Fingerprint Based Voting

Akshat Bheda[1], Udit Intwala[2], Rahul Ghimire[3], Prof. Sanketi Raut[4]

,[1,2,3] Student, [4]Assistant Professor, Department of Information Technology Engineering, Universal college of Engineering, Maharashtra, India

---***---

Abstract - *The idea of this paper arises by taking the present Covid crisis into consideration, as public gathering is prohibited/limited and risky. If people gather for voting in such situations there it may give rise to vast spread of virus instead of keeping it under control. People need to maintain long queues which costs them a lot of time and energy. The main motive of this project is to offer reliability and prohibit the challenges that are in the conventional balloting system. The simple idea of this project is the user has to register on the app, submit their fingerprint which will be processed using SIFT algorithm and complete their voting. So, by using the idea of this project people can avoid long queues and also avoid gathering and can vote easily from their homes without risking their lives keeping the Covid crisis under control.*

***Key Words*: Fingerprint, Voting, Java, Firebase, Python, Android, SITF.**

1.INTRODUCTION

These days the mechanical device has become an efficient toll compared with traditional paper-based voting techniques. Thus, we decided to style a system to beat the already existing electoral system. the most scheme of this project is to possess safer, no duplication of votes and declare the results as early as possible.

Requirements are often classified into functional and security requirements. Reviewing these requirements as they affect e-voting system, the important factors are:

- Authentication
- Integrity
- Confidentiality
- Non-Repudiation
- Accuracy
- Flexibility
- Auditability
- Eligibility

2. LITERATURE SURVEY

We have studied some of available fingerprint recognition system and how it can be implemented.

Biometric Based Mobile Voting

This paper shows various methods voting can be carried on mobile devices. It creates simple application to perform voting with help of biometric stored in government database. Accessing it and based on that voting is carried out.

Fingerprint Recognition System: Design & Analysis Fingerprint are of large variety. Some has more no of ridges and curve than others, some individual has almost no fingerprint. It is really important to design a system which is capable to counter almost all possible scenarios. There are many techniques developed in recent years from optical to ultrasonic fingerprint reader.

3. PROBLEM STATEMENT

Current system majority of them use ballot system with digital system. A more hybrid method to ensure the voting is done without cheat. It is said that no system is complete system. Every system has major or minor flaw which is exploit for unfair advantages. We cannot make a fully secure system, if something is working it does not mean it can't be better.

Problem arise as such voting system needs each individual to physically present at the time of voting. Which is true as to conform identity, it is not possible always as many individuals has to face difficulty especially for disabled individual.

Also, at time when there is a global catastrophe where we are prohibited to gather in crowd and perform mass voting. This project is to tackle exactly such situations. It makes voting literally at your doorsteps. Proposed system is privacy protected authentication system in which database of biometric identities is created using fingerprint templates.

4. IDEA OF IMPLEMENTATION

Fingerprint recognition system uses android camera to capture image of voter's finger and then pass it to the database to match the fingerprints in order to authenticate the voter.

It works along with built-in android biometrics authentication to enhance the authentication of voting system.

4.1 TECHNILOGY USED

4.1.1. Android Studio

Android Studio is an Integrated Development Environment (IDE) for android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, It offers many great features that help increase your productivity while building android apps.

4.1.2. Java

Java may be a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable.

4.1.3. Firebase

Google Firebase may be a Google-backed application development software that permits developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and merchandise experiment.

Firebase offers a number of services, following are used:

• Authentication, Realtime database, Firebase Storage, Firebase Analytics

4.1.4. OpenCV

This paper describes a system created using SIFT technology to identify fingerprint and with help of it we can verify user fingerprint. It allows user to verified and authenticated.

It can be used and implemented in multiple ways, It provide freedom to create dynamic system with solid security setup. We can replace current version of working system to much faster and more secure version.

4.1.5. Python

It is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of serious indentation. Its language constructs also as its object-oriented approach aim to assist programmers write clear, logical code for little and large-scale projects.

4.2 SYSTEM DESIGN

The motive of our system design is to create a procedural solution that can fulfil the functional requirements for secure scheme for e-democratic governance. The architecture that describes the components and interfaces of the entire system derived from system requirements are modelled to solve identified problem.

