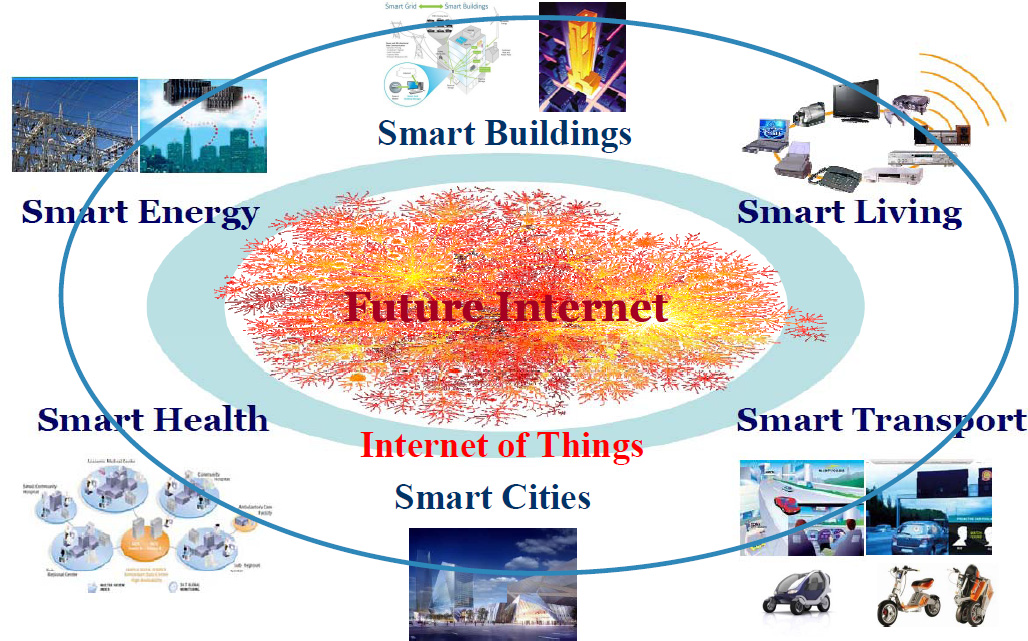
**CHAPTER 1**

**INTRODUCTION**

* 1. **Introduction to IoT**

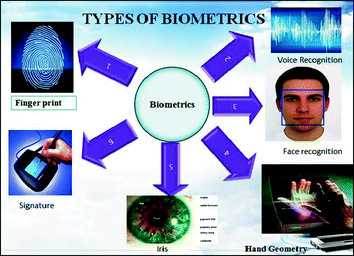
The Internet of things (IoT) is the extension of internet connectivity into physical devices and everyday objects. Embedded with electronics, Internet connectivity, and the forms of hardware (such as sensors), these devices can communicate and interact with others over the internet, and they can remotely monitored and controlled. It means taking all the things in the world and connecting to the internet.



**Fig 1.1 IOT**

* 1. **Biometrics**

The word biometrics comes from the Greek words bios and metric, meaning “life measurement”. By measuring something unique about an individual and using that to identify them, we can achieve a dramatic improvement in security of the key store. Newer biometric measurements include DNA from tissue samples, voice pattern, face pattern or even the arrangement of blood vessels in the retina or pattern of coloration in the cornea of the eye.



**Fig 1.2 Biometrics**

* 1. **What is fingerprint**

Fingerprints for identification and recognition purposes have been in use since a long time. Thumb prints using ink is one of the oldest application of biometrics. There are multiple reasons for fingerprints becoming common as biometric feature. First, the fingerprint recognition provides a reliable form of biometric security even when a person ages, whereas the iris and facial recognition are affected by ageing related feature changes. Second, fingerprint recognition is not affected much by change in appearance, whereas iris recognition needs a person to remove lenses or glasses. Third, fingerprint scanners can successfully identifying a registered fingerprint even with certain amount of unwanted substance present on the skin such as dust ,oil ,dirt, powder or liquid etc., whereas facial recognition systems are prone to error due to occlusion by person’s facial hair. In order to use fingerprints as biometric feature, one needs to understand several features of the fingerprint pattern. It is essential to understand the structure and properties of the human skin, in order to employ some image processing methods to observe the unique characteristics in these patterns.



**Fig 1.3 Fingerprint**

**These unique characteristic are:**

* Ridges
* Minutiae points

**Ridges:**

In the biometric process of finger scanning, a ridge is curved line in a finger image .Some ridges are continuous curves, and others terminates at specific points called ridge endings. A normal fingerprint pattern is made up of lines and spaces. These lines are called ridges, and the spaces between the ridges are called valleys.

**Three basic patterns:**

* Loop
* Whorl
* Arch

****

**Minutiae points:**

The unique fingerprint traits are termed as minutiae. It refers to specific plot points on a fingerprint.

**Major features:**

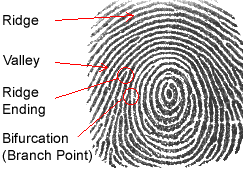
* Ridge ending
* Bifurcation

**Ridge ending:**

The ridge ending is the point at which a ridge terminates.

**Bifurcation:**

Bifurcations are points at which a single ridge splits into two ridges.



* 1. **Why use fingerprints**

Fingerprints are considered to be the best and fastest method for biometric Identification. Besides these, implementation of fingerprint recognition system is cheap, easy and accurate up to satisfiability. A person’s fingerprints do not change over time.

* 1. **Fingerprint Recognition**

It is one of the best known and most widely used biometric technologies. This technology extracts features from impressions made by the distinct ridges on the fingerprint. It refers to the automated method of identifying or confirming the identity of an individual based on the comparison of two fingerprint .The fingerprint recognition can be grouped into two sub-domains: (a) Fingerprint verification and (b) Fingerprint identification.

