

Biometric Authentication System

Submitted in partial fulfilment of the requirements
of the degree of
Bachelor of Technology (ECE)

By

Bishwarup Neogy (114207)

Akshay Meher (114210)

Srikar Chintapalli (114211)

Supervisor:

Sri S.K.L.V. Sai Prakash

Associate Professor



Department of Electronics and Communication Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL

(2015)

Approval Sheet

This Project Work entitled Biometric Authentication System by Bishwarup Neogy , Akshay Meher and Srikar Chintapalli is approved for the degree of Bachelor of Technology

Examiners

Supervisor

Sri S.K.L.V. Sai Prakash

Chairman

Dr. T. Kishore Kumar

Date : _____

Place : _____

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/ data/ fact/ source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Bishwarup Neogy

(114207)

(Signature)

Akshay Meher

(114210)

(Signature)

Srikar Chintapalli

(114211)

Date: _____

Certificate

This is to certify that the project work entitled “*Biometric Authentication System*” is a bonafide record of work carried out by *Bishwarup Neogy (114207)*, *Akshay Meher (114210)* and *Srikar Chintapalli (114211)* submitted to the faculty of “Electronics and Communication Engineering Department”, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in “Electronics and Communication Engineering” at National Institute of Technology, Warangal during the academic year 2014-2015.

Dr. T. Kishore Kumar

Associate Professor & Head
Department of ECE
NIT Warangal

Sri S.K.L.V. Sai Prakash

Associate Professor
Department of ECE
NIT Warangal

Abstract

The objective of this project is to create/build a reliable, efficient, and cost-effective authentication system to be used at ATMs. Current systems of password or pin based authentication that are employed at ATMs are prone to fraud, and customers can forget their means of authentication. The proposed solution to this is to find an affordable yet quick way to authenticate customers at an ATM at any given time with minimal to no room for malpractice. In order to do this, we will interface a fingerprint scanner and a webcam with an affordable and durable microcontroller which can be deployed as a biometric recognition system at the ATM. In order to minimize malpractice, using biometric traits that are unique to individuals is the best proposed plan of action. Several features of fingerprints make them an ideal biometric trait to use, justifying their ubiquitous presence all over the globe. We can further add another tier to the system by incorporating facial recognition.

The proposed plan is to capture a fingerprint from each customer and store them on a host computer's memory. Any incoming prints will be compared with those on memory, and that customer will either be forwarded to the next tier of security or rejected. The next step of the process is to run a one-to-one facial recognition algorithm on the person who's fingerprint was detected. If his/her face matches with the one associated on file, then the person is authenticated to make any transactions.

Table of Contents

1. Introduction	1
2. Review of Literature	
2.1 Face Recognition	
2.1.1 Detection and Localization Techniques	4
2.1.2 Feature Extraction Techniques	5
2.1.3 Recognition Techniques	6
2.2 Fingerprint Recognition	
2.2.1 Pre-Processing	7
2.2.2 Minutiae Extraction Techniques.....	8
3. Hardware and Software Elements	
3.1 Hardware	
3.1.1 Microcontroller	11
3.1.2 Fingerprint Scanner	14
3.1.3 Webcam	15
3.2 Software	
3.2.1 Raspbian OS	15
3.2.2 OpenCV Platform	16
4. Face Recognition Implementation	
4.1 Approach	17
4.2 Algorithm	
4.2.1 Pre-Processing	18
4.2.2 Local Binary Patterns	19
4.2.3 Face Description and Recognition.....	20
4.3 Execution	21
5. Fingerprint Recognition Implementation	
5.1 Approach	22
5.2 Algorithm	
5.2.1 Pre-Processing	23
5.2.2 Binarization and Thinning	25
5.2.3 Minutiae Extraction and Filtering.....	27
5.2.4 Matching	28
5.3 Execution	28

6. Results and Discussions.....	30
7. Summary and Conclusions	34
Literature Cited.....	38

List of Tables

Table 3.2	Comparison of Microcontrollers/Microprocessors	page 13
Table 3.2	Specifications of Futronic FS88 Scanner	page 15

List of Figures

Figure 4.1	Illumination Normalization	page 19
Figure 4.2	Local Binary Pattern Thresholds	page 20
Figure 5.1	Normalization	page 23
Figure 5.2	Gabor Filtering	page 24
Figure 5.3	Binarization	page 25
Figure 5.4	Thinning	page 26
Figure 6.1	System Block Diagram	page 33

Chapter 1

Introduction

Biometrics authentication has long been used because of its ability to uniquely identify people with minimal error and consistently avoid fraudulent identification when compared to traditional methods of identification. Token based systems such as cards and passwords can easily be replicated or compromised, leading to potentially catastrophic consequences.

Several criteria are outlined that can be measures of effectiveness of a biometric trait to be used in a system. Universality means that every person using a system should possess the trait. Uniqueness means the trait should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another. Permanence relates to the manner in which a trait varies over time. More specifically, a trait with 'good' permanence will be reasonably invariant over time with respect to the specific matching algorithm. Measurability relates to the ease of acquisition or measurement of the trait. Performance relates to the accuracy, speed, and robustness of technology used. Acceptability relates to how well individuals in the relevant population accept the technology such that they are willing to have their biometric trait captured and assessed. Circumvention relates to the ease with which a trait might be imitated using an artifact or substitute [6].

These factors are all taken into account when choosing a suitable biometric trait for identification. Some commonly used biometrics used in ID systems includes facial recognition, retina scanning, fingerprint scanning, and voice recognition. These major

“conditions” are all very well satisfied by the fingerprint and face biometrics, and they are also significantly less expensive to process than the other biometric traits.

Biometric authentication has several advantages. First, biometrics can authenticate only people; machines cannot access systems with biometric authentication methods. Biometric characteristics that are used in authentication systems are unique for each person, making it very difficult for criminals to commit fraud. The major advantage of the biometrics is that you have always with you your way to authenticate yourself. For example, you can forget a password or lose an access card, but it is impossible to forget your fingerprint, your face, or your voice. Using biometric is more practical for the user as opposed to remembering several passwords or access codes and cards. It can reduce the cost of password and access-card administration. In most cases, it is more difficult to attack a biometric authentication system than it is to attack an authentication system based on a password or an access card. One can guess a poorly set password or steal an access card. Great lengths have to be taken to fool a good biometric authentication system.

If biometrics were introduced as a form of authentication into transactions, overall security would be increased since we are providing a convenient and low-cost additional tier of security. Fraud is reduced by employing hard-to-forge technologies and materials, and the opportunity for fake identification is minimized. The problems caused by lost IDs or forgotten passwords are also eliminated by using physiological attributes; this prevent unauthorized use of lost, stolen or "borrowed" ID cards. Password administration cost is highly reduced; hard-to-remember passwords are replaced by shared or observed passwords. A wide range of biometric solutions, technologies, customer applications, and databases can be integrated into a robust and scalable control solution for facility and network access. We can unequivocally link an individual to a transaction or event. Transaction monitoring fraud detection systems are now commonplace at financial institutions. As a result of fraudsters evolving and altering their behavior, legacy transaction monitoring systems are known to miss some fraudulent transactions because they are consistent with historical, legitimate transactions. By integrating biometric data into the process, fraud detection rates can be improved significantly. Biometric analysis can assist with the post-fraud transaction verification processes, so calls to fraudsters who confirm their fraudulent activity as "legitimate" are detected. Fraudsters in wire transfer fraud and card fraud schemes are known to have researched the legitimate customer's prior transaction activity in order to make their fraud transactions look as normal and customary as possible. For example, fraudsters are

known to falsely generate wire transfer requests that are only slightly different than previous legitimate wire transfer requests. Similarly, fraudsters are known to mirror prior card activity with "normal" merchant category codes, transaction amounts, and transaction geography on fraudulent card transactions. All of these can be vastly curbed if any customer can be confidently linked physiologically to any transaction that he or she makes.

Fingerprint identification is the process of comparing two instances of friction ridge skin impressions from human fingers to determine whether these impressions could have come from the same individual. The flexibility of friction ridge skin means that no two finger or palm prints are ever exactly alike in every detail; even two impressions recorded immediately after each other from the same hand may be slightly different. Fingerprint identification involves an expert computer system operating under threshold scoring rules, determining whether two friction ridge impressions are likely to have originated from the same finger. Similarly, facial recognition involves the extraction of certain key features on the face such as nose, mouth, eyes, eyebrows, etc. These features are stored based on their distances and orientations from each other, and thresholds are used once again to compare two sets of features. The main challenges in face recognition are to automatically locate the face followed by recognition of the face from a general view point under different illumination conditions, facial expressions, and aging effects.