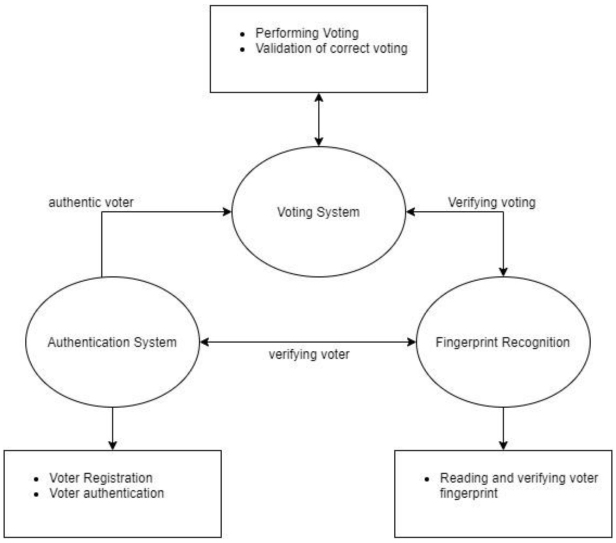


Figure 1: System Design

The first part of this work is to create database of different fingerprints of voter, who are eligible for voting in election. Fingerprint sensor is used for scanning of finger and generates the templates. After creating a database, the complete process is classified in three parts.

First one is identification in which, fingerprint of the voter is matched with the previously stored templates in database.

Second part is verification, in which whether the voter is valid or not is to be checked. If one of the templates matches with the finger of voter then, the voter is eligible for voting. Repeated voting is also monitored in this process, which is not allowed by the same candidate.

Third part of the work is voting, in which, voter will choose the option, which is available for voting. This is developed using java, python using android studio IDE.

4.3 ALGORITHM

4.3.1. SITF [5]

Scale Invariant Feature Transformation (SIFT). The SIFT algorithm proposed by Lowe[1] is an approach for extracting distinctive invariant features from images. It has been successfully applied to a spread of computer vision problems supported feature matching, which incorporates visual perception, pose estimation, image retrieval, etc. Since the SIFT feature points have already showed their efficacy within the visual perception field, there are high expectation to be used in fingerprint verification.

This method extracts distinctive features from images, which can be used to perform reliable matching between different images. There are four major stages of computation used for generating the SIFT features: Scale-space extrema detection, Key point localization, Orientation assignment and the Key point descriptor.

4.3.2. Implementation

The Process start with taking clear image of voter fingerprint, it is then stored into firebase with respect to the authenticated voter.

After performing voting the voter will submit his fingerprint which is send to algorithm, It then access stored fingerprint image from the firebase database storage.

It then performs matching of fingerprint as follows:

1. Taking Initial Fingerprint image:

```
(sample = cv2.imread("Input Image"))
```

2. Taking Stored Fingerprint image from Database:

```
(for [file stored in Database ("Image from Database")]
```

3. Extracting Keypoints from both the images:

```
(keypoints_1,descriptors_1=sift.detectAndCompute(sample, None)

keypoints_2,descriptors_2=sift.detectAndCompute(fingerprint_image, None))
```

4. Matching and Comparing Key points of both images:

```
(matches1=cv2.FlannBasedMatcher({'algorithm': 1, 'trees': 10}, {}).knnMatch(descriptors_1, descriptors_2, k=2)

bf = cv2.BFMatcher(cv2.NORM_L1, crossCheck=False)

matches=bf.match(descriptors_1,descriptors_2)

matches = sorted(matches, key=lambda x: x.distance)

for p, q in matches1:

if p.distance < 0.1 * q.distance: match_points.append(p)

keypoints = 0

if len(keypoints_1) < len(keypoints_2): keypoints = len(keypoints_1)

else: keypoints = len(keypoints_2))
```

5. Displaying result with best possible match:

```
(print("BEST MATCH: " + filename)
print("SCORE: " + str(best_score))

result = cv2.drawMatches(sample, kp1, image, kp2, mp, None)

plt.rcParams['figure.figsize'] = [14.0, 7.0]
plt.title('Best Matching Points') plt.imshow(result)
plt.show())
```