**(a)Fingerprint verification:**

It also known as fingerprint authentication. It is the process of confirming that a user who they claim to be.

**(b)Fingerprint identification:**

It is the method of identification based on the different patterns of human fingers, which is actually unique among each person. It is most popular way of acquiring details of any person and his the most easy and convenient way of identifying a person.

**Implementation:**

* Image enhancement
* Thinning
* Minutiae extraction

**Image enhancement:**

It is used to make the image clearer for easy further operations.



**Thinning:**

It is to eliminate the redundant pixels of ridges till the ridges are just one pixel wide.



**Minutiae extraction:**

Minutiae points are extracted from the enhanced fingerprint image. These are the major features of a fingerprint image and are used in the matching of fingerprints .These minutiae points are used to determine the uniqueness of a fingerprint image. A good quality fingerprint image can have 25 to 80 minutiae depending on the fingerprint scanner resolution and the placement of finger on the sensor. It is the most widely used technique of fingerprint representation and its configuration is highly distinctive. It is more accurate compared to other correlation based systems and the template size is smaller in minutiae-based fingerprint representation .In this system, two fingerprints match if their minutiae points match. Minutiae based fingerprint technique is the backbone of most currently available fingerprint recognition products.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 “Biometric Attendance System,” IEEE/ICME International Conference on Complex Medical Engineering, 2011.**

**Authors: A Ujan, Ismaili IA.**

This research work has application for attendance system of employer’s and students in general. The system will facilitate institutions/organization to make attendance individual in time along with data information thumb impression will be taken as a signature for the system entry. Main design and challenge in this system is the design of database architecture and its business logic.

**2.2”Automated Attendance Management System based on face Recognition,”IEEE International Conference on Computational Intelligence and Computing Research, 2013.**

**Authors: Chintalapati S, Raghunadh MV.**

In this paper we propose an automated attendance management system. This system, which is based on face detection and recognition algorithms, automatically detects the student when he enters the classroom and marks the attendance by recognizing him. The system architecture and algorithms used in each stage are described in this paper. Different real tie scenarios are considered to evaluate the performance of various face recognition systems. This paper also proposes the techniques to be used in order to handle the threats like spoofing. When compared to traditional attendance marking this system saves the time and also helps to monitor the students.

**2.3 “Fingerprint Biometric based Access Control,” IEEE INDICON, 2015**

**Authors: Yash Mittal, Aishwary Varshney,Prachi Aggarwal.**

Fingerprint is a reliable biometric feature having a wide range of applications that require authentication. Person specific verification is needed in many scenarios such as access control, classroom attendance and financial transactions etc. In this paper, two applications of fingerprint biometric are proposed. An other prototype of a Classroom Attendance Management System (CAMS) is developed that uses fingerprint as biometric feature for classroom attendance. The CAMS consists of modules for database, web-user interface and views at multiple levels of access. Both systems are expected to mitigate the shortcomings of alternative existing systems, and eliminate the possibilities of spoofing or proxy. These systems store fingerprints along with the date/time-stamp for each user. Fingerprints are stored dynamically in a database for computing the different statistics, e.g., month-wise or semester-wise trends in the case of CAMS. The CAMS can also provide a solution to the problem of late-coming. Experiments are conducted for measuring the recognition accuracy at 87% for ACS and 92% for CAMS are encouraging. The proposed systems can be further scaled up for real-time deployment, in applications such as employee attendance and controlled access to high-security areas etc.

**2.4”A Robust Fingerprint Recognition Technique Applying Minutiae Extractors and Neural Network,” International Journal of Engineering Research and Advance Technology (IJERAT) , 2019.**

Fingerprint recognition is one of the interesting and mature biometric technique for personal identification or authentication application. The objective of the work is to retrieve personal data by using the fingerprint. In this paper the fingerprint of a person is captured via digital persona U.are.U 4500 fingerprint scanner. The input fingerprint is firstly enhanced using contrast stretching and two morphological operations (dilation and erosion). Next, 160\* 160 size of the center region of the fingerprint is cropped and extracted minutiae such as ridges and bifurcations. These minutiae are used as the features of a particular fingerprint. These features are put into the neural network for person recognition. Various experiments will confirm and prove the accuracy and performance of the proposed fingerprint recognition system.

**2.5 “An efficient Algorithm for fingerprint preprocessing and feature extraction” ScienceDirect, 2010.**

In practice, the placement of finger on the scanner for authentication is not done with the utmost care as when placed during the enrollment and this result in rejections of genuine users. Moreover, user behavior and environmental conditions decrease the genuine acceptance rate (GAR) for authentication of fingerprints. To overcome these limitations, an efficient preprocessing algorithms is proposed to achieve good vertical orientation and high ridge curvature area around the core point for fingerprint authentication and analysis. The algorithm is implemented in two stages. The process of obtaining the vertical oriented fingerprint image is carried out in the first step. The developed algorithm is tested using a line based feature extraction algorithm with a large internal database and samples of fingerprint verification competition (FVC).Only for the poor quality images, broken ridges are identified which results in a difference in minutiae points. With the proposed algorithm 94% of the tested images were oriented vertically and its genuiness is verified by comparing the minutiae details of the oriented and unoriented image of the same subject.