Chapter 2

Review of Literature

2.1 Face Recognition

In general, an automatic face recognition system comprised of three steps. Among them, detection may include face edge detection, segmentation and localization, namely obtaining a pre-processed intensity face image from an input scene, either simple or cluttered, locating its position and segmenting the image out of the background. Feature extraction may denote the acquirement of the image features from the image such as visual features, statistical pixel features, transform coefficient features, and algebraic features, with emphasis on the algebraic features, which represent the intrinsic attributes of an image. Face recognition may represent to perform the classification to the above image features in terms of a certain criterion. Segmentation among three steps is considered to be trivial, easy and simple for many applications such as mug shots, drivers' licenses, personal ID card, and passport pictures. Thus this problem did not receive much attention. Scholars have given more interest on addressing other problems. However, recently more effort is devoted to the segmentation problem with the advancement of face recognition systems under complex background.

2.1.1 Detection and Localization Techniques

a) Geometrical Approach:

The method is based on face geometrical configuration. Generic knowledge about faces employed is facial organs' position, symmetry, and edge shape as follows: a face contains

four main organs, i.e., eyebrows, eyes, nose and mouth; a face image is symmetric in the left and right directions; eyes are below two eyebrows; nose lies between and below two eyes; lips lie below nose; the contour of a human head can be approximated by an ellipse, and so on [28]. By using the facial components as well as positional relationship between them we can locate the faces easily. When a face image is fed into the system, a preprocessing step will be applied to remove small light details and to enhance the contrast. Then, the processed image will be the threshold to produce a binary image. Finally, a labeling step and a grouping algorithm will be used to group detected features block by block to locate the faces [25].

b) Color/Texture Based Approach:

Color and texture are two important modalities in many images processing tasks, ranging from remote sensing to medical imaging, robot vision, face recognition, etc. By now their analysis methods have been widely utilizing to detect faces for different races, sexes, and ages [26]. Some research results show that human skin colors cluster in a small region only in the GRB color space instead of the HIS color space; human skin colors differ more in brightness than in colors; and every texture is distinctive and distinguishable from one another. Therefore, the normalized GRB or texture models are considered to be capable of characterizing human face with less variance in color or texture [31].

c) Eigen-face Based Method:

This method approximates the multi-template T by a low-dimensional linear subspace F , usually called the face space. Images are initially classified as potential members of T , if their distance from F is smaller than a certain threshold. The images which pass this test are projected on F and these projections are compared to those in the training set [36].

2.1.2 Feature Extraction Techniques

a) Knowledge Based Method:

This approach depends on generic visual and statistical knowledge to extract features. So far there has been much literature concerning the problem of extracting these features. In facial features are extracted based on generic knowledge of facial components [23].

b) Mathematical Transform Methods:

Some well-known mathematical transforms used to extract features of an image are:

- Fourier transform
- Hadamard transform

- Karhunen-Loeve transform
- Singular-Value Decomposition
- Foley-Sammon transform, etc.

New algorithms based on KLT have been proposed to overcome edge problems due to illumination variation and pose change. DCT algorithm was exploited to develop normalization techniques to improve system robustness.

c) Neural Networks/ Fuzzy Extractor Method

Fuzzy extractors have been recently proposed to be relatively more error-tolerant cryptographic primitives that are potentially useful to protect biometric templates [33]. The data obtained from both verification and enrollment, are required to in the same feature depiction by such extractors. Fuzzy extractors are only concerned about the strength of the secret key extracted, and does not directly assure that privacy is protected [34].

2.1.3 Recognition Techniques

a) Statistical Approach:

In this approach quantitative description of faces is characteristic, elementary numerical description features are used. The set of all possible patterns forms the pattern or feature space. The classes form clusters in the feature space, which can be separated by discrimination hyper-surface. The approach chiefly embraces geometrical parameterization method, eigen-face method, Fisherface method, evolutionary pursuit algorithm, etc. [30].

b) Feature Matching Approach:

This method stores feature points detected using the Gabor wavelet decomposition or multi-scale morphological dilation-erosion into data files for each image. Its identification process utilizes the information present in a topological graphic representation of the feature points.

c) Neural Networks Approach:

Neural networks can be viewed as massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections. Neural networks attempt to use some organizational principles (such as learning, generalization, adaptability, fault tolerance and distributed representation, and computation) in a network

of weighted directed graphs in which the nodes are artificial neurons and directed edges (with weights) are connections between neuron outputs and neuron inputs. The main characteristics of neural networks are that they have the ability to learn complex nonlinear input-output relationships, use sequential training procedures, and adapt themselves to the data [35].

2.2 Fingerprint Recognition

Fingerprint recognition, like facial recognition is an area that is constantly being researched to make the process more streamlined. Since scanned fingerprints come from scanners of different qualities, a procedure must be adopted to make all fingerprints comparable. This procedure is the first step to fingerprint recognition, and it is called pre-processing. Pre-processing further consists of three main steps which include normalization, ridge frequency estimation, ridge orientation estimation, and filtering. After the fingerprint is preprocessed, the next and most important step is feature extraction. In this case, the features that need to be extracted are called minutiae, which are local ridge patterns. These patterns and their positioning are what make a person's fingerprint unique. There are several minutiae extraction methods that are currently being employed in fingerprint recognition systems, all of which have their own merits and demerits. After the minutiae have been extracted from a fingerprint, the last step is to match them with another set of minutiae. Minutiae and the defining ridge features of a fingerprint are used to uniquely identify a person by comparing against an existing database of prints. However, there are several problems that may arise, one of which is low image quality. Incorrect minutiae extraction can lead to errors in identification, which is, of course, undesirable. This corruption may occur due to variations in skin and impression conditions such as scars, humidity, dirt, and non-uniform contact with the fingerprint capture device [9]. This corruption is the main reason that increasingly more research is being done on making the extraction algorithms more efficient and accurate.

2.2.1 Pre-Processing

The most widely cited fingerprint enhancement techniques is based on the convolution of the image with Gabor filters tuned to the local ridge orientation and ridge frequency. The main stages of this algorithm include normalization, ridge orientation estimation, ridge frequency estimation, and filtering [11][13].

Step one of the pre-processing procedure usually involves normalization of the fingerprint image so that it has a particular mean and variance. Because of distortions due to the fingerprint image capture process such as non-uniform ink intensity or non-uniform contact with the fingerprint capture device, a fingerprint image may exhibit irregular levels of variation in grey-level values along the ridges and valleys. Normalization is employed to help smoothen these irregularities and facilitate further steps in the pre-processing procedure.

An orientation image is calculated by determining a matrix of direction vectors representing the ridge orientation at each location in the image. After the image is partitioned into square blocks, the gradient is calculated for every pixel, and the orientation vector of each block can be found by averaging all the vectors orthogonal to the gradient pixels in the block [18]. Because of noise and other sources of image corruption, the orientation image may not be correctly determined. Since ridge orientation varies slowly in a local area, the acquired orientation image is then smoothed using a low-pass filter to reduce the effect of outliers [14].

The third step in the image enhancement process is the estimation of the ridge frequency image. This image outlines the local frequency of the ridges contained in the print. Ridge frequency in a fingerprint can be defined as the average distance between the ridges in the print.

Gabor filters are band-pass filters that are the main constituent of the image enhancement process as a whole. They have frequency-selective as well as orientation-selective properties, allowing them to be tuned to specific frequency and orientation values/requirements [20]. Since fingerprints are known to have well defined local ridge orientation and frequency, the enhancement algorithm takes advantage of this regularity of spatial structure by applying Gabor filters that are tuned to match the local ridge orientation and frequency [14]. The Gabor filter is effectively applied to each pixel and performs the necessary work to reduce noise.

2.2.2 Minutiae Extraction Techniques

a) Binarization

Most of today's employed minutiae extraction techniques involve binarization of the fingerprint image. This means to have only two possible colors: black or white. This makes

the color scale handling of the image much easier. Grey-scale minutiae extraction and fingerprint is actively being researched to save the overhead of binarization.

b) Thinning

The objective of thinning is to find the ridges of one pixel width [21]. The process consists of performing successive erosions until a set of connected lines of unit-width is reached. The line connections form an image that is called the “skeleton of the fingerprint.” The main objective of thinning is to preserve the connectivity and topology. The primary problem in the minutiae extraction method using thinning processes stems from the fact that minutiae in the skeleton image don’t always match up to true minutiae in the fingerprint image. Many spurious minutiae are observed because of undesired spikes, breaks, and holes.

2.2.2.1 Extraction with Unthinned Binarized Images

a) Chain-code based minutiae extraction

Chain-code based minutiae extraction hinges on the fact that the pixel image can be fully recovered from the chain-code of its contour [16]. When using this method, the image is scanned from top to bottom and right to left. The transitions from white to black are detected and the contour is then traced counterclockwise and expressed as an array of contour elements. Each contour element represents a pixel on the contour, which contains the Cartesian coordinates of the pixel, the slope of the contour into the pixel, and other information such as curvature or degree of curvature.

b) Run representation based minutiae extraction

Fingerprint images are represented by a cascade of runs after run-length encoding. The adjacency of the runs is verified and runs with specific characteristics are identified with specific geometric constraints set to detect false minutiae. A Gabor filter is employed for proper binarization of the image because of its effectiveness in reducing noise. In this method, successive black pixels along a “scan line” are defined as a run. Usually, a run-length encoding of a binary image is a list of continual horizontal runs of black-pixels. For each run, the location of the starting pixel of a run and the location of its ending pixel must be recorded [9].

2.2.2.2 Extraction with Thinned Binarized Images

a) Crossing Number minutiae extraction

This extraction technique focuses on creating an array of pixels that surrounds the pixel in question [9][17][21]. Once the image in question is made a skeleton using image thinning, the minutiae are extracted by scanning the local surroundings of each ridge pixel in the image using a 3X3 window.

$$CN = 0.5 \sum_{i=1}^8 | P_i - P_{i+1} | \quad (1)$$

Using this formula we can find differences between adjacent pixel values, and based on this CN value, we can place a meaning onto which pattern is present around a particular pixel. This is the most common form of minutiae extraction.

b) Morphology based minutiae extraction

This extraction procedure consists of essentially comparing specific parts of the fingerprint image with masks that represent different minutiae. Thresholds must obviously be employed to prevent fake detection [11].

Chapter 3

Hardware and Software Elements

3.1 Hardware

3.1.1 Microcontroller

Recent development in mini-computers/microcontrollers has triggered the growth in the development of new cost-effective technologies to enhance tools of everyday usage or those used in hi-tech machinery in industries. This study carries out a comparison between various microprocessors/microcontrollers available in the market for all kinds of users. The idea behind this report is to select one mini-computer among all the others considering certain key parameters, which include, memory, processing ability, portability, compatibility, cost .etc. Upon selection of such a computer, the project aims to implement a potential real-life device that may find its application in one of the following verticals of security, robotics, medical instrumentation, education, communication etc. The main areas of focus will include portability, low power consumption, cost-effectiveness and reliability [4].

Arduino:

The first microcontroller to evaluate for its features is the Arduino. Arduino is a single-board microcontroller, intended to make the application of interactive objects or environments more accessible to the general public. The hardware consists of an open-source hardware board designed around an 8-bit Atmel AVR microcontroller, or a 32-bit Atmel ARM. Current models feature a USB interface, 6 analog input pins, as well as 14 digital I/O pins which allow the user to attach various extension boards. Arduino is used widely used by students and

professionals to interact with the surrounding environment using sensors or actuators. It comes with a simple integrated development environment (IDE) that runs on regular personal computers and allows users to write programs for Arduino using C or C++. The Arduino board exposes most of the microcontroller's I/O pins for use by other circuits. The functionality depends solely on what the user programs and it is a hardware based microcontroller, so many components need to be bought based on the project in question [4].

BeagleBoard Black:

Another, relatively newer item on the market is a microprocessor called the BeagleBoard. It is a low-power open-source hardware single-board computer produced by Texas Instruments in association with Digi-Key and Newark element14. The BeagleBoard was also designed with open source software development in mind, and as a way of demonstrating the Texas Instrument's OMAP3530 system-on-a-chip. The BeagleBoard measures approximately 75 by 75 mm and has all the functionality of a basic computer. The OMAP3530 includes an ARM Cortex-A8 CPU (which can run Linux, FreeBSD, OpenBSD, RISC OS, or Symbian; Android is also being ported. With two 46 pin headers, the BeagleBone Black has a total of 92 possible connection points. The BeagleBone Black is a prime option to use for networking related projects and sensor interfacing [4].

Raspberry Pi:

The fourth and final microprocessor we will evaluate is the Raspberry Pi. The Raspberry Pi is one of the most innovative and affordable products in the technology field today. It is a credit-card sized computer that was founded in 2009 in the UK by the Raspberry Pi Foundation with the intent of promoting computer science in schools. It supports Debian and Arch Linux ARM distributions. The raspberry pi doesn't include an on board solid state drive but, we can use an SD card to do what we need to do. Model A has one USB port and no Ethernet controller, and costs less than the Model B with two USB ports and a 10/100 Ethernet controller. Though the Model A does not have an Ethernet port, it can connect to a network by using an external user-supplied USB Ethernet or Wi-Fi adapter. On the model B the Ethernet port is provided by a built-in USB Ethernet adapter. Like modern computers, generic USB keyboards and mice are compatible with the Raspberry Pi. The Raspberry Pi doesn't come with a real time clock, so an OS must use a network time server or prompt the user on booting for time and date details to timestamp any documents and such. The Broadcom SoC used provides performance comparable to that of a smartphone [4].

Table 3.1 Comparison of Microcontrollers/Microprocessors

	Arduino Uno	Raspberry Pi (Model B)	BeagleBone Black
Processor	ATMega328	ARM11	AM335x
Speed	16 MHz	700 MHz	1 GHz
RAM	2 Kbyte	512 MB	512 MB
USB	NA	2	1
Audio	NA	HDMI/Analog	HDMI
Video	NA	HDMI/Analog	Mini-HDMI
Ethernet	NA	10/100	10/100
I/O	14 GPIO, 6 10 bit analog	8 GPIO	69 GPIO, LCD, GPMC, MMC1, MMC2, 7 AIN, 4 Times, 4 Serial Ports, CAN0
Size	2.95" x 2.1"	3.37" x 2.125"	3.4" x 2.1"
Operating System	NA	Linux	Android, Linux, Windows, Cloud9, CE, etc.
Dev Environmet/Toolkits	Arduino IDE	Linux, IDLE, OpenEmbedded, QUEMU, Scratchbox, Eclipse	Python, Scratch, Linux, Eclipse, Android ADK
Cost	\$30	\$35	\$45

Selection of Microcontroller:

We have decided to proceed with the Raspberry Pi as our processor of choice not only because of its affordability, but mainly because it is tailor fit to our desired applications. It is a software oriented microcomputer and it can be molded to implement several cost efficient hardware projects as well. Pi's huge strength lies in its multimedia interfacing; with high quality video and audio components and Internet capability, it allows for a range of media processing projects as well as internet related work. The Raspberry Pi's HDMI port means it's easy to plug into a TV, and the two USB ports make it so you can operate it like a computer with a mouse and keyboard easily. Last but not least, its Linux based OS provides portability and a new learning experience in terms of beginning microprocessors.

3.1.2 Fingerprint Scanner

FS88 is an advanced optical USB2.0 Fingerprint Scanner from Futronic. It has been certified by FBI to be compliant with PIV-071006 Image Quality Specification for Single Finger Reader, which is a list of specifications and requirements that the biometric scanner must satisfy.

Therefore, FS88 meets the US Federal Information Processing Standard 201(FIPS 201) for Personal Identification Verification (PIV). It is also listed in the US General Services Administration (GSA) FIPS 201 Evaluation Program Approved Product List.

FS88 uses advanced CMOS sensor technology and precise optical system to meet the rigorous requirement on fingerprint image quality of PIV-071006 [2]. Its fingerprint scanning window is crown glass with a thickness of 14mm that resists scratches and other stress to ensure long term heavy duty usage. It is a robust but cost effective single finger capture device and ideal for border control, identity card, driver license, election and any type of civilian AFIS application.

Special electronic circuit is built into FS88 to do Live Finger Detection (LFD). With appropriate software in PC, user can select this LFD feature so that only live finger's fingerprint will be scanned into PC. Fake fingers made from silicone rubber, clay, etc. will be rejected.

Live Finger Detection:

Futronic uses active sensing technology to detect live human finger. In Futronic optical fingerprint scanner, a special signal is emitted to the finger to be authenticated [3]. This signal goes beyond the human skin and then returns to sensor inside the scanner. The returned signal of a live human finger is unique compared to that from any other material. Futronic has developed an algorithm to differentiate the returned signal of live human finger from that of all other material. As a result, Futronic optical fingerprint scanner only captures the fingerprint of live fingers and rejects all other material that puts on it. Fake fingers made from silicone, rubber, clay, etc. cannot be used to get access through Futronic optical fingerprint scanner.

Table 3.2: Specifications of Futronic FS88 Scanner

Scanner Name:	Futronic FS88 Fingerprint Scanner
Manufacturer:	Futronic Technology Co. Ltd.
Connection:	USB 2.0
Supported OS:	Microsoft Windows (32-bit and 64-bit), Linux (32-bit and 64-bit), Mac OS X (x86 32-bit and 64-bit), Android
Resolution:	500 ppi
Image Capture Area:	16 x 24 mm (0.6" x 0.9")
Fingerprint Image Size:	320 X 240 pixels
Sensor Type:	Optical, CMOS
Illumination:	Infrared LEDs
Device Size:	66 x 66 x 29 mm (2.6" x 2.6" x 1.1")
Device weight:	150 grams (0.3 lbs.)
Operating Temperature:	-10°C ~ +55°C

3.1.3 Webcam

The webcam we chose for the system was the Logitech C270 for its relatively high resolution and easy interfacing.

The specs: HD video calling (1280 x 720 pixels) with recommended system, Video capture: Up to 1280 x 720 pixels, Logitech Fluid Crystal™ Technology, Photos: Up to 3.0 megapixels (software enhanced), Built-in mic with noise reduction, Hi-Speed USB 2.0 certified (recommended), and Universal clip fits laptops, LCD or CRT monitors.

3.2 Software

3.2.1 Raspbian OS

Raspbian is a free operating system based on Debian optimized for the Raspberry Pi hardware. An operating system is the set of basic programs and utilities that make the Raspberry Pi run. However, Raspbian provides more than a pure OS: it comes with over 35,000 packages, pre-compiled software bundled in a nice format for easy installation on the Raspberry Pi. Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of

floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi [4].

3.2.2 OpenCV Platform

OpenCV (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision, developed by Intel Russia research center in Nizhny Novgorod, and now supported by Willow Garage and Itseez. It is free for use under the open-source BSD license. The library is cross-platform. It focuses mainly on real-time image processing.

OpenCV's application areas include: 2D and 3D feature toolkits, ego-motion estimation, facial recognition system, gesture recognition, human–computer interaction (HCI), mobile robotics, motion understanding, object identification, segmentation and recognition, motion tracking, and augmented reality among others.

OpenCV is written in C++ and its primary interface is in C++, but it still retains a less comprehensive though extensive older C interface. There are now full interfaces in Python, Java and MATLAB/OCTAVE.

OpenCV runs on Windows, Android, Maemo, FreeBSD, OpenBSD, iOS, BlackBerry 10, Linux and OS X. The user can get official releases from SourceForge or take the current snapshot under SVN from there. OpenCV uses CMake.

Chapter 4

Face Recognition Implementation

4.1 Approach

The implementation of Face Recognition in the Raspberry Pi was carried out using the Local Binary Patterns approach in the OpenCV platform. Here we will focus on descriptors based on Local Binary Patterns (LBP), as they are simple, computationally efficient and have proved to be highly effective features for face recognition. Within LBP-based algorithms, most of the face recognition algorithms using LBP follow the approach proposed by Ahonen et al. In this approach the face image is divided into a grid of small, non-overlapping regions, where a histogram of the LBP for each region is constructed. The similarity of two images is then computed by summing the similarity of histograms from corresponding regions. One drawback of the previous method is that it assumes that a given image region corresponds to the same part of the face in all the faces in the dataset. This is only possible if the face images are fully frontal, scaled, and aligned properly. In addition, while LBP are invariant against monotonic grayscale transformations, they are still affected by illumination changes that induce non monotonic gray-scale changes such as self-shadowing [31].

The LBP operator was originally designed for texture description. The operator assigns a label to every pixel of an image by thresholding the 3x3-neighborhood of each pixel with the center pixel value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. See Figure 1 for an illustration of the basic LBP operator. To be able to deal with textures at different scales, the LBP operator was later extended to use neighborhoods of different sizes [38]. Defining the local neighborhood as a

set of sampling points evenly spaced on a circle centered at the pixel to be labeled allows any radius and number of sampling points. The algorithm for this approach has been explained in the next section in detail.

4.2 Algorithm

We start by summarizing the main common steps of the algorithms used in this work. Then each step is further discussed in detail. The face recognition process consists of four main parts:

- Preprocessing - We begin by applying the Tan and Triggs' illumination normalization algorithm [38] to compensate for illumination variation in the face image. No further preprocessing, such as face alignment, is performed.
- LBP operator application: In the second stage LBP are computed for each pixel, creating a fine scale textural description of the image.
- Local feature extraction: Local features are created by computing histograms of LBP over local image regions.
- Classification - Each face image in test set is classified by comparing it against the face images in the training set. The comparison is performed using the local features obtained in the previous step.

4.2.1 Preprocessing

Illumination accounts for a large part of the variation in appearance of face images. Various preprocessing methods have been created to compensate for this variation [38]. We have chosen to use the method proposed by Tan and Triggs since it is simple, efficient, and has been shown to work well with local binary patterns. The algorithm consists of four steps:

- i. Gamma correction to enhance the dynamic range of dark regions and compress light areas and highlights. We use $\gamma = 0.2$.
- ii. Difference of Gaussians (DoG) filtering that acts as a "band pass", partially suppressing high frequency noise and low frequency illumination variation. For the width of the Gaussian kernels we use $\sigma_0 = 1.0$ and $\sigma_1 = 2.0$.
- iii. Contrast equalization to rescale image intensities in order to standardize intensity variations. The equalization is performed in two steps:

$$I(x, y) \leftarrow \frac{I(x', y')}{(\text{mean}(|I(x', y')|a))^{1/a}} \quad (2)$$

$$I(x, y) \leftarrow \frac{I(x', y')}{(\text{mean}(\min(\tau, |I(x', y')|)a))^{1/a}} \quad (3)$$

where $I(x, y)$ refers to the pixel in position (x, y) of the image I and τ and a are parameters. We use $a = 0.1$ and $\tau = 10$.

- iv. Compress all values into the range $(0, 1)$ with a hyperbolic tangent function:

$$I(x, y) \leftarrow 0.5 \tanh\left(\frac{I(x', y')}{\tau}\right) + 0.5 \quad (4)$$

The values of the parameters γ , σ_0 , σ_1 , a and τ are those suggested by Tan and Triggs. Figure 4.1 illustrates the effects of the illumination compensation.

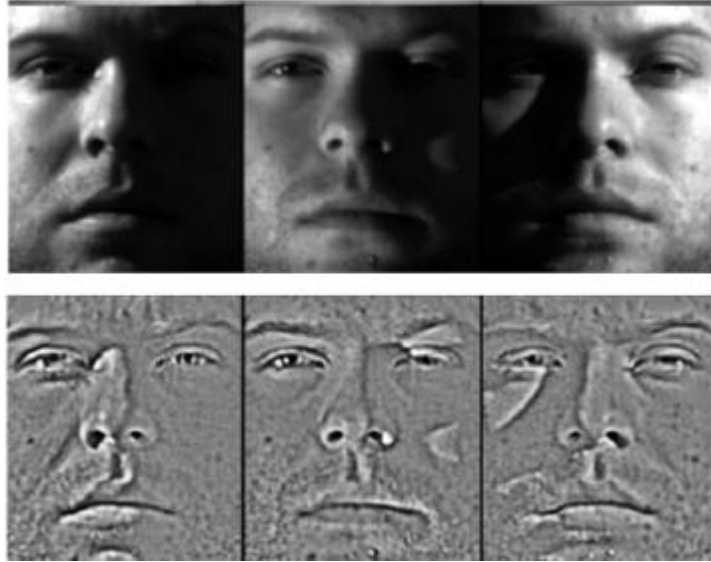


Fig 4.1 The upper row shows three images of a subject dataset under different lighting conditions. The bottom row shows the same images after processing with Tan and Triggs' illumination normalization algorithm.

4.2.2 Local Binary Patterns

Local binary patterns were introduced by Ojala et al [38] as a fine scale texture descriptor. In its simplest form, an LBP description of a pixel is created by thresholding the values of the 3×3 neighborhood of the pixel against the central pixel and interpreting the result as a binary number. The process is illustrated in figure 2. In the LBP operator is generalized by allowing larger neighborhood radii r and different number of sampling points s . These parameters are indicated by the notation $\text{LBP}_{s,r}$. For example, the original LBP operator with radius of 1 pixel and 8 sampling points is $\text{LBP}_{8,1}$. Another important extension is the definition of “uniform

patterns”. An LBP is defined as uniform if it contains at most two 0-1 or 1-0 transitions when viewed as a circular bit string. Thus the 8-bit strings 01100000 and 00000000 are uniform, while 01010000 and 00011010 are not. Ojala observed that when using 8 sampling points, uniform patterns accounted for nearly 90% of the patterns in their image datasets. Therefore, little information is lost by assigning all non-uniform patterns to a single arbitrary number. Since only 58 of the 256 possible 8 bit patterns are uniform, this enables significant space savings when building LBP histograms.

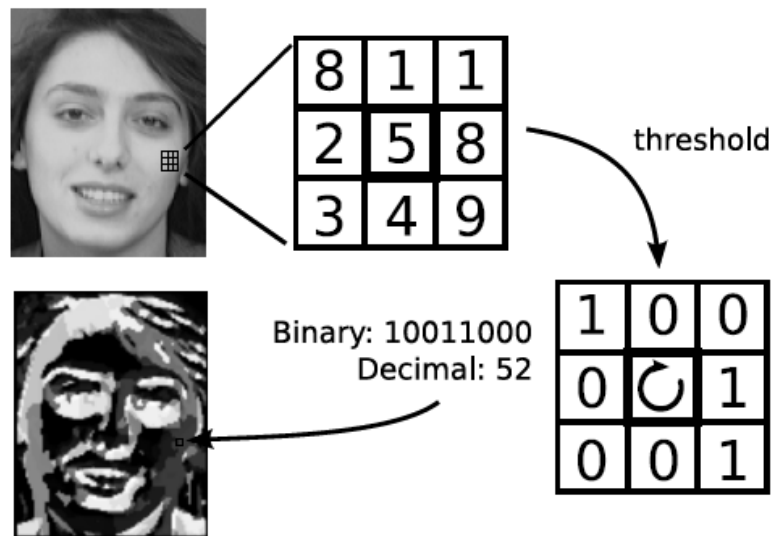


Fig 4.2 The LBP operator thresholds each pixel against its neighboring. In the bottom image, each gray-level value corresponds to a different local binary pattern.

4.2.3 Face Description and Recognition

In order to build the description of a face image we follow the basic methodology proposed by Ahonen [38]. Once the LBP operator is applied to the face image, the face image is divided into regions and a histogram of LBP is computed for each region. The final description of each face is a set of local histograms.

In Ahonen’s system, each face image is partitioned into a grid of non-overlapping square regions. An LBP histogram is computed independently for each region. Then, all the resulting histograms are concatenated together into a large vector. Ahonen et al call this vector a “spatially enhanced histogram”, since the order of histograms that compose it implicitly encode spatial information. This method tends to produce fairly high dimensional vectors. For

example, if an image is divided into an 8 X 8 grid and the $LBP^{u2}_{8;2}$ operator is used (so the histograms have length 59) the spatially enhanced histogram has length 8 X 8 X 59 = 3776. In order to perform face recognition under this scheme, each face image in the training and test sets is converted to a spatially enhanced histogram via the process described above. Then ordinary nearest neighbor classification is performed with a histogram distance measure such as X^2 or histogram intersection [38].

$$X^2(x, y) = \sum_{i=1}^D \frac{(x_i - y_i)^2}{(x_i + y_i)} \quad (5)$$

Where, D is the dimensionality of the spatially enhanced histograms.

4.3 Execution

For the purpose of implementation of the face recognition system we took the help of OpenCV computer vision software installed on top of Raspbian OS in the Raspberry Pi mini-computer.

- As per the requirements of the algorithm, a camera routine was written for the purpose of capturing the candidates' face images. For this we used a Logitech C270, which is a typical USB2.0 enabled webcam and has a reasonable performance in both well-lit and low light condition.
- The next step required the captured images to be cropped, resized, rotated and normalized such that they are all of the same size and contain only the faces of the candidates. Aligning faces is an important preprocessing step that requires to be done as it can have adverse effects on the working of the algorithm.
- Once the images have been aligned, it is necessary to train the face recognizer algorithm. To do so we took the help of some neutral images from the University of Texas at Dallas database. This set was treated as a group of negative images.
- For the recognition section, when the candidates face is detected in the frame, it is compared with the positive (concerned candidate's face database) and the negative database. Only if the probing candidate is valid and is present in the positive database, the recognizer returns an affirmative result.

Chapter 5

Fingerprint Recognition Implementation

5.1 Approach

The implementation of fingerprint recognition on Raspberry Pi was carried out using the SourceAFIS libraries on a host computer and transmitting comparison results over the network to the Pi. In the SourceAFIS libraries, the approach taken for fingerprint template extraction is the NBIS MINDTCT framework and the approach for matching is the Bozorth3 methodology. Before any extraction can occur, pre-processing of the scanned fingerprint image must occur. The pre-processing includes the following steps: normalization, ridge orientation estimation, ridge frequency estimation, filtering, binarization, thinning, and smoothing. These steps will be further broken down in the following part describing the algorithm. The primary part of the recognition process, the minutiae extraction, is done using the Crossing Number algorithm. The most commonly employed method of minutiae extraction is the Crossing Number (CN) concept. This method involves the use of the skeleton image where the ridge flow pattern is eight-connected. The minutiae are extracted by scanning the local neighborhood of each ridge pixel in the image using a 3×3 window. The CN value is then computed for each pixel and is then associated with a particular type of minutiae. After the minutiae are extracted, there needs to be a filtering process that can iron out false minutiae such as holes and breaks. The final step of the approach is the actual matching part, which is the Bozorth3 method. In this method, adjacent minutiae are taken and “edges” are formed in which the distances and orientations are formed using a reference point.

If the distances and orientations of a group of edges (tree) are the same, then a similar cluster is identified. Matching clusters are used to define a recognition score between two fingerprints. However, everything is based on approximation and proper threshold settings, because exact matches are impossible to achieve, even with the same print.

5.2 Algorithm

We start by summarizing the main common steps of the algorithms used in this work. Then each step is further discussed in detail. The face recognition process consists of four main parts:

- Preprocessing: Normalization, ridge orientation estimation, ridge frequency estimation., Gabor Filtering
- Binarization: NIST-like binarization
- Thinning: Zhang-Suen thinning algorithm
- Minutiae Extraction: Morphology Based Algorithm
- Matching: Bozorth3 method using edge distances and orientations

5.2.1 Preprocessing

- (i) The first part of pre-processing involves normalizing the image in order to give it a particular mean and variance. Because of distortions caused by the process of capturing the fingerprint image such as non-uniform contact with the device, a fingerprint image can display irregular levels of variance in grey-levels near the ridges and the valleys. Normalization is employed to help smoothen these irregularities and facilitate further steps in the pre-processing procedure. Histograms are used, some top/bottom pixels are clipped, and normalization is complete [39].

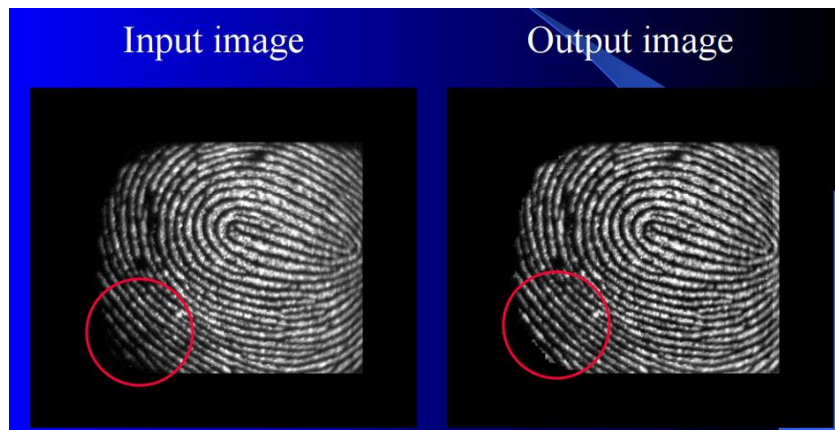


Fig 5.1 Normalization

- (ii) A matrix of direction vectors that represents location-wise ridge orientation helps to calculate an orientation image. After the image is divided into square blocks, each pixel's gradient is calculated, and each block's orientation vector is found by taking the average of all the vectors that are orthogonal to the gradient pixels [18]. Because of noise and other sources of image corruption, the orientation image may not be correctly determined. The effect of outliers is reduced by smoothing the orientation image by way of a low-pass filter [22].
- (iii) The next step to be taken to enhance the image is to estimate the ridge frequency image. This image outlines the local frequency of the ridges contained in the print. After dividing the image into square blocks, an oriented window needs to be calculated for each block. The ridges and valleys of this window are used to construct an x-signature signal. This signal represents the projection of all the oriented window's grey level values along a direction orthogonal to the ridge orientation [14]. A sinusoidal wave shape is formed by the projection in which a ridge's center is mapped as a local minimum and the distance between consecutive peaks is used in ridge frequency estimation.
- (iv) Gabor Filters are employed to just apply precomputed Gabor filter specialized for ridge orientation and frequency.

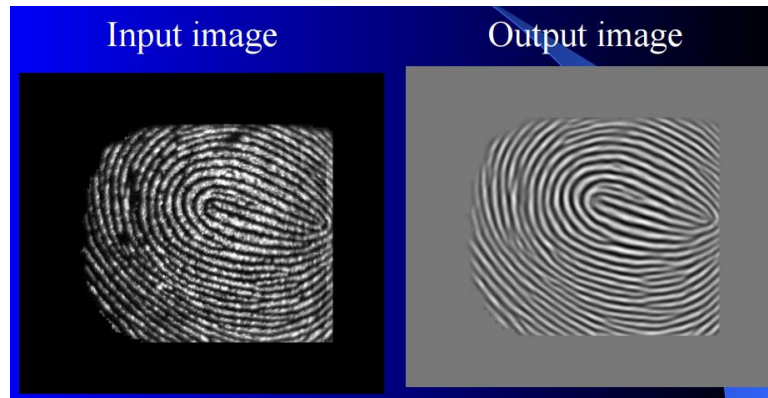


Fig 5.2 Gabor Filtering

5.2.2 Binarization and Thinning

In binarization, the local zero level is computed by averaging pixels along short line orthogonal to ridge orientation. A pixel is then binarized depending on whether its value is above or below this zero level. Median smoothing then takes place to further smoothen the image. The median is computed within circle of about radius 7, optionally adapted to ridge frequency. The circle is sampled with small number of lines crossing its center. Value of the point is determined as a median of sampled pixels. This reliably kills thin lines though. Thin lines are additionally killed in a separate stage designed to do this.

