MEDIHELP - FINGERPRINT AUTHENTICATION TO RECORD MEDICAL HISTORY

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

NITESH TYAGI (191292)

SILKY AGARWAL (191531)

UNDER THE SUPERVISION OF

Dr. Shweta Pandit

&

Prof. Dr. Vivek Kumar Sehgal

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh

Candidate’s Declaration

I hereby declare that the work presented in this report entitled **“Medi Help-Fingerprint Authentication to Record Medical History”** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2022 to December 2022 under the supervision of **Dr. Shweta Pandit,** Assistant Professor(SG), Department of Electronics and Communication Engineering and **Prof. Dr. Vivek Kumar Sehgal**, Professor and Head, Department of Computer Science & Engineering and Information Technology. I also authenticate that I have carried out the above mentioned project work under the proficiency stream **Data Science**.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

(Student Signature)

Nitesh Tyagi,191292

(Student Signature)

Silky Agarwal,191531

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

(Supervisor Signature)

Dr. Shweta Pandit

Assistant Professor(SG)

Department of Electronics and
Communication Engineering

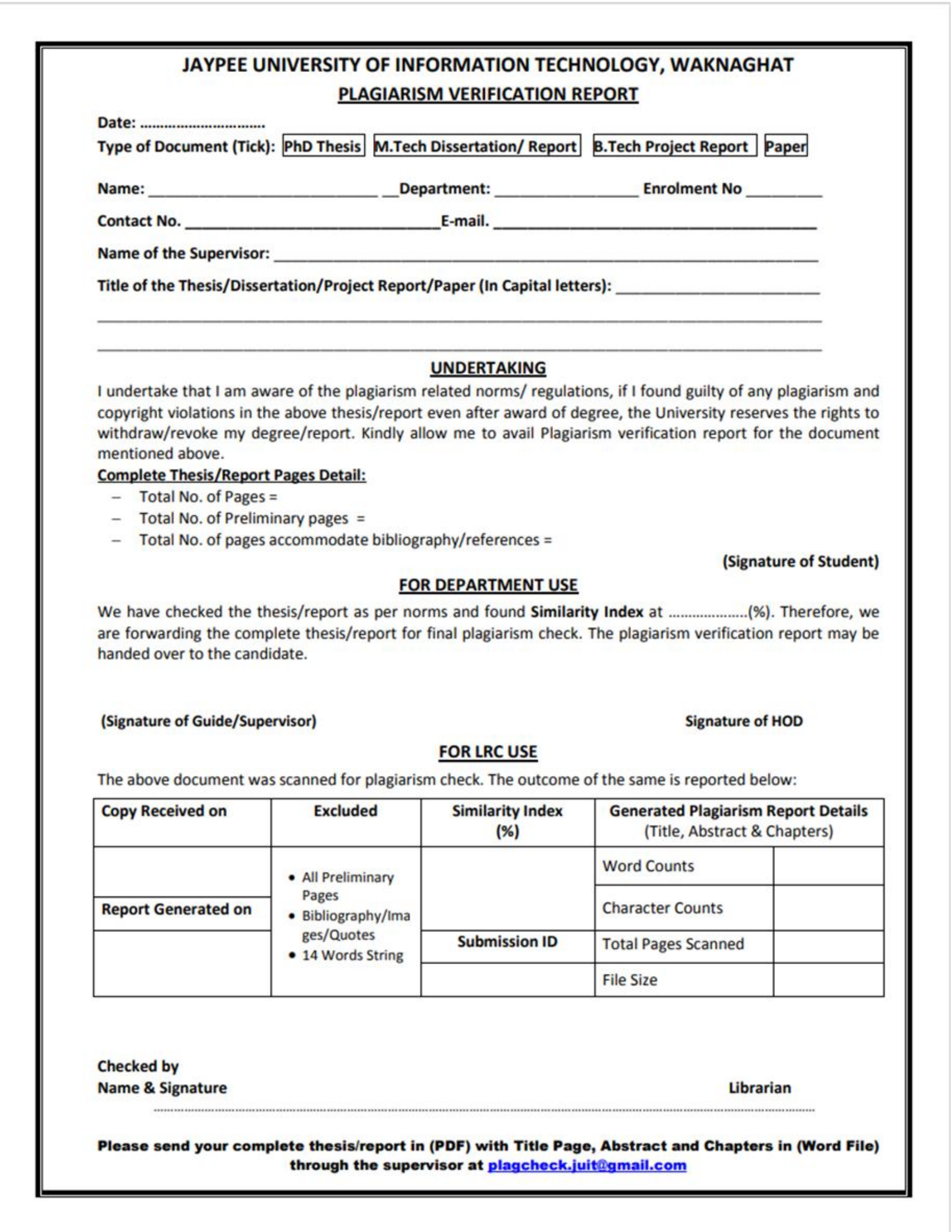
(Co-supervisor Signature)

Prof. Dr. Vivek Kumar Sehgal

Professor and Head

Department of Computer Science
and Engineering and Information
Technology

PLAGIARISM CERTIFICATE



JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ________________________ __**Department:** _______________ **Enrolment No** ________

Contact No. ____________________________**E-mail.** ___________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	**Excluded**	**Similarity Index (%)**	**Generated Plagiarism Report Details** (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

I would like to express my profound gratitude to Prof. Dr. Vivek Kumar Sehgal (Professor and Head, Fellow IEI, SM-IEEE, SM-ACM), of the Computer Science and Information Technology department, and Prof. (Dr.) Ashok Kumar Gupta (Dean) of the Jaypee University of Information Technology for their contributions to the completion of my project titled Medi Help - Fingerprint Authentication to Record Medical History.

I would like to express my special thanks to our mentor Dr. Shweta Pandit for the time and efforts she provided throughout the year. Your useful advice and suggestions were helpful to me during the project's completion. In this aspect, I am eternally grateful to you.

I would like to acknowledge that this project was completed entirely by me and not by someone else.

Nitesh Tyagi

(191292)

Silky Agarwal

(191531)

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Abbreviations	Meaning
HCP	Healthcare Professionals
FLANN	Fast Library for Approximate Nearest Neighbours
SIFT	Scale-Invariant Feature Transform
PIN	Personal Identification Number
ANN	Artificial Neural Network
SVM	Support Vector Machine
KNN	K- Nearest Neighbour
WNN	Weightless Neural Networks
CNN	Convolutional Neural Networks
AU	Authorized User
ECG	Electrocardiogram
LSTM	Long Short-Term Memory
2FA	2 Factor Authentication
AWGN	Additive White Gaussian Noise
RGB	Red, Green, Blue
FC	Fibre Channel
SMS	Short Message Service

LIST OF FIGURES

ABSTRACT

A biometric system is an ever-evolving technology utilised in forensics, secured areas, and security systems, among other areas. The most widely used biometric authentication system is one that recognises fingerprints. The earliest and most popular biometric authentication method is fingerprint identification.

Biometric user identification systems are rapidly evolving in society as a result of recent advances in sensor technology, machine learning, lower processing costs, and portable computer devices' widespread utilisation portable computer devices.

Modern approaches to fingerprint verification, built on advanced machine learning algorithms, are unable to avoid retaining either explicit user biometric data or trained-classifier details, exposing users' credentials vulnerable to fraud.

Concerns about fingerprint biