International Journal of Engineering and Emerging Scientific Discovery
ISSN: 2536-7250 (Print): 2536-7269 (Online)
Volume 3, Number 1, March 2018
http://www.casirmediapublishing.com

A Comparative Analysis of a Fingerprint and Facial Recognition Biometric System

Onaifo Frank, Alao Peter Olufemi; Ojo Rotimi & Adeleke Mueez
Department of Electronic/Electrical Engineering
Olabisi Onabanjo University, Ogun State, Nigeria
Email: frankonaifo@oouagoiwoye.edu.ng, excitefranko@yahoo.com
Corresponding Author: Onaifo Frank

ABSTRACT
This paper highlights electronic voting technology which can be deployed to reduce malpractice commonly prevalent during voting exercise. A careful analysis of biological traits involving fingerprint and facial recognition for verification and identification is presented. This involves the development of java programming codes for face recognizer and fingerprint scanner. Facial recognizer makes use of software that compares one image with another image and produces a score that measures how similar the images are to each other. Fingerprint scanner involves the development of a biometric fingerprint system using sensor module, feature extraction module, matcher module and system database module. It is shown here that finger print used as biometric verification is better than using facial recognition.
***Keyword**: Biometric, java programming, Face Recognizer, Fingerprint Scanner, Voting Technology.*

INTRODUCTION

Voting in Nigeria has always been associated with violence with ballot snatching and multiple voting by a single Individual. Fraudulent individuals had forced themselves into office resulting in looting of the common wealth of the people. The consequences are weak economy, underdevelopment, unemployment and poverty of appreciable number of citizenry.

LITERATURE REVIEW

Governance in traditional African setting with several Kingdoms and Empires was hereditary in some places in pre-Colonials times. When several Africa nations gained independence, democratic form of government was embraced but later truncated by military interventionist. The collapse of the military rule led to the later emergence of democratic government across Africa. The challenge is how the people choice gets into office. Credible form of voting can be developed using biological and behavioural traits for verification and identification during election process known as biometric voting system. A low cost development of which trait is more efficient from either using fingerprint or facial recognition led to this paper. Using multiple biometric traits is costly and can be narrowed down to one. Biometrics is basically a method of identifying an individual using any unique physical characteristics of their body. Biometric methods help in ensuring maximum accuracy in identifying a person [1]The traits such as fingerprint, facial recognition, iris recognition, voice recognition, veins and retina recognition are all being

use for electoral processes. Biometric technologies use physical characteristics, such as voice tone or hand shape, to identify people automatically. The systems is developed using extended GUI and MySQL integrated with Digital Persona Fingerprint Reader which stores the bio-data of the voters and candidates to be voted for in an internal database located in the user's computer. When needed, the device interactively sends the voters names and candidates to compatible applications and websites which can be used to perform the voting operation automatically. At this initial session, your biometric characteristic, such as an eye scan, is recorded and linked to this externally-supplied personal information. At future sessions, the computer links you to the previously supplied information using the same physical characteristic. Even if the biometric system works perfectly, the personal data in the computer, such as your voting eligibility, is only as reliable as the original "source" documentation supplied. [2] This Biometric System is a combination of software and hardware. It recognizes fingerprint images and facial recognition ensuring that they match within the database. The database needs to be created and it contains information of the person voting and also the candidates. In order to create the database, programming codes are needed and the Microsoft SQL server is used. The next step is to link the Database to the hardware. The hardware is the Fingerprint Images Scanner Device and the facial capture (camera).

Fingerprint

A fingerprint is an impression of the friction ridges of all or any part of the finger. A friction ridge is a raised portion of the epidermis on the palmer (palm and fingers) or plantar (sole and toes) skin, consisting of one or more connected ridge units friction ridge skin. These ridges are sometimes known as “dermal ridges” or “dermal papillae”. Fingerprint ridges are not continuous as there are a number points at which ridges change and end and these points are called minutia points. The unique identifying features are provided by these minutia points. A raw image is taken from the sensor and algorithms are implemented on the image to enhance it and further extract the minutia points directly from this representation. This procedure provides a much more efficient and reliable result as compared to other methods of fingerprint verification [1].

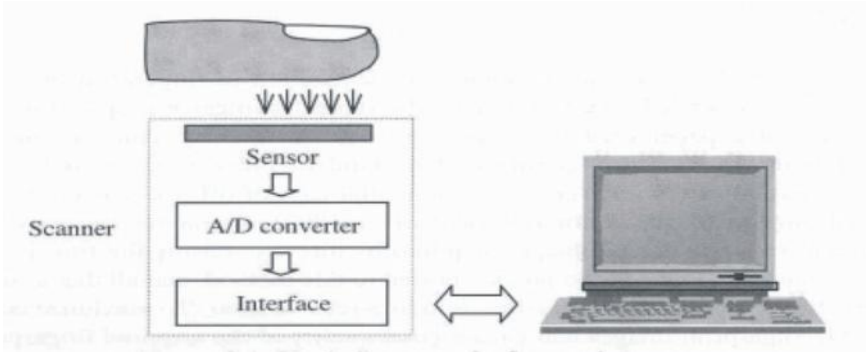


Fig 1.1: General Structure of a Scanner

Facial Recognition

Facial recognition systems are built on computer programs that analyze images of human faces for the purpose of identifying them. The programs take a facial image, measure characteristics such as the distance between the eyes, the length of the nose, and the angle of the jaw, and create a unique file called a "template." Using templates, the software then compares that image with another image and produces a score that measures how similar the images are to each other. The fact is that faces are highly complex patterns that often differ in only subtle ways, and that it can be impossible for man or machine to match images when there are differences in lighting, camera, or camera angle, let alone changes in the appearance of the face itself. [3] There are about 80 nodal points on a human face. Here are a few of the nodal points that are measured by the software: [4] Facial recognition methods may vary, but they generally involve a series of steps that serve to capture, analyze and compare your face to a database of stored images.

A biometric system is designed using the following four main modules:

i. Sensor module: It captures the biometric data of an individual. An example is a fingerprint sensor that images the ridge and valley structure of a user's finger.

ii. Feature extraction module: Here the acquired biometric data is processed to extract a set of salient or discriminatory features. For example, the position and orientation of minutiae points (local ridge and valley singularities) in a fingerprint image are extracted in the feature extraction module of a fingerprint-based biometric system.

iii. Matcher module: Here the features during recognition are compared against the stored templates to generate matching scores.

iv. System database module: It is used by the biometric system to store the biometric templates of the enrolled users. The enrolment module is responsible for enrolling individuals into the

biometric system database. Usually, multiple templates of an individual are stored to account for variations observed in the biometric trait and the templates in the database may be updated over time. [5] Biometrics voting system involves certain procedures which includes the following:

Registration processes

During the registration process, the finger is scanned for each person. We registered using facial recognition and the finger because in the event of an accident or injury to the finger, the facial recognition can be used to identify the individual.

The steps involved in registration process are as follows:

i Initialization

ii Capture, extract and register

iii Store data

Initialization

Run the biometric voting system application that has already been installed on the computer system and make sure all devices such as biometric fingerprint scanner and the webcam are well connected.

Capture, Extract and Register

i. Begin the registration process.

ii. Capture a series of fingerprint scan(s), for each scan:

Store Data

Click register to store the registration process in the database.

2 Identifying/Verifying Processes

Fingerprint recognition involves two types of operation during identifying / verifying process:

i Identification- Comparing a fingerprint against the database of enrolled fingerprints and confirming that the fingerprint is.

ii Verification - Comparing a fingerprint against a specific user's enrolled fingerprint(s) to verify a specific person's identity.

To perform these operations, the application does the following steps:

Initialization

Initialize the library. Discover the available readers and open a connection to a reader.

Capture and Extract

1. Wait for a fingerprint. When a fingerprint is detected, capture the image and create an FID.

2. Extract fingerprint minutiae and create an FMD.

This sequence is exactly the same as for the capture/extraction process during registration.