**CHAPTER 3**

**SYSTEM ANALYSIS AND DESIGN**

**3.1 EXISTING SYSTEM**

* The project work aims at designing a student attendance system which could effectively manage attendance of students.
* In this project work, attendance is marked after student’s biometric identification. For student identification, a fingerprint recognition based identification system is used.
* Fingerprint features are considered to be the best and fastest method for biometric identification.
* These features are more secure to use and unique for every person that don’t change in one's lifetime.
* Fingerprint recognition is a mature field today, but still identifying individual from a set of enrolled fingerprints is a time taking process.
* It was very necessary to improve the fingerprint identification system for implementation on large databases, e.g. of an institute or a country.
* In this project, the minutiae algorithm is used to develop the identification system which is faster in implementation than any other available today in the market.
* The proposed automated attendance system based on fingerprint recognition was tested on a class of student fingerprint databases and achieved significant results for taking an attendance of the students.

Fingerprint processing

Fingerprint verification

Take class attendance

User interface

Fingerprint enrollment

Report generation

**3.1.1 Limitations:**

* There are some limitations of the fingerprint technology.
* There is no fixed time based attendance system.
* The process is very time–consume of the students as well as staffs.

**3.2 PROPOSED SYSTEM**

The proposed work is centered at marking the attendance of a candidate by taking his ﬁnger print and then comparing some speciﬁc features with those of the ﬁnger prints present in a predeﬁned database. It involves,

**Step-1** **Load Image**

The load image function is created to provide users to input an image for validation.

**Step-2** **Binary Conversion and Thinning Operation**

The input image is then converted into binary image. Thinning operation is then carried out so that the edges in the binary image is made single pixel thick. This is used to reﬁne the output for edge detection.

**Step-3 Feature Extraction and False Feature Detection and Removal**

The features like ridge ending, bifurcation are detected. As there is a lot of false features detection, a quality check measure is done.

**Step-4** **Region/Area of Interest**

The region/area of interest is detected. The ROI is performed in two morphological methods – OPEN and CLOSE. The OPEN operation can enlarge the images and eliminate background noise. And CLOSE operation can shrink images.

**Step-5** **Validation**

At this point the user can either choose to add the input image to the database or to validate it. In case of validation, the system compares the output of the previous step with all the images present in the database.

**CHAPTER 4**

**SYSTEM REQUIREMENTS**

**4.1 GENERAL**

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware, hardware requirements list is often accompanied by a hardware compatibility list (HCL), especially in case of operating systems. An HCL lists tested, compatible, and sometimes incompatible hardware devices for a particular operating system or application. The following sub-sections discuss the various aspects of hardware requirements.

**4.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

* System : Intel corei3 7th gen.
* Hard Disk : 1 TB.
* Monitor : 15 VGA Color.
* Mouse : Logitech
* RAM : 4 GB

**4.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Operating system : Windows 10.
* Front End : Matlab 2015.

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 GENERAL**

Design Engineering deals with the various UML (Unified Modeling Language) diagrams for the implementation of project. Software design is a process through which the requirements are translated into representation of the software. Design is the means to accurately translate customer requirements into finished product.

**5.1.1 SYSTEM ARCHITECTURE**

**ENROLLMENT**

User interface

Feature extraction

Database

**AUTHENTICATION**

Minutiae extraction

User interface

Matching

Result

**Description:**

The system architecture involves two steps namely (a) Enrollment and (b) Authentication. It involves the process of giving input that communicate with the system via user interface. Then it performs the process of feature extraction. The database contain all the stored fingerprint pattern of the students. Finally, it leads to the process of Authentication. In this step it involves the user interface that is student fingerprint. Then it involves feature extraction. It involves the task of matching that is by comparing with all the preregistered fingerprint in the database. If valid match, then it returns the result.

**5.1.2 Use Case Diagram**

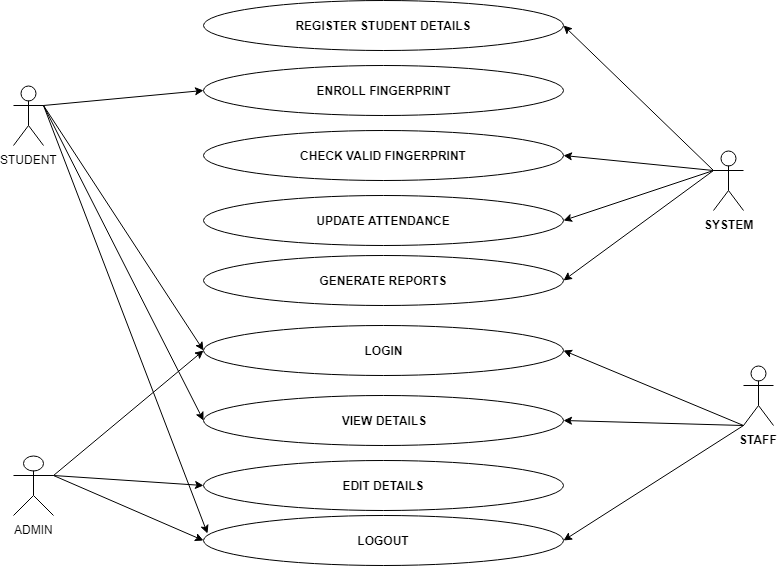
A use case is a list of actions or event steps typically defining the interactions between a role of an actor and a system to achieve a goal. It is a useful technique for identifying, clarifying, and organizing system requirements. It is made up of a set of possible sequences of interaction between system and users that defines the features to be implemented and the resolutions of any errors that may be encountered.

**Characteristics:**

* Organizes functional requirements
* Models the goals of system/ actor (user interaction)
* Describes one main flow of events

**Notations:**

* Actor
* Use case
* Relationship
* System boundary



**5.1.3 Data Flow Diagram**

A data flow diagram (DFD) is a graphical view of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be use for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

**Notations:**

* Data Flow
* Process
* Entity
* Data Store

**Data Flow:**

They are pipelines through the packets of information flow.

**Process:**

A process or task performed by the system.

**Entity:**

They are object of the system. A source or destination data of a system.

**Data Store:**

A place where data to be stored.

**Rules:**

* Each process should have at least one input and an output.
* Each data store should have at least one data store in and one data flow out.
* Data stored in a system must go through a process.
* All processes in a DFD go to another process or a data store.

**Categories:**

* DFD level 0
* DFD level 1
* DFD level 2

**DFD level 0:**

It is also called a Context Diagram. It’s basic overview of the whole system.

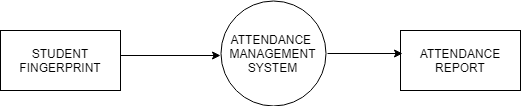
**DFD level 1:**

It provides a more detailed breakout of pieces of the Context Level Diagram.

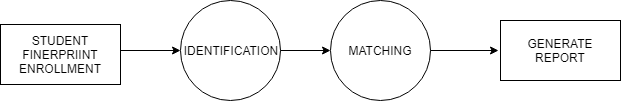
**DFD level 2:**

It may require more text to reach the necessary level of detail about the system’s functioning.

**DFD Level 0:**

****

**DFD Level 1:**

****

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 Module Specifications**

* Enrollment
* Authentication
* System database

**6.2 Description**

**Enrollment:**

The task of enrollment module is to enroll users and their ﬁngerprints into the system database. During enrolment, the fingerprint and other bio-data of the user is captured and the unique features are extracted from the fingerprint image and stored in a database as a template for the subject along with the user’s ID. Staff bio data to be captured includes: employee number, surname, other names, sex, position, staff type, phone number, email, department and passport photograph. Student bio data includes: matriculation number, surname, other-names, sex, department, level, studentship, phone number and passport photograph. To improve the quality of a captured image during enrolment/registration, two image samples per fingerprint used are captured for a higher degree of accuracy. When the ﬁngerprint images and the user name of a person to be enrolled are fed to the enrollment module, a minutiae extraction algorithm is ﬁrst applied to the ﬁngerprint images and the minutiae patterns (features) are extracted. These features form a template that is used to determine the identity of the user, formulating the process of authentication. The enrolment process is carried out by an administrator of the attendance management system. The enrolment and registration phase is an administrative phase. The user fingerprint as well as other bio-data is stored for the first time into the database for registration. All data and information required for the proper recording of attendance are enrolled in this module.

**Authentication:**

The task of the authentication module is to validate the identity of the person who intends to access the system. The person to be authenticated indicates his/her identity and places his/her finger on the fingerprint scanner. The fingerprint images captured is enhanced and thinned at the image processing stage, and at feature extraction stage, the biometric template is extracted. It is then fed to a matching algorithm, which matches it against the person’s biometric template stored in the system database to establish the identity. During authentication, for staff attendance, a staff supply his/her department and name, then places his/her finger over the fingerprint reader, the fingerprint recognition unit compares the fingerprint features with those stored in the database, after a successful match, the staff’s employee number is sent to the database alongside the time of making such an attendance and update the status (either present/absent) of user’s attendance for the day. Staff attendance is captured twice a day for both arrival and departure time. For student attendance, the lecturer (or a designated personnel as the case may be) selects his/her department, level, course code, attendance type (for example lecture) and the attendance ID, then the student places his/her fingerprint on the fingerprint reader; the fingerprint recognition unit compares the fingerprint features with those stored in the database, after a successful match, the student’s matriculation number is sent to the database alongside the time of making such attendance and update the status (either present/absent) of student’s attendance for the class. Student attendance is captured only once for each attendance type. Fingerprint matching approaches includes minutiae based matching, ridge-based matching and the correlation matching approaches. However, it is believed that minutiae-based matching approach, upon which this work is based facilitates the design of a robust, simple, and fast veriﬁcation algorithm while maintaining a small template size. Minutiae-based representation is commonly used, primarily because forensic examiners have successfully relied on minutiae to match fingerprints for more than a century, minutiae-based representation is storage efficient, and expert testimony about suspect identity based on mated minutiae is admissible in courts of law.

**Database:**

The attendance management system database consists of tables that stores records, each of which corresponds to an authorized person that has access to the system. Each record may contain the minutiae templates of the person’s fingerprint and user name of the person or other information such as pin no as an index to the template. The database design for the system implements relational data model which is a collections of tables in which data are stored.

**CHAPTER 7**

**CONCLUSION AND FUTURE ENHANCEMENT**

**CONCLUSION**

An applicable attendance management system was designed for educational organizations in this project. This project mainly comprised of development of attendance management system and fingerprint identification system. This project presented a framework in which attendance management was made automated and on-line. There are some limitations of the fingerprint technology. These are the inability to enroll some users for poor fingerprints. For these cases one need to consider another biometric features. Also it can suffer some small changes along the time. To overcome this problem, the system may be necessary to re-enroll the fingerprint and/or use multiple fingerprints enrollment. The system needs to deploy specialized devices for fingerprint enrollment. In future this project can be extended to store fingerprint databases on the remote server that can be used over world-wide. A website will be hosted on the server for online access to attendance reports. The proposed system can be implemented for all classes of the university if sufficient funds will be provided to us.

**FUTURE WORK**

1.Two computers connected via LAN and a ﬁngerprint scanner will be used initially .One computer will serve the purpose of server for storing reports which may be MS Access, MS Excel or SQL/Oracle database. Other one will be storing the enrolled database, will have software for automatic attendance management and will be connected to USB ﬁngerprintscanner.

