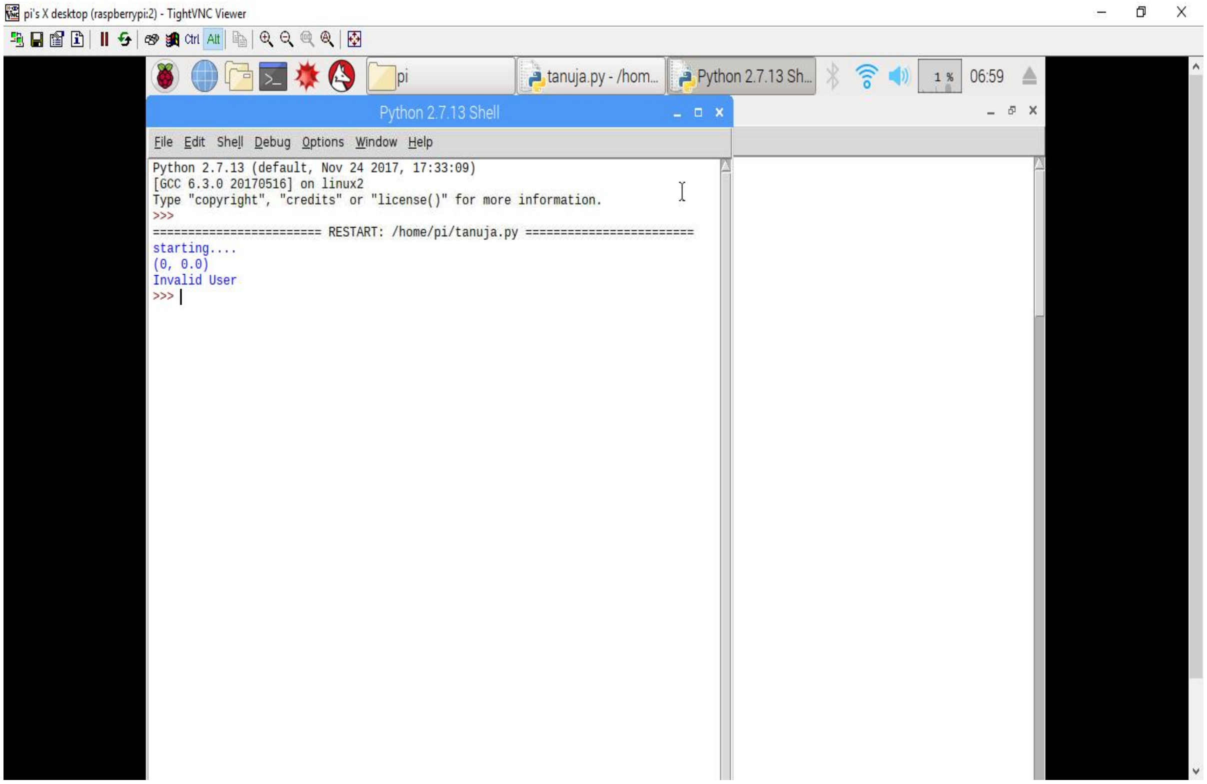
```
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: /home/pi/capture.py =======================
Scan Complete
>>>
```

Fig 8.5 Screen after capturing the database for Eye samples.


pi's X desktop (raspberrypi:2) - TightVNC Viewer
Ctrl Alt
Gmail -...
db
pi
tanuja....
Python ...
1 %
06:56
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help

```
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: /home/pi/tanuja.py =======================
starting....
(33, 42.67573663500446)
ID4
Users-4 IRIS Validated
place your finger
>>> |
```

Fig 8.6 Output screen for a valid user.


pi's X desktop (raspberrypi:2) - TightVNC Viewer
pi
tanuja.py - /hom...
Python 2.7.13 Sh...
1 %
06:59
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help

```
Python 2.7.13 (default, Nov 24 2017, 17:33:09)
[GCC 6.3.0 20170516] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>
======================= RESTART: /home/pi/tanuja.py =======================
starting....
(0, 0.0)
Invalid User
>>> |
```

Fig 8.7 Output screen for an Imposter or Invalid user.