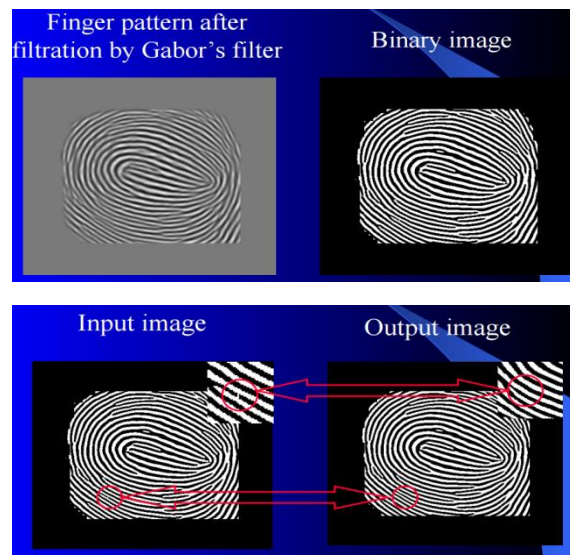


Fig 5.3 Binarization

Thinning takes place usually with the help of the Zhang-Suen thinning algorithm. Assume black pixels are one and white pixels zero, and that the input image is a rectangular N by M array of ones and zeroes.

The algorithm operates on all black pixels P1 that can have eight neighbors. The neighbors are, in order, arranged as:

P9	P2	P3
P8	P1	P4
P7	P6	P5

Obviously the boundary pixels of the image cannot have the full eight neighbors.

- Define $A(P1)$ = the number of transitions from white to black, (0 \rightarrow 1) in the sequence P2,P3,P4,P5,P6,P7,P8,P9,P2. (Note the extra P2 at the end - it is circular).
- Define $B(P1)$ = The number of black pixel neighbors of P1. (= sum(P2 .. P9))

Step 1

All pixels are tested and pixels satisfying all the following conditions (simultaneously) are just noted at this stage.

- The pixel is black and has eight neighbors
- $2 \leq B(P1) \leq 6$
- $A(P1) = 1$
- At least one of P2 and P4 and P6 is white
- At least one of P4 and P6 and P8 is white

After iterating over the image and collecting all the pixels satisfying all step 1 conditions, all these condition satisfying pixels are set to white.

Step 2

All pixels are again tested and pixels satisfying all the following conditions are just noted at this stage.

- The pixel is black and has eight neighbors
- $2 \leq B(P1) \leq 6$
- $A(P1) = 1$
- At least one of P2 and P4 and **P8** is white
- At least one of **P2** and P6 and P8 is white

After iterating over the image and collecting all the pixels satisfying all step 2 conditions, all these condition satisfying pixels are again set to white.

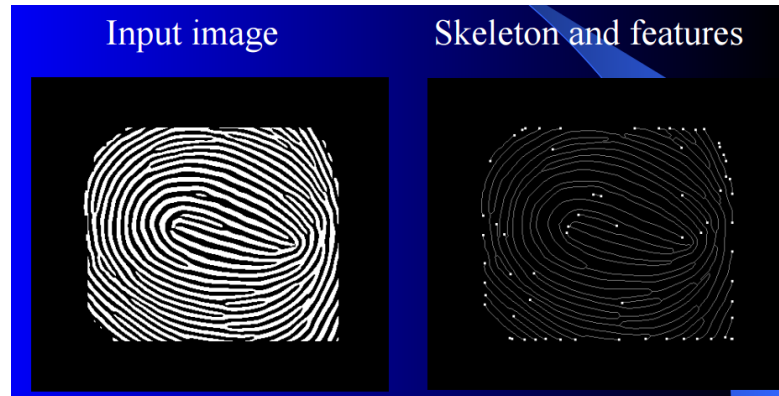


Fig 5.4 Thinning

5.2.3 Minutiae Extraction and Filtering

This step methodically scans the binary image of a fingerprint, identifying localized pixel patterns that indicate the ending or splitting of a ridge. The patterns searched for are very compact as illustrated. The left-most pattern contains six binary pixels in a 2×3 configuration. This pattern may represent the end of a black ridge protruding into the pattern from the right. The same is true for the next 2×4 pattern. The only difference between this pattern and the first one is that the middle pixel pair is repeated. In fact, this is true for all the patterns depicted. This "family" of ridge ending patterns can be represented by the right-most pattern, where the middle pair of pixels (signified by “*”) may repeat one or more times [40].

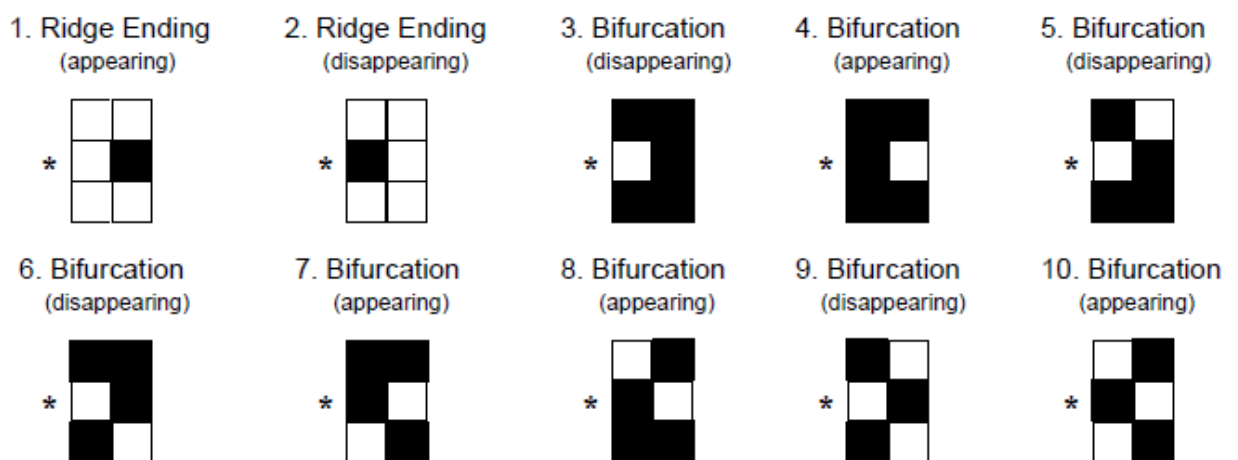


Fig 5.5 Masks

Candidate ridge endings are detected in the binary image by scanning consecutive pairs of pixels in the image looking for sequences that match this pattern. Pattern scanning is conducted both vertically and horizontally. The pattern as illustrated is configured for vertical scanning as the pixel pairs are stacked on top of each other. To conduct the horizontal scan, the pixel pairs are unstacked, rotated 90° clockwise, and placed back in sequence left to right. Using the representation above, a series of minutiae patterns are used to detect candidate minutia points in the binary fingerprint image. There are two patterns representing candidate ridge endings, the rest represent various ridge bifurcations. A secondary attribute of appearing/disappearing is assigned to each pattern. This designates the direction from which a ridge or valley is protruding into the pattern. All pixel pair sequences matching these patterns, as the image is scanned both vertically and horizontally, form a list of candidate minutia points [40].

5.2.4 Matching

- Local minutia match – Matches two minutiae relatively to already matched reference minutia. Filters by distance error, position angle error, and rotation angle error. Additionally filters by local orientation and count of matched neighbors.
- Match tree – This is developed using local minutia matches from some initial minutia match.
- Minutia neighborhood – Some fixed number of closest minutiae are precomputed to reduce number of tested minutiae.

Features used for matching:

- Shift on X , Shift on Y
- Distance concerning to other minutiae (Local Shift)
- Angle between two compared minutia
- Angle concerning to other minutiae
- Information about area around minutiae

5.3 Execution

For the purpose of implementation, we used the matching libraries provided by SourceAFIS using C# on a Windows environment.





- As per the requirements of the algorithm, a routine was taken from the libraries of the Futronic company for the purpose of capturing fingerprints on the FS88 scanner which











is interfaced with the Raspberry Pi. This scanner is compatible with both the Linux and Windows environments, so it works well with the Raspbian OS, which is Linux based.







- The next step required the captured image to be either enrolled into the database or checked with the existing database to see if there is a match. A shared folder is maintained on both the host computer as well as the Pi. Anytime someone tries to enroll, a redundancy check is done by seeing if the person already exists in the database (this reuses the match program). If not, the finger/person is enrolled into the database.
- Matching takes place when a person tries to identify with the database. In this case, the Pi sends the fingerprint image over the network to the host computer, where the SourceAFIS matching program is run and the result of the test is returned to the Pi terminal. Matching scores vary but the higher the score, the better the match. If below a certain threshold, a non-match will be returned.
- This method employs a one-to-many matching scenario and thus takes more time than a one-to-one situation. In the case that a candidate is matched to a person stored in the database, this prompts the facial recognition program to kick in and a one-to-one match with the existing user's file is executed.
- Thus, a two-tier security system is created and run.

Chapter 6

Results and Discussions

ID	Fingerprint	Fingerprint Score	Face	Face Prediction Score
sri		87.117		0.9
Bish		94.675		0.8

p1		88.881		0.75
p2		108.113		0.8
p3		86.678		0.85
p4		91.563		0.9
p5		113.783		0.8

p6		99.567		0.95
p7		102.452		0.85
p8		85.128		0.8

System Specifications

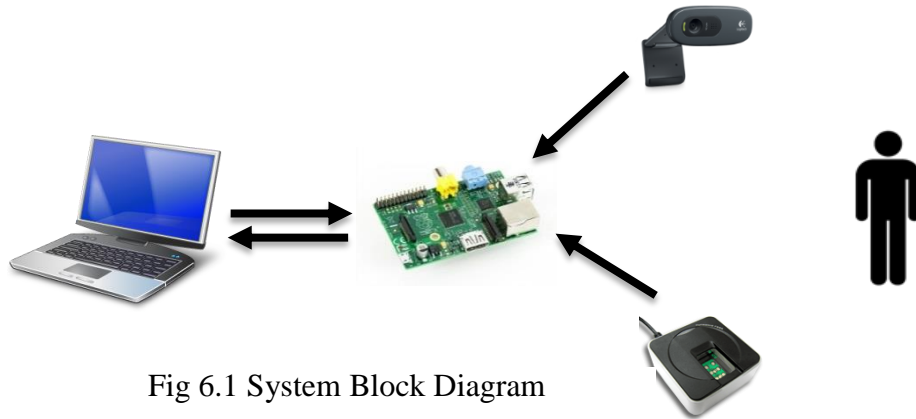


Fig 6.1 System Block Diagram

Fingerprint Test Database: 150 Fingerprints

Face Test Database: 30 Faces

- Enrollment Time (including redundancy check): 2 Minutes
- Fingerprint Recognition Time: 10 Seconds
- Facial Recognition Time: 40 – 50 Seconds
- Total System Authentication Time: 1 Minute
- Fingerprint Type 1 Error, False Rejection Rate: $0/150 = 0\%$
- Fingerprint Type 2 Error, False Acceptance Rate: $1/150 = 0.67\%$
- Facial Recognition Type 1 Error, False Rejection Rate: $1/30 = 3.33\%$
- Facial Recognition Type 2 Error, False Acceptance Rate: $1/30 = 3.33\%$

Chapter 7

Summary and Conclusions

Biometric authentication is a type of system that uses the unique biological characteristics of individuals to verify identity for secure logins into electronic systems. The biometric technologies involved are based on the ways in which individuals can be uniquely identified through one or more distinguishing biological traits, such as fingerprints, hand geometry, earlobe geometry, retina and iris patterns, voice waves, keystroke dynamics, DNA and signatures. Biometric authentication is the application of that proof of identity as part of a process validating a user for access to a system. Biometric technologies are used to secure a wide range of electronic communications, including enterprise security, online commerce and banking even just logging in to a computer or smart phone. We have used two traits in our project for authentication. The first one is fingerprint and the second is facial recognition.

Facial recognition (or face recognition) is a type of biometric software application that can identify a specific individual in a digital image by analyzing and comparing patterns. Based on various methods there are different approaches of facial recognition. Some of them are:

Input image normalization: Image normalization is the first stage for all face recognition systems. Firstly face area is detected in the image. We used template matching to localize a face. Then the eye (iris) centers should be detected because the distance between them is used as a normalization factor.

Geometrical Approach: The first historical way to recognize people was based on face geometry. There are a lot of geometric features based on the points. Geometric features may be generated by segments, perimeters and areas of some figures formed by the points.

Elastic face matching approach: Applying elastic transform we can change geometry and image texture and then compare two images. Given the variability of a face, even under controlled conditions it is futile to try to compare directly original images or even their feature maps. In fact, we have got very low similarity scores with the popular method of mosaic images, and using linear correlation.

Here we have carried it using Local Binary Pattern approach in OpenCV. The image is divided into small non overlapping grids. A histogram is generated that corresponds to each such grid. Then histograms of corresponding areas are summed up to evaluate the similarity of image. The facial recognition consists of following steps.

- Preprocessing
- LBP Operator application
- Local feature extraction
- Classification

Fingerprint recognition or fingerprint authentication refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify individuals and verify their identity. Pattern based algorithms compare the basic fingerprint patterns (arch, whorl, and loop) between a previously stored template and a candidate fingerprint. This requires that the images can be aligned in the same orientation. To do this, the algorithm finds a central point in the fingerprint image and centers on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match.

We start by summarizing the main common steps of the algorithms used in this work. Then each step is further discussed in detail. The face recognition process consists of four main parts:

Preprocessing: Normalization, Ridge orientation estimation, Ridge frequency estimation, Gabor Filtering

- Binarization: NIST-like binarization
- Thinning: Zhang-Suen thinning algorithm
- Minutiae Extraction: Crossing Number Algorithm
- Matching: Bozorth3 method using edge distances and orientations

Several components have been used in design of the system. Then main components are listed below.

- Raspberry Pi
- Futronic FS88
- Logitech C270

Security Enhancement

Biometrics has already improved the level of security. A single biometric is more susceptible to fraud. This could lead to easy break-in leading to loss of important. We have increased the level of security of the systems using two biometric in our system design. This gives us more secure transaction and access to important or confidential data.

A system is generally unlocked using passwords and patterns. When typing passwords or drawing patterns, however, they can be easily spied on or deduced. In contrast, fingerprint-based verification is safer because even if someone tries to spy on the fingerprint input process, fingerprint information cannot be leaked or deduced. Moreover, because the user simply has to touch the fingerprint sensor, the inconvenience resulted from the typing password. All system manufacturers are implementing biometric authentication systems in their products or developing corresponding sensors for system for improving security. Hence, the security of the password input process, which is the weakest link in controlling user access to system, can be improved by using dual biometric authentication.

Areas of Implementation

ATM machines: ATMs have been conventionally using Card-Pin type of authentication. These cards have a probability of been lost, duplicated or PIN being forgotten etc. With the mentioned type of authentication, we are not only reducing the overhead of Card/PIN but also improving the security. Fingerprint can't be lost and are unique to each person.

Mobile Security: The same is applicable to mobile security. With increase in usage of cell smart devices throughout the world there needs to a significant increase in privacy and security of cell phone data.

One of the most crucial problems in face recognition practice is the variations of light intensity in input images the present algorithm works properly in a good lighting condition. Modification of algorithm is required so that it can properly work in a poor lightening condition too.

At present we have been using 2- dimensional facial recognition. Further improvements can be made using 3-Dimensional facial recognition. This avoids such pitfalls of 2D face recognition algorithms as change in lighting, different facial expressions, make-up and head orientation. Another approach is to use the 3D model to improve accuracy of traditional image based recognition by transforming the head into a known view.

During fingerprint authentication the finger print data is transferred from Raspberry pi to Windows system. Here the system is susceptible to fraud or theft. There we can add a cryptographic technique which is based biometric key generation. A cryptographic key based on biometric traits is generated which is in turn used to encrypt the transmitted data.