2. Software for automatic attendance management that will run on nodes could either be developed in the Mat lab or Java. We can run java code on mat lab, also java is better for handling network communications. So except ﬁngerprint related functions (which are already coded in Mat lab), rest of automatic attendance management software will be designed using java.

3. A website will be hosted on the server for online access to attendance reports. For this purpose, html, JSP or ASP dot net would be used.

4. Fingerprint identiﬁcation system will be improved further using more indexing techniques like ridge density tolerance etc.

5. Instead of using database available on internet, we would be using database of students.

**CHAPTER 8**

**CODING**

clear all; clc; addpath(genpath(pwd));

load('db.mat');

filename='101\_4.tif';

img = imread(filename);

if ndims(img) == 3; img = rgb2gray(img); end % Color Images

disp(['Extracting features from ' filename ' ...']);

ffnew=ext\_finger(img,1);

S=zeros(10,1);

for i=1:10

second=['10' num2str(fix((i-1)/8)+1) '\_' num2str(mod(i-1,8)+1)];

fprintf(['Computing similarity between ' filename ' and ' second ' from FVC2002 : ']);

S(i)=match(ffnew,ff{i});

fprintf([num2str(S(i)) '\n']);

drawnow

end

**MATCHED FINGERPRINTS**

Matched\_FigerPrints=find(S==1)

if (Matched\_FigerPrints == 1)

fprintf(['Akash IT ROLL.NO:1 Present']);

end

if (Matched\_FigerPrints == 2)

fprintf(['Adelin IT ROLL.NO:2 Present']);

end

if (Matched\_FigerPrints == 3)

fprintf(['Gracia IT ROLL.NO:3 Present']);

end

if (Matched\_FigerPrints == 4)

fprintf(['Sowmiya IT ROLL.NO:4 Present']);

end

if (Matched\_FigerPrints == 5)

fprintf(['Sri IT ROLL.NO:5 Present']);

end

if (Matched\_FigerPrints == 6)

fprintf(['prabhu IT ROLL.NO:6 Present']);

end

if (Matched\_FigerPrints == 7)

fprintf(['venkat IT ROLL.NO:7 Present']);

end

if (Matched\_FigerPrints == 8)

fprintf(['raju IT ROLL.NO:8 Present']);

end

if (Matched\_FigerPrints == 9)

fprintf(['suba IT ROLL.NO:9 Present']);

end

if (Matched\_FigerPrints == 10)

fprintf(['viki IT ROLL.NO:10 Present']);

end

**EXTRACTING FEATURE FROM A FINGERPRINT IMAGE**

function [ ret ] = ext\_finger( img, display\_flag )

if nargin==1; display\_flag=0; end

block\_size\_c = 24; YA=0; YB=0; XA=0; XB=0;

**Enhancement**

if display\_flag==1; fprintf(' >>> enhancement '); end

yt=1; xl=1; yb=size(img,2); xr=size(img,1);

for x=1:55

if numel(find(img(x,:)<200)) < 8

img(1:x,:) = 255;

yt=x;

end

end

for x=225:size(img,1)

if numel(find(img(x,:)<200)) < 3

img(x-17:size(img,1),:) = 255;

yb=x;

break

end

end

for y=200:size(img,2)

if numel(find(img(:,y)<200)) < 1

img(:,y:size(img,2)) = 255;

xr=y;

break

end

end

for y=1:75

if numel(find(img(:,y)<200)) < 1

img(:,1:y) = 255;

xl=y;

end

end

[ binim, mask, cimg, cimg2, orient\_img, orient\_img\_m ] = f\_enhance(img);

**Finding Minutiae**

if display\_flag==1; fprintf('done.\n >>> finding minutiae '); end

minu\_count = 1;

minutiae(minu\_count, :) = [0,0,0,0,0,1];

min\_path\_index = [];

% loop through image and find minutiae, ignore certain pixels for border

for y=20:size(img,1)-14

for x=21:size(img,2)-21

if (thinned(y, x) == 1) % only continue if pixel is white

% calculate CN from Raymond Thai

CN = 0; sx=0; sy=0;

for i = 1:8

t1 = p(thinned, x, y, i);

t2 = p(thinned, x, y, i+1);

end

progress=progress+1;

end

if progress < 10

continue

end

if mod(atan2(y-yy,xx-x), 2\*pi) > pi

m\_o=m\_o+pi;

end

end

minutiae(minu\_count, :) = [ x, y, CN, m\_o, m\_f, 1];

min\_path\_index(minu\_count, :) = [sx sy];

minu\_count = minu\_count + 1;

end

end % if pixel white

end % for y

end % for x

**Filtering False Minutiae**

if display\_flag==1; fprintf('done.\n >>> filtering false minutiae '); end

minu\_count = minu\_count -1;

t\_minutiae = [];

t\_minu\_count = 1;

t\_mpi = [];

for i=1:minu\_count

X = minutiae(i,1); Y = minutiae(i,2);

rc=0;

for y=max(Y-2,1):min(Y+2, size(binim,1))

if rc > 0

break

end

for x=max(X-2,1):min(X+2, size(binim,2))

if mask(y,x) == 0

rc = rc + 1;

break

end

end

end

if rc > 0

continue;

else

t\_minutiae(t\_minu\_count, :) = minutiae(i, :);

t\_mpi(t\_minu\_count, :) = min\_path\_index(i, :);

t\_minu\_count = t\_minu\_count + 1;

end

end

minutiae = t\_minutiae;

min\_path\_index = t\_mpi;

minu\_count = size(minutiae,1);

t\_minu\_count = 1; t\_minutiae = [];

dist\_m = dist2(minutiae(:,1:2), minutiae(:,1:2));

dist\_test=49;

for i=1:minu\_count

reject\_flag = 0;

P\_x = minutiae(i,1); P\_y = minutiae(i,2);

for j = i + 1 : minu\_count

if dist\_m(i,j) <= dist\_test

reject\_flag = 1;

end

end

if reject\_flag == 0 && mask(P\_y, P\_x) > 0

reverse\_p = 0;

if min\_path\_index(i,1) == 0

x = P\_x;

y = P\_y;

else

x = min\_path\_index(i,1);

y = min\_path\_index(i,2);

end

p1x=P\_x; p1y=P\_y;

x1=x; y1=y;

iter = 0;

for m=1:path\_len

iter = iter + 1;

cn = 0;

for ii = 1:8

t1 = p(thinned, x1, y1, ii);

t2 = p(thinned, x1, y1, ii+1);

cn = cn + abs (t1-t2);

minutiae\_for\_sc = [minutiae(:,1)/size(img,2) (tmpvec1 - minutiae(:,2) + tmpvec2)/size(img,1)];

dist\_m = sqrt(dist2(minutiae\_for\_sc(:,1:2), minutiae\_for\_sc(:,1:2)));

for i=1:minu\_count

[d,ind] = sort(dist\_m(i,:));

for j = 1 : minu\_count

if dist\_m(i,ind(j)) == 0

continue

end

theta\_t = mod(atan2(minutiae(i,2) - minutiae(ind(j),2), minutiae(i,1) - minutiae(ind(j),1)), 2\*pi);

ridge\_count = 0;

p\_y = minutiae(i,2); p\_x = minutiae(i,1);

t\_x = 0; t\_y = 0;

current=1; radius = 1;

while p\_y ~= minutiae(ind(j),2)

if thinned(p\_y, p\_x) > 0 && current == 0 && (t\_x ~= p\_x || t\_y ~= p\_y)

current = 1;

ridge\_count = ridge\_count + 1;

else

if thinned(p\_y, p\_x) == 0

current = 0;

end

end

t\_x = p\_x; t\_y = p\_y;

p\_x = round(minutiae(i,1) - radius\*cos(theta\_t));

p\_y = round(minutiae(i,2) - radius\*sin(theta\_t));

radius = radius + 1;

end

end

end

if core\_val < 1

minutiae(minu\_count+1, :) = [core\_x, core\_y, 5, start\_t, 0,1];

minu\_count = minu\_count + 1;

end

if dt1 < 1

minutiae(minu\_count+1, :) = [delta1\_x, delta1\_y, 7, 0, 1,1];

minu\_count = minu\_count + 1;

end

if dt2 < 1

minutiae(minu\_count+1, :) = [delta2\_x, delta2\_y, 7, 0, 1,1];

minu\_count = minu\_count + 1;

end

if dt3 < 1

minutiae(minu\_count+1, :) = [delta3\_x, delta3\_y, 7, 0, 1,1];

minu\_count = minu\_count + 1;

end

**Return Minutiae**

if display\_flag == 1

fprintf('done.\n');

minutiae\_img = uint8(zeros(size(img, 1),size(img, 2), 3));

for i=1:minu\_count

x1 = minutiae(i, 1); y1 = minutiae(i, 2);

if minutiae(i, 3) == 1 %Termination

if minutiae(i, 4) > pi

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [255, 0, 0];

end

end

else

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [205, 100, 100];

end

end

end

elseif minutiae(i, 3) == 2

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [255, 0, 255];

end

end

elseif minutiae(i, 3) == 3 **Bifurcation**

if minutiae(i, 4) > pi

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [0, 0, 255];

end

end

else

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [255, 0, 255];

end

end

end

elseif minutiae(i, 3) == 5

for k = y1-4: y1 + 4

for l = x1-4: x1 + 4

minutiae\_img(k, l,:) = [0, 255, 0];

end

end

elseif minutiae(i, 3) > 5

for k = y1-2: y1 + 2

for l = x1-2: x1 + 2

minutiae\_img(k, l,:) = [128, 128, 0]; % gold for delta

end

end

end

end

combined = uint8(minutiae\_img);

for x=1:size(binim,2)

for y=1:size(binim,1)

if mask(y,x) == 0

combined(y,x,:) = [0,0,0];

continue

end

if (thinned(y,x)) % binim(y,x))

combined(y,x,:) = [255,255,255];

else

combined(y,x,:) = [0,0,0];

end % end if

if ((minutiae\_img(y,x,3) ~= 0) || (minutiae\_img(y,x,1) ~= 0) ) || (minutiae\_img(y,x,2) ~= 0)

combined(y,x,:) = minutiae\_img(y,x,:);

end

end % end for y

end % end for x

if core\_val < 1 && YA > 0

for k = YA-2: YA + 2

for l = XA-2: XA + 2

combined(k,l,:) = [20, 255, 250];

end

end

for k = YB-2: YB + 2

for l = XB-2: XB + 2

combined(k,l,:) = [20, 255, 250];

end

end

end

subplot(1,2,1), subimage(img), title('Original image')

subplot(1,2,2), subimage(combined), title('Minutiae')