Identify/Verify

Call the appropriate function to *verify* a specific person OR to *identify* a valid user.

METHODOLOGY

The methods utilize involves:

(i) Hardware

(ii)Software Application design for facial and fingerprint recognition

Hardware

The Hardware used are the Fingerprint Reader or scanner and

Camera. The software is interfaced and integrated with the two (2) hardware – the Digital Persona Fingerprint Reader to detect and scan the fingers for voting exercise or registration through an USB interface. Logitech webcam to also take facial recognition of the person during registration and voting. The java development kit toolbox (JDK) is also applicable in this case to allow the creation of application for the software package or software framework which will be compatible with the hardware platform on the system (laptop). It is something as simple as the implementation of one or more application programming interfaces (APIs) to interface the programming language (Java program) to the hardware which can be communicate with the embedded system.

Software Development

This aspect is the stage at which the software which has been created using a java development kit (JDK) gives readable instructions which directs the computer's processor to perform specific operations.

Face Recognizer

```
packagecom.biometrics.edu.ibile;
importcom.googlecode.javacpp.Loader;
importcom.googlecode.javacv.cpp.opencv_contrib.FaceRecognizer;
importcom.googlecode.javacv.VideoInputFrameGrabber;
importstatic com.googlecode.javacv.cpp.opencv_contrib.createFisherFaceRecognizer;
importcom.googlecode.javacv.cpp.opencv_core;
import static com.googlecode.javacv.cpp.opencv_core.CV_AA;
import static com.googlecode.javacv.cpp.opencv_core.IPL_DEPTH_8U;
import static com.googlecode.javacv.cpp.opencv_core.cvClearMemStorage;
import static com.googlecode.javacv.cpp.opencv_core.cvFlip;
import static com.googlecode.javacv.cpp.opencv_core.cvGetSeqElem;
import static com.googlecode.javacv.cpp.opencv_core.cvLoad;
import static com.googlecode.javacv.cpp.opencv_core.cvPoint;
import static com.googlecode.javacv.cpp.opencv_core.cvRectangle;
import static com.googlecode.javacv.cpp.opencv_imgproc.CV_BGR2GRAY;
import static com.googlecode.javacv.cpp.opencv_imgproc.CV_INTER_AREA;
import static com.googlecode.javacv.cpp.opencv_imgproc.cvCvtColor;
import static com.googlecode.javacv.cpp.opencv_imgproc.cvResize;
importcom.googlecode.javacv.cpp.opencv_objdetect;
importstatic com.googlecode.javacv.cpp.opencv_o
```

```
bjdetect.CV_HAAR_DO_CANNY_PRU
NING;
import static
com.googlecode.javacv.cpp.opencv_o
bjdetect.cvHaarDetectObjects;
importcom.googlecode.javacv.cpp.vid
eoInputLib;
import java.awt.Graphics2D;
importjava.awt.GraphicsConfiguratio
n;
importjava.awt.GraphicsDevice;
importjava.awt.GraphicsEnvironment
;
importjava.awt.Image;
importjava.awt.image.BufferedImage;
importjava.io.File;
importjava.io.FileInputStream;
importjava.io.IOException;
importjava.io.ObjectInputStream;
importjava.util.HashMap;
importjavax.imageio.ImageIO;
importjavax.swing.Icon;
importjavax.swing.ImageIcon;
importjavax.swing.JLabel;
importjavax.swing.JOptionPane;
public class FaceRecognizerUI
extends javax.swing.JDialog {
publicint FISHERFACES_NUM=10;
public double
FISHERFACES_THRESHOLD=500.5;
  //-----------------------------------------
---------------------------------------
public final String
FACE_CLASSIFIER_RESOURCE =
"resources/haarcascade_frontalface_al
t.xml";
public final String
EYE_CLASSIFIER_RESOURCE =
"resources/haarcascade_eye.xml";
private final
opencv_core.CvMemStoragefaceStora
ge;
private final
opencv_core.CvMemStorageeyeStora
ge;
private final WebCam webcam;
private final
opencv_objdetect.CvHaarClassifierCa
scadefaceClassifier;
private final
opencv_objdetect.CvHaarClassifierCa
scadeeyeClassifier;
private final
ObjectMetricseyeMetrics[];
private final
ObjectMetricsfaceMetrics;
privateint snapped;
private final JLabel[] encs;
privateboolean ready;
private final boolean register;
  /**
   * Creates new form FaceRecognizer
   *
   * @param parent
   * @param modal
   * @param register
   */
publicFaceRecognizerUI(java.awt.Fra
me parent, boolean modal, boolean
register) {
super(parent, modal);
this.register = register;
initComponents();
FileInputStream is = null;
try {
HashMap<String, Number>setting ;
is = new FileInputStream(new
File("resources/settings.dat"));
ObjectInputStreamois = new
ObjectInputStream(is);
setting = (HashMap<String,
Number>) ois.readObject();
ois.close(); FISHERFACES_NUM =
setting.get(FisherFacesSettings.FISHE
RFACES_NUM).intValue();
```

```
FISHERFACES_THRESHOLD=
setting.get(FisherFacesSettings.FISHE
RFACES_THRESHOLD).doubleValue
();
      } catch (Exception ex) {
System.err.println(ex);
      } finally {
try {
is.close();
          } catch (Exception ex) {
          }}
if (!register) {
save.setText("Identify");
pic2.setVisible(false);
pic3.setVisible(false);
pic4.setVisible(false);
pic5.setVisible(false); }
setLocationRelativeTo(parent);
reCapture.setEnabled(false);
save.setEnabled(false);
encs = new JLabel[5];
encs[0] = pic1;
encs[1] = pic2;
encs[2] = pic3;
encs[3] = pic4;
encs[4] = pic5;
      // Load object detection
Loader.load(opencv_objdetect.class);
      // Construct classifiers from xml.
faceClassifier                    =
loadHaarClassifier(FACE_CLASSIFIE
R_RESOURCE);
eyeClassifier                     =
loadHaarClassifier(EYE_CLASSIFIER
_RESOURCE);
faceStorage                       =
opencv_core.CvMemStorage.create();
eyeStorage                        =
opencv_core.CvMemStorage.create();
eyeMetrics = new ObjectMetrics[2];
eyeMetrics[0] = new ObjectMetrics();
eyeMetrics[1] = new ObjectMetrics();
faceMetrics = new ObjectMetrics();
getCameraList();
webcam = new WebCam();}
   //                          <editor-fold
defaultstate="collapsed"
desc="Generated          Code">//GEN-
BEGIN:initComponents
private void initComponents() {
      enc1 = new javax.swing.JPanel();
      pic1 = new javax.swing.JLabel();
base = new javax.swing.JPanel();
enc = new javax.swing.JPanel();
      base2 = new javax.swing.JLabel();
      enc2 = new javax.swing.JPanel();
      pic2 = new javax.swing.JLabel();
      enc3 = new javax.swing.JPanel();
      pic3 = new javax.swing.JLabel();
      enc4 = new javax.swing.JPanel();
      pic4 = new javax.swing.JLabel();
      enc5 = new javax.swing.JPanel();
      pic5 = new javax.swing.JLabel();
      jPanel1          =          new
javax.swing.JPanel();
startCamera          =          new
javax.swing.JButton();
cameras              =          new
javax.swing.JComboBox();
      jLabel1          =          new
javax.swing.JLabel();
      jSeparator1      =          new
javax.swing.JSeparator();
      jPanel2          =          new
javax.swing.JPanel();
reCapture            =          new
javax.swing.JButton();
save = new javax.swing.JButton();
      jSeparator2      =          new
javax.swing.JSeparator();
setDefaultCloseOperation(javax.swin
g.WindowConstants.DISPOSE_ON_C
LOSE);
addWindowListener(new
java.awt.event.WindowAdapter() {
```

```
public void windowClosing(java.awt.event.WindowEventevt) {
closing(evt);}});

enc1.setBorder(javax.swing.BorderFactory.createMatteBorder(2, 2, 2, 2, new java.awt.Color(153, 153, 153)));
enc1.setMaximumSize(new java.awt.Dimension(120, 147));
enc1.setMinimumSize(new java.awt.Dimension(120, 145));
enc1.setOpaque(false);
javax.swing.GroupLayout enc1Layout = new javax.swing.GroupLayout(enc1);
enc1.setLayout(enc1Layout);
enc1Layout.setHorizontalGroup(
enc1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(pic1, javax.swing.GroupLayout.DEFAULT_SIZE, 116, Short.MAX_VALUE)
enc1Layout.setVerticalGroup( );
```

Finger Print Scanner

```
packagecom.biometrics.edu.ibile;
importcom.digitalpersona.uareu.Engine;
importcom.digitalpersona.uareu.Fid;
importcom.digitalpersona.uareu.Fmd;
importcom.digitalpersona.uareu.Reader;
importcom.digitalpersona.uareu.ReaderCollection;
importcom.digitalpersona.uareu.UareUException;
importcom.digitalpersona.uareu.UareUGlobal;
importjava.awt.Image; 98
importjava.awt.image.BufferedImage;
importjava.io.DataInputStream;
importjava.io.DataOutputStream;
importjava.io.File;
importjava.io.FileInputStream;
importjava.io.FileNotFoundException;
importjava.io.FileOutputStream;
importjava.io.IOException;
importjava.util.logging.Level;
importjava.util.logging.Logger;
importjavax.swing.ImageIcon;
importjavax.swing.JOptionPane;
/**
 *
 * @author workhouse
 */
public class FingerPrintScanner extends javax.swing.JDialog {
privateReaderCollectionrc;
private Reader reader;
privateboolean cancel;
private Fid image;
privateFmdfmd;
private final boolean register;
privatebooleanisFingerprintMatched;
   /**
    * Creates new form FingerPrintScanner
    *
    * @param parent
    * @param modal
    * @param register
    */
publicFingerPrintScanner(java.awt.Frame parent, boolean modal, boolean register) {
super(parent, modal);
this.register = register;
initComponents();
if(!register){
save.setText("Identify"); }
setLocationRelativeTo(null);
refreshScanners();    }
```

```
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents
private void initComponents() {
    jPanel1 = new javax.swing.JPanel();
scan = new javax.swing.JButton();
scanners = new javax.swing.JComboBox();
    jLabel1 = new javax.swing.JLabel();
    jSeparator1 = new javax.swing.JSeparator();
base = new javax.swing.JPanel();
scroller = new javax.swing.JScrollPane();
pic = new javax.swing.JLabel();
save = new javax.swing.JButton();
    jLabel3 = new javax.swing.JLabel();
    jScrollPane1 = new javax.swing.JScrollPane();
    jTextArea1 = new javax.swing.JTextArea();
setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
setTitle("Fingerprint Scanner");
jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder(""));
scan.setText("Scan");
scan.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEventevt) {
scanActionPerformed(evt);  }});

scanners.setFocusable(false);
jLabel1.setText("Select Fingerprint Scanner");
javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
addGroup(jPanel1Layout.createSequentialGroup()
 addContainerGap()
addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(scanners, 0,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel1)
            .addGap(0, 0, Short.MAX_VALUE))

.addComponent(jSeparator1)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()
            .addGap(150, 150, 150)
            .addComponent(scan,
javax.swing.GroupLayout.PREFERRED_SIZE, 137,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap()))
```

Performance Indicators

The two performance indicators used in this paper are False Accept Ratio (FAR) and False Reject Ratio (FRR).

False Accept: . The FAR normally states, either in a percentage or a fraction, the probability of someone else matching as you. FAR is defined by the formula:

*FAR=FR/N*100*

Where FA is the number of false accept and N is the total number of verification.

False Reject**:** When the acquired biometric signal is of poor quality, even a genuine user may be rejected during authentication. This form of error is labeled as a 'false reject'. If you fail to match against your own template, then you have been falsely rejected. The probability of this happening is referred to as the false rejection rate, or FRR.