Literature Cited

- [1] 2006 Futronic Technology Company Limited, FS88 FIPS201/PIV Compliant USB2.0 Fingerprint Scanner, http://www.futronic-tech.com/product_fs88.html
- [2] 2006 Futronic Technology Company Limited, Fact sheet about Futronic's Live Finger Detection(LFD) technology, http://www.futronic-tech.com/product_lfd.html
- [3] ALASDAIR ALLAN, Roger Meike, Arduino Uno vs BeagleBone vs Raspberry Pi, Makezine, 2013, <http://makezine.com/2013/04/15/arduino-uno-vs-beaglebone-vs-raspberry-pi/>
- [4] Le Hoang Thai and Ha Nhat Tam, Fingerprint recognition using standardized fingerprint model in International Journal of Computer Science Issues, Vol. 7, Issue 3, No 7, Doolar Lane, Mahebourg, Republic of Mauritius, 2010, pp. 11-17
- [5] Arun Ross and Anil Jain, Biometric Sensor Interoperability: A Case Study In Fingerprints in Proc. of International ECCV Workshop on Biometric Authentication (BioAW), LNCS Vol. 3087, New York City, Springer, 2004, pp.134-135
- [6] Biometric Functionality in Biometrics, <http://en.wikipedia.org/wiki/Biometrics>
- [7] Kresimir Delac, Mislav Grgic, A SURVEY OF BIOMETRIC RECOGNITION METHODS, 46th International Symposium Electronics in Marine, Zadar, Croatia, June 2004, pp. 185-193
- [8] Seema Rao, Prof.K.J.Satoa, An Attendance Monitoring System Using Biometrics Authentication, Volume 3 Issue 4, *International Journal of Advanced Research in Computer Science and Software Engineering*, Jaunpur, India, IJARCSSE, 2013, pp.379-383

- [9] Roli Bansal, Priti Sehgal, Punam Bedi, “Minutiae Extraction from Fingerprint Images - a Review” in IJCSI International Journal of Computer Science Issues, Mauritius: IJCSI, September 2011, Vol. 8, Issue 5, No 3
- [10] Manvjeet Kaur, Mukhwinder Singh, Akshay Girdhar, Parvinder S. Sandhu, “Fingerprint Verification System using Minutiae Extraction Technique” in World Academy of Science, Engineering and Technology, India, 2008, p.46
- [11] Abbad Khalid, Tairi Hamid, Aarab Abdellah, “Minutiae Extraction Based on Propriety of Curvature” in International Journal of Computer Theory and Engineering, Singapore, IJCTE, June 2011, Vol. 3, No. 3
- [12] Nimitha Chama, “Fingerprint Image Enhancement and Minutiae Extraction”, Dept. of Electrical & Computer Engineering, Clemson University
- [13] Raymond Thai, “Fingerprint Image Enhancement and Minutiae Extraction”, School of Computer Science and Software Engineering, The University of Western Australia, 2003
- [14] Joan Climent, “Fingerprint minutiae extraction using topographic distances” in The 8th International Symposium on Mathematical Morphology, Brazil: ISMM, October 2007, v. 2, p. 65–66
- [15] Zhixin Shi and Venu Govindaraju, “A chaincode based scheme for fingerprint feature extraction” in Elsevier BV: Pattern Recognition Letters, 2006, V.27, pp. 462–468
- [16] Atul S. Chaudhari, Dr. Girish K. Patnaik, Sandip S. Patil, “Implementation of Minutiae Based Fingerprint Identification System using Crossing Number Concept” in International Journal of Computer Trends and Technology, India: IJCTT, February 2014, volume 8, number 4
- [17] Hartwig Fronthaler, Klaus Kollreider, Josef Bigun, “Local Features for Enhancement and Minutiae Extraction in Fingerprints” in IEEE Transactions On Image Processing, IEEE, March 2008, VOL. 17, NO. 3
- [18] Alessandro Farina, Zsolt M. Kovacs-Vajna, Alberto Leone, “Fingerprint minutiae extraction from skeletonized binary images” in Elsevier BV: Pattern Recognition, 1999, V.32, pp. 877–889
- [19] Philippe Parra, “Fingerprint minutiae extraction and matching for identification procedure”, Department of Computer Science and Engineering, University of California, San Diego

- [20] Sunny Arief Sudiro and Rudi Trisno Yuwono, "Adaptable Fingerprint Minutiae Extraction Algorithm Based-On Crossing Number Method For Hardware Implementation Using FPGA Device" in International Journal of Computer Science, Engineering and Information Technology, India: IJCSEIT, June 2012, Vol.2, No.3
- [21] Chandra Bhan Pal, Amit Kumar Singh, Nitin, Amrit Kumar Agrawal, "An Efficient Multi Fingerprint Verification System Using Minutiae Extraction Technique", Department of CSE & IT, Jaypee University of Information Technology, India
- [22] Feng Zhao and Xiaou Tang, "Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction" in Elsevier BV: Pattern Recognition, 2007, V.40, pp. 1270–1281
- [23] Ming-Hsuan Yang, Narendra Ahuja, and David Kriegman, "A Survey on Face Detection Methods", March 1999.
- [24] Rama Chellappa, Charles L. Wilson and Saad Sirhoei, "Human and Machine Recognition of Faces: A Survey", Proc. IEEE, Vol. 83, No. 5, May 1995.
- [25] Hua Gu, Guangda Su, Cheng Du, " Feature Points Extraction from Faces", Research Institute of Image and Graphics, Dept. of Electronic Engineering, Tsinghua University, Beijing, China..
- [26] Yongzhong Lu, Jingli Zhou, Shengsheng Yu, "A Survey Of Face Detection, Extraction And Recognition", National Storage System Laboratory, School of Software Engineering, Huazhong University of Science and Technology, China, June 2002.
- [27] Sushil Kumar Paul, Mohammad Shorif Uddin and Saida Bouakaz, "Extraction of Facial Feature Points Using Cumulative Histogram", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January 2012.
- [28] Wu-Chih Hu, Ching-Yu Yang, Deng-Yuan Huang, and Chun-Hsiang Huang, "Feature-based Face Detection Against Skin-color Like Backgrounds with Varying Illumination", Journal of Information Hiding and Multimedia Signal Processing, Volume 2, Number 2, April 2011
- [29] Jean-Jacques Orban de Xivry, Meike Ramon, Philippe Lefevre and Bruno Rossion, "Reduced fixation on the upper area of personally familiar faces following acquired prosopagnosia", Universite' catholique de Louvain, Louvain-la-Neuve, Belgium, Journal of Neuropsychology (2008), 2, 245–268, 2008 The British Psychological Society.

- [30] Ivana Atanasova, Biljana Perchinkova, "Minimum Set Of Geometric Features in Face Recognition", European University – RM, The 8th International Conference for Informatics and Information Technology (CIIT 2011).
- [31] Yuseok Ban, Sang-Ki Kim, Sooyeon Kim, Kar-Ann Toh , Sangyoun Lee, "Face Detection Based On Skin Color Likelihood", in Pattern Recognition vol. 47, pp. 1573–1585, 2014.
- [32] Hongming Zhang and Debin Zhao, "Spatial Histogram Features for Face Detection in Color Images" Department of Computer Science and Engineering, Harbin Institute of Technology, Harbin, China, Springer-Verlag Berlin Heidelberg: PCM 2004, LNCS 3331, pp. 377–384.
- [33] Qiming Li, Muchuan Guo and Ee-Chien Chang, "Fuzzy Extractors for Asymmetric Biometric Representations", National University of Singapore.
- [34] Haiyuan Wu, Qian Chent and Masahiko Yachida, "A Fuzzy-Theory-Based Face Detector", IEEE, Proc. ICPR, 1996.
- [35] Henry A. Rowley, Shumeet Baluja and Takeo Kanade, "Neural Network-Based Face Detection", in Computer Vision and Pattern Recognition, School of Computer Science, Carnegie Mellon University, Pittsburg, USA, 1996.
- [36] M. Turk and A. Pentland, "Eigen-faces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.
- [37] Vidya Manian and Arun Ross, "A Texture-based Approach to Face Detection", Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, West Virginia.
- [38] Daniel Maturana, Domingo Mery and A´lvaro Soto, "Face Recognition with Local Binary Patterns, Spatial Pyramid Histograms and Naive Bayes Nearest Neighbor classification", Pontificia Universidad Cat´olica, Santiago, Chile.
- [39] Oleg Ostap, "Presentation Preprocessing and Features Extraction Algorithm", <http://fingerprintreco.cvs.sourceforge.net/viewvc/fingerprintreco/fingerprintreco/Documents/Presentations/Processing.pdf?revision=1.1>
- [40] Craig I. Watson, Michael D. Garris, Elham Tabassi, Charles L. Wilson, R. Michael McCabe, Stanley Janet, and Kenneth Ko "User's Guide to NIST Biometric Image Software (NBIS)", National Institute of Standards and Technology, Gaithersburg, MD

- [41] Volodymyr Ostap, "Matching Algorithm Presentation",
<http://fingerprintreco.cvs.sourceforge.net/viewvc/fingerprintreco/fingerprintreco/Documents/Presentations/Matching.pdf?revision=1.1>

Acknowledgements

We are highly appreciative of Sri S.K.L.V. Sai Prakash, Associate Professor, Department of Electronics and Communication Engineering, NIT Warangal, for his continuous guidance, involvement and motivation throughout our project work and also for providing all the facilities required for the fulfillment of the project.

We are grateful to the panel members who helped oversee highly constructive critique of the work done, and especially to Sri. M.V. Raghunath and Sri.P. Sreehari Rao for their valuable suggestions and constant support throughout the duration of the project.

We would also like to extend our gratitude to the head of the department, Dr. T. Kishore Kumar, for providing us the resources needed to perform various tasks essential for the completion of the project. All the department faculty members were extremely helpful with their suggestions and input. Finally, we would like to thank the technical staff of the department for their cooperation and assistance in the utilization of the necessary facilities.

We are also thankful to our friends and family, as well as everyone else that helped push this project in the right direction.

Bishwarup Neogy

Srikar Chintapalli

Akshay Meher