Fig 8.1 shows the fingerprint sensor R307 connected to Arduino board. Fig 8.2 depicts the connection of Pi camera with Raspberry. Fig 8.3 shows the connection between relay and gear motor. Fig 8.4 depicts the overall connection of the system along with its components. Fig 8.5, Fig 8.6 and Fig 8.7 shows the output of screens - when the eye images are captured, valid user output and invalid user output respectively.

CHAPTER 9

CONCLUSION

9.1 TASKS

Through working on this project we have accomplished many Tasks. Some of the tasks are discussed .The embedded system that we have developed can be implemented in Banks, Highly secured Vaults' and is very helpful for ultra high security to avoid imposters and intruders to access the entry .The samples are taken, matched and then the decision is made and is displayed to the user.

9.2 ACHIEVEMENTS

During our project, we were able to achieve many tasks successfully. We were able to take two samples input from the user and then match it against the stored database and then display the results. We could successfully complete the task of validating the user authentication.

9.3 FINAL OUTCOMES

The embedded system that we are using is Raspberry Pi and Arduino board. First, by using Pi camera connected to the Raspberry Pi we capture the eye part of the user, and validate it against the stored datasets .Then, if the validation is successful it goes to validation of Fingerprint otherwise it displays invalid user .During Fingerprint validation, the sample is taken using R307 fingerprint sensor connected to Arduino board ,then it is processed and validated against the datasets stored inside the sensor. If matched then we get the validation message and the motor runs else it is categorized as invalid user.

CHAPTER 10

FUTURE ENHANCEMENTS

The following are few of the enhancements that can be focused on:

- This project can be further enhanced and we can include the liveliness factor during the collection of samples.
- This project is portable, it can be implemented in various fields of security irrespective of their domain and hence it should be made domain specific.
- Our project uses low resolution camera for testing so to get the clear images we can use high clarity camera.
- As this project uses two biometrics traits we can extend this project to include more traits like face, ear, iris, voice, gait etc.
- In the enhanced project we can add a property that the system sends the message to the user whenever only eye part is validated and then if the user clicks ok then the further process occurs.

BIBLIOGRAPHY

[1] Kamer Vishi, S¸ule Yildirim Yayilgan, “Multimodal Biometric Authentication using Fingerprint and eye Recognition in Identity Management”, IEEE Ninth International Conference on Intelligent Information Hiding and Multimedia Signal Processing 2013.

[2] Safaa S. Omran , Maryam A. Salih “Design and Implementation of Multi-Model Biometric Identification System”,International Journal of Computer Applications (0975 – 8887) Volume 99 – No.15, August 2014 .

[3] B.Sabarigiri a and D.Suganyadevi b, “An Efficient Multimodal Biometric Authentication based on IRIS and Electroencephalogram (EEG)”, Proc. of Int. Conf. on Control, Communication and Power Engineering, CCPE, Elsevier, 2014,pg 606-618.

[4] Youssef ELMIR, Zakaria ELBERRICHI, Reda ADJOUDJ,” Score Level Fusion Based Multimodal Biometric Identification (Fingerprint & Voice)”, 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT),pp. 146-150,2012.

[5] Richa Jani, Navneet Agrawal,” A Proposed Framework for Enhancing Security in Fingerprint and Finger-Vein Multimodal Biometric Recognition - IEEE Conference Publication”, International Conference on Machine Intelligence Research and Advancement,pp. 440-444, 2013.

[6] J. Malik, D. Girdhar, R. Dahiya, G. Sainarayanan, "Reference Threshold Calculation for Biometric Authentication", International Journal of Image Graphics and Signal Processing (IJIGSP), vol. 6, pp. 46, 2014.

[7] Sattar B. Sadkhan-SMIEEE, Baheeja K. AL-Shukur, Ali k. Mattar, “Human Voice Extracted Biometric Features: What Can be Used for”, International Conference on Current Research in Computer Science and Information Technology (ICCIT), Slemani – Iraq, pp. 7-12,2017.