*FRR=FR/N*100*

Where FR is the number of false reject and N is the total number of verification.

RESULTS

Two sets of students were verified. The first sets are 50 students and the second sets are 40 students. The total numbers of verified first set students are fifty (50). The success rate of 96% was obtained from the test carried out for fingerprint and a success rate of 75% for facial recognition.

The total number of verified students is N; the numbers of false reject (FR) i.e. students that are initially registered by the application but their fingerprints was rejected during voting processing is 2.

<u>For Fingerprint</u>

FAR=FR/N*100

Where, N is the number of verified student,

FR is the false rejected,

FAR is the false acceptance rate.

N= 50; FR = 2

FAR= 2/50*100= 4%.

<u>For Facial Recognition.</u>

FAR=FR/N*100

N= 50; FR = 12

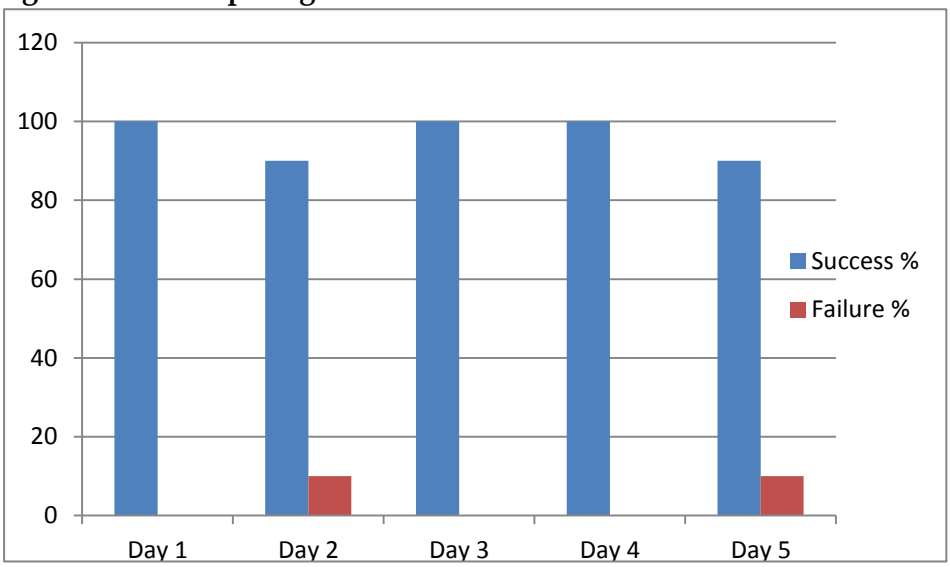
FAR= 12/50*100= 24%.

The results of the test for fingerprint are shown below in the chart (Table 1)

Table 3.1: Comparison of Success and Failure Rate using Fingerprint.

Days	**1**	**2**	**3**	**4**	**5**
Success %	100	90	100	100	90
Failure %	0	10	0	0	10

Fig 3.1 Chart comparing the Success and Failure rate



The results of the test for facial recognition are shown below.

Table 3.2: Comparison of Success and Failure Rate for facial Recognition

Days	**1**	**2**	**3**	**4**	**5**
Success %	80	90	70	100	80
Failure %	20	10	30	0	20

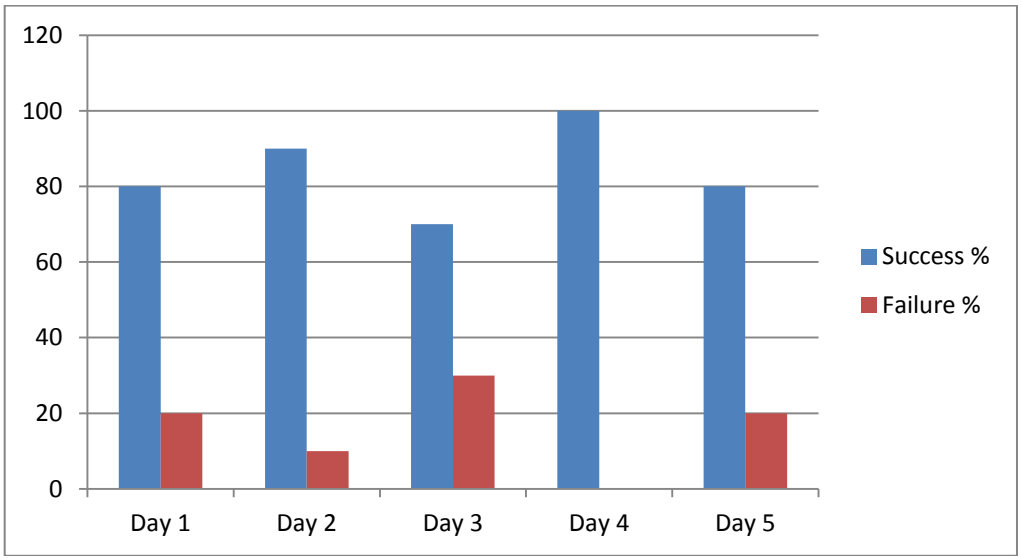


Fig 3.2 Chart comparing the Success and Failure rate for Facial Recognition.

Table 3.3: Comparison of fingerprint and facial recognition Authentication for first set of students

Authentication	First	Second	Third	Fourth	Fifth
Fingerprint %	95	100	-	-	-
Facial recognition %	26	32.4	24	10.5	29.4

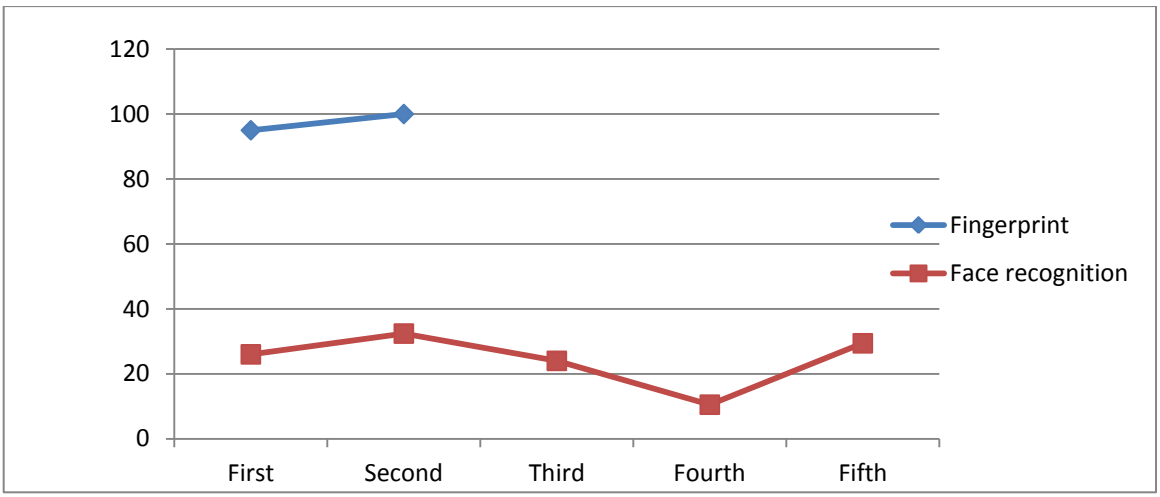


Fig 3.3: Graph showing the authentication during registration between fingerprint and facial recognition for the first set of students.