4.3.3. Result

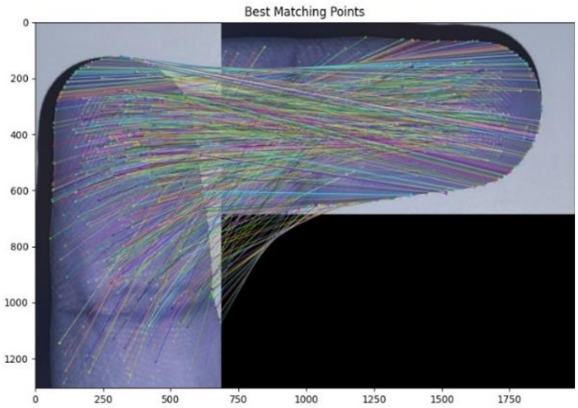
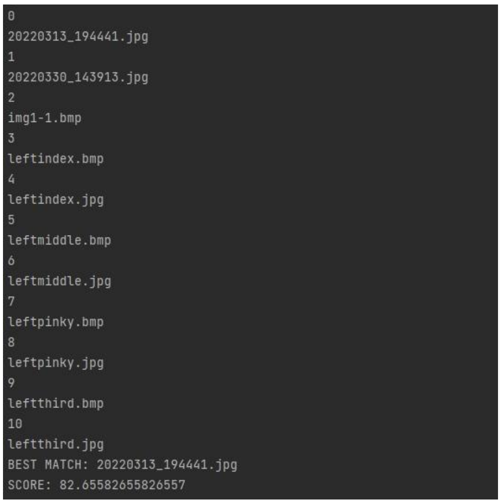
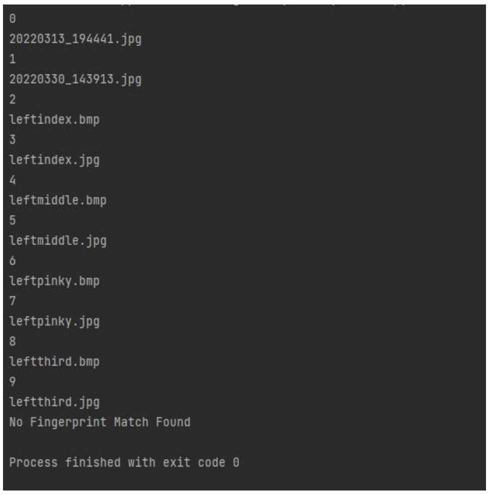


Figure 2: Test 1 Image Output



```
0
20220313_194441.jpg
1
20220330_143913.jpg
2
img1-1.bmp
3
leftindex.bmp
4
leftindex.jpg
5
leftmiddle.bmp
6
leftmiddle.jpg
7
leftpinky.bmp
8
leftpinky.jpg
9
leftthird.bmp
10
leftthird.jpg
BEST MATCH: 20220313_194441.jpg
SCORE: 82.65582655826557
```

Figure 3: Test 1 Output



```
0
20220313_194441.jpg
1
20220330_143913.jpg
2
leftindex.bmp
3
leftindex.jpg
4
leftmiddle.bmp
5
leftmiddle.jpg
6
leftpinky.bmp
7
leftpinky.jpg
8
leftthird.bmp
9
leftthird.jpg
No Fingerprint Match Found

Process finished with exit code 0
```

Figure 4: Test 2 Output

Fingerprint Test Table		
Fingerprints		Result %
Fingerprint 1	Fingerprint 2	82.65
Fingerprint 1	None	No Match

5. CONCLUSIONS

This paper describes a system created using SIFT technology to identify fingerprint and with help of it we can verify user fingerprint. It allows user to verified and authenticated.

It can be used and implemented in multiple ways, it provides freedom to create dynamic system with solid security setup. We can replace current version of working system to much faster and more secure version.

REFERENCES

[1] David G. Lowe, "Object recognition from local scale-invariant features", International Conference on Computer Vision, Corfu, Greece, pp. 1150- 1157, 1999

[2] D.G.Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Jounal of Computer Vision, 60(2), 91-110, 2004

[3] P. Gnanasivam, S. Muttan, "An efficient Algorithm for fingerprint preprocessing and feature extraction" Procedia Computer Science 2 (2010) 133–142.

[4] Donovan Gentles, Dr. Suresh Sankaranarayanan Adjunct Professor, "Biometric Secured Mobile Voting" 2011 IEEE.

[5] Ru Zhou, SangWoo Sin, Dongju Li, Tsuyoshi Isshiki and Hiroaki Kunieda, "Adaptive SIFT-Based Algorithm for Specific Fingerprint Verification" 2011 IEEE.

[6] Kavita Rathi and Sudhir Sawarkar, "Finger Print Matching Algorithm for Android" International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 2 Issue 10, October – 2013.

[7] G.A. Bahgat, A.H. Khalil, N.S. Abdel Kader, S. Mashali, "Fast and accurate algorithm for core point detection in fingerprint images" Egyptian Informatics Journal (2013) 14, 15–25

[8] Dibyendu Nath, Saurav Ray, Sumit Kumar Ghosh, "Fingerprint Recognition System: Design & Analysis" Conference Paper · January 2011 Research Gate 2014.

[9] Masao Yamazaki, Dongju Li, Tsuyoshi Isshiki and Hiroaki Kunieda, "SIFT-based Algorithm for Fingerprint authentication on smartphone". 2015 IEEE.

[10] Farah Dhib Tatar, "fingerprint recognition algorithm" David C. CCSEIT, AIAP, DMDB, ICBB, CNSA – 2017.

[11] Khadija Hasta, Aditya Date, Aparna Shrivastava, Prajakta Jhade, S. N. Shelke, "Fingerprint Based Secured Voting".

[12] Noorazlina Mohamid Salih, "A Design of Voting System by Using an Android Apps for Fisherman.