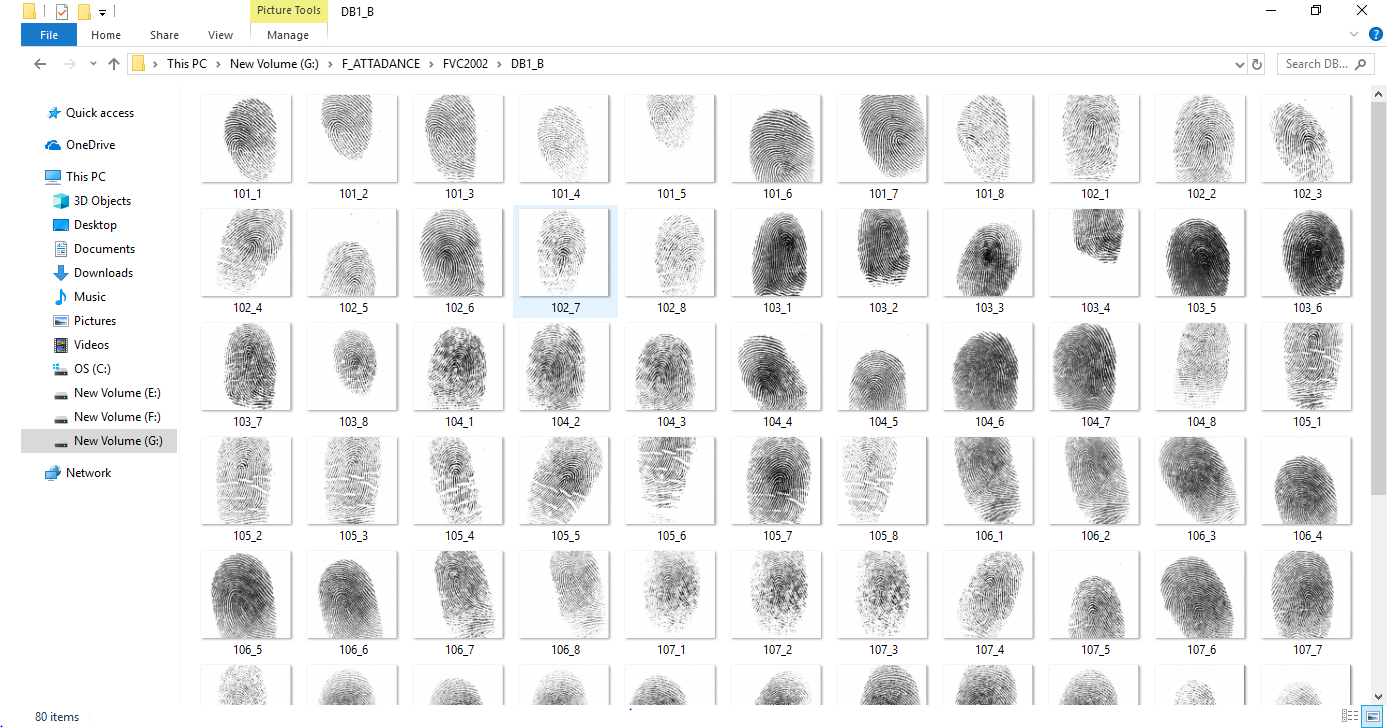
end

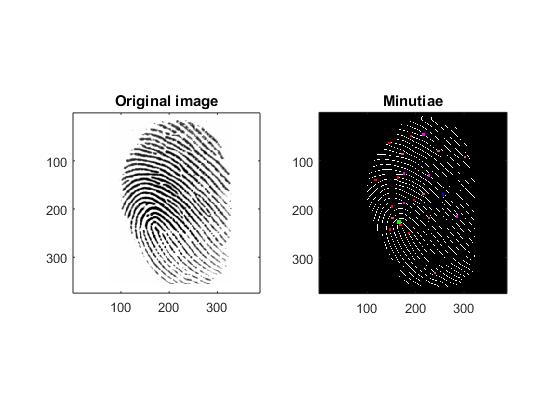
ret=minutiae;

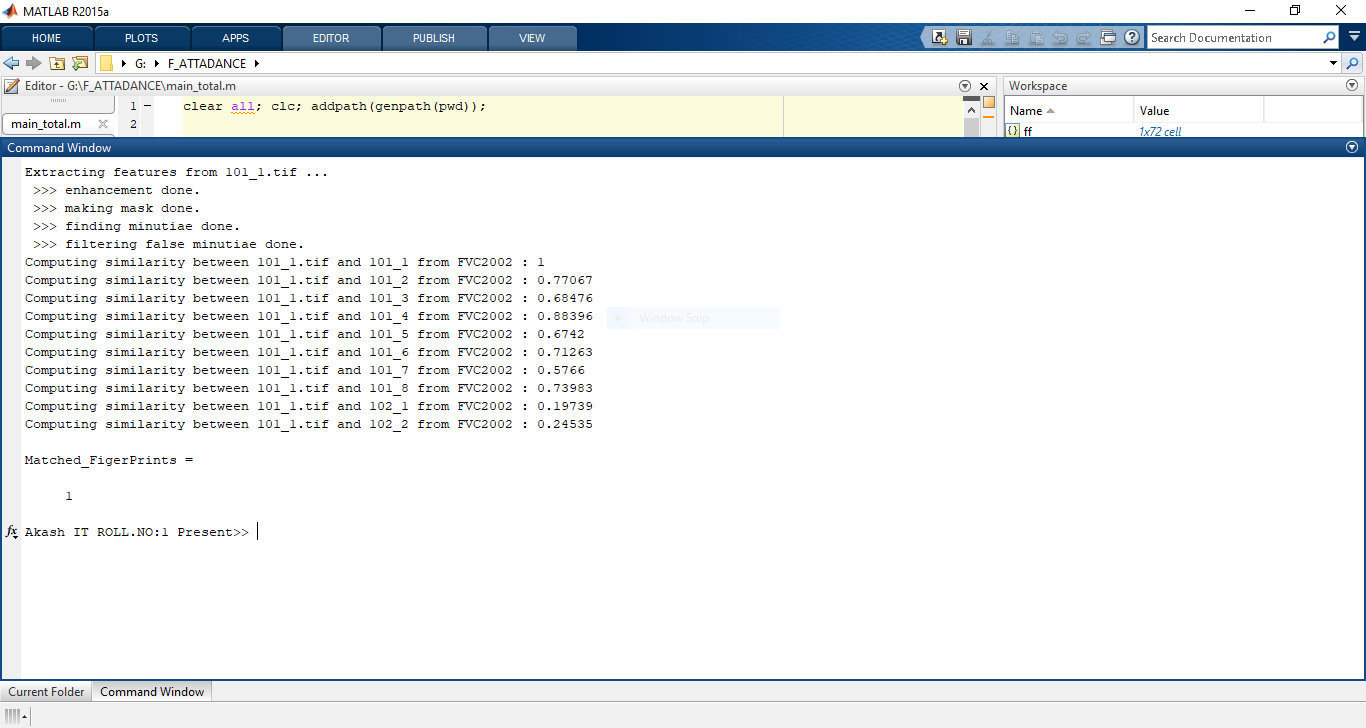
end

**CHAPTER 9**

**SCREEN SHOT**







**CHAPTER 10**

**APPENDIX**

**10.1 GENERAL**

This chapter is about the software language and the tools used in the development of the project. The software used here is Matlab 2015.

**10.2 Matlab 2015b**

**OVERVIEW**

MATLAB (matrix laboratory) is a fourth-generation high-level programming language and interactive environment for numerical computation, visualization and programming. MATLAB is developed by MathWorks. It allows matrix manipulations; plotting of functions and data; implementation of algorithms; creation of user interfaces; interfacing with programs written in other languages, including C, C++, Java, and FORTRAN; analyze data; develop algorithms; and create models and applications. It has numerous built-in commands and math functions that help you in mathematical calculations, generating plots, and performing numerical methods.

**MATLAB's Power of Computational Mathematics**

MATLAB is used in every facet of computational mathematics. Following are some commonly used mathematical calculations where it is used most commonly:

 Dealing with Matrices and Arrays

 2-D and 3-D Plotting and graphics

 Linear Algebra

 Algebraic Equations

 Non-linear Functions

 Statistics

 Data Analysis

 Calculus and Differential Equations

 Numerical Calculations

 Integration

 Transforms

 Curve Fitting

 Various other special functions

**FEATURES OF MATLAB**

 It is a high-level language for numerical computation, visualization and application development.

 It also provides an interactive environment for iterative exploration, design and problem solving.

 It provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

 It provides built-in graphics for visualizing data and tools for creating custom plots.

 MATLAB's programming interface gives development tools for improving code quality, maintainability, and maximizing performance.

 It provides tools for building applications with custom graphical interfaces.

 It provides functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET and Microsoft Excel.

**USES OF MATLAB**

MATLAB is widely used as a computational tool in science and engineering encompassing the fields of physics, chemistry, math and all engineering streams. It is used in a range of applications including:

 Signal processing and Communications

 Image and video Processing

 Control systems

 Test and measurement

 Computational finance

 Computational biology

**REFERENCES**

1. Salil Prabhakar, Sharath Pankanti, Anil K Jain. Biometric Recognition: Security and Privacy Concerns (2003) IEEE Security and Privacy Magazine.

2. Kumar DA, Ummal Sariba Begum. A Comparative Study on Fingerprint Matching Algorithms for EVM (2013) Journal of Computer Sciences and Applications 55-60.

3. Sangeeta Narwal, Daljit Kaur, Comparison between Minutiae Based and Pattern Based Algorithm of Fingerprint Image (2016) I.J. Information Engineering and Electronic Business 2: 23-29.

4. Yager N, Amin A. Fingerprint verification based on minutiae features: a review (2004) Pattern Analysis and Applications 7: 94-113.

5. Md. Mijanur Rahman, Sifat Nur Rahman, Mahbubur Rahman, Firoz Haider. Biometric Student Registration and Verification System (2016) Discovery 52: 2399-2407.

6. Roberto Brunelli. Template Ma tching Techniques in Computer Vision: Theory and Practice. Wiley Publication, USA. 16. Jain AK, Prabhakar S, Hong L, Pankanti S. Filterbankbased fingerprint matching (2000) IEEE Trans Image Process 9: 846-

View.

7. Yasmin Khan. Punch Card Attendance Monitoring System (2010) MS Thesis, Dept. of Computer Science and Information Technology, Jinnah University for Women, Bangladesh.

8. Daniel Ramsbrock, Stepan Moskovchenko, Christopher Conroy. Magnetic Swipe Card System Security A case study of the University of Maryland, College Park.

9. A Ujan, Ismaili IA. Biometric attendance system (2011) The 2011 IEEE/ICME International Conference on Complex Medical Engineering, Harbin Heilongjiang, 499501.

10. Akinduyite CO, Adetunmbi AO, Olabode OO, Ibidunmoye EO. Fingerprint-Based Attendance Management System (2013) Journal of Computer Sciences and Applications 1, no. 5 (2013): 100-105.

11. Khatun AK, Haque MF, Ahmed S, Rahman MM. Design and implementation of iris recognition based attendance management system (2015) International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka.

12. Chintalapati S, Raghunadh MV. Automated attendance management system based on face recognition algorithms (2013) IEEE International Conference on Computational Intelligence and Computing Research, Enathi.