A second set of 40 students were verified and the results are tabulated below:

Table 3.4: Finger Print and Facial Recognition data during registration

	Finger print 1st	**2nd**	**3rd**	**4th**	**Facial Recognition 1st**	**2nd**	**3rd**	**4th**	**5th**
1	X					X			
2	X					X			
3	X					X			
4		X					X		
5	X						X		
6	X								X
7	X						X		
8	X				X				
9	X				X				
10		X				X			
11	X					X			
12	X					X			
13	X					X			
14	X					X			
15	X								
16	X				X				
17	X				X				
18	X				X				
19	X				X				
20	X						X		
21	X							X	
22	X				X				
23	X					X			

24	X					X			
25		X				X			
26	X						X		
27	X				X				
28	X				X				
29	X				X				
30	X						X		
31	X						X		
32	X					X			
33	X					X			
34	X					X			
35	X				X				
36	X				X				
37	X					X			
38	X				X				
39	X								X
40	X						X		

Table 3.5: Comparison of fingerprint and facial recognition Authentication for the second set of students

Authentication	First	Second	Third	Fourth	Fifth
Fingerprint %	92.5	100	-	-	-
Facial recognition %	32.5	55.5	66.6	50	50

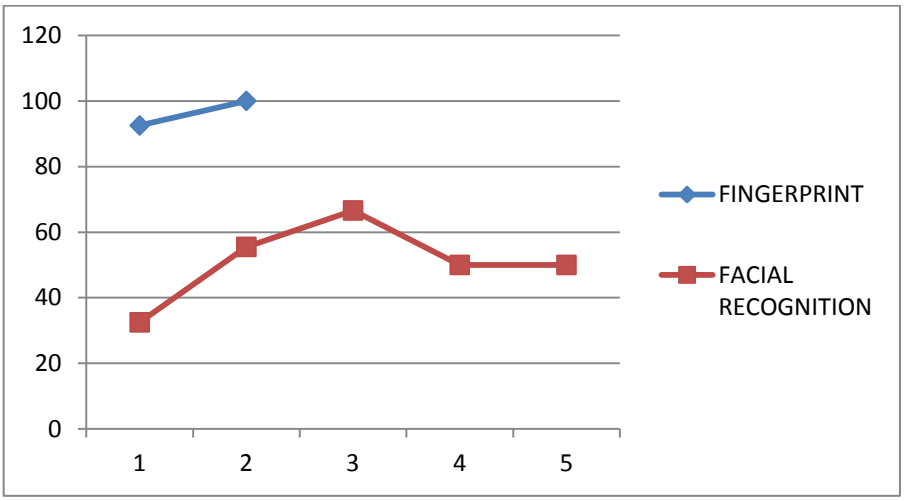


Fig3.4: Graph showing the authentication rate during registration between fingerprint and facial recognition for the second set students

OBSERVATION

The result for authentication in both cases shows that finger print is better than facial recognition biometric system.

DISCUSSION

The test results shows that the system is effective and it has a fast response. There was no false identification of students, few cases of false reject which was later accepted and only pre-registered students were authenticated. The matrixes of the identified students were enrolled for voting automatically.

CONCLUSION

It can be seen that the success rate of finger print used as biometric verification is better than using facial recognition. Using finger print as a means of authentication should be encouraged as it has at least a success rate of 20% higher than facial recognition. During the enrolment and verification stages, making use of fingerprint is more accurate compare to that of facial recognition. This is because the FAR (False acceptance rate) for the first set of student for facial recognition is higher than that of fingerprint recognition. It is also seen that the authentication for the first set of 50 students and second set of 40 students for facial recognition is very unpredictable compared to the fingerprint which is more predictable. The only drawback using fingerprint as a means of verification include either when the individual is amputated or have hand accidents where the fingerprint markings are distorted.

REFERENCES

[1] support.sumhr.com/support/articles/88151-what-is-a-biometric-fingerprint-reader-and-rfid-machine

[2] "Guide to Fingerprint Recognition", DigitalPersona, Inc. 720 BayRoad Redwood City, CA 94063, USA, http://www.digitalpersona.com

[3] Facial Recognition, hi-tech Security Solution Magazine, May 2005, American Civil Liberties Union

[4] Jain A.K, Maio D., Maltoni D., and Prabhakar S. (2003): Handbook of Fingerprint Recognition, Springer, New York, 2003

[5] Voting System Using Fingerprint Recognition, Hemlata Patel and PallaviAstrodia, Department of Computer Science and Engineering, Jawharlah Institute of Technology, Boravan, Khargone (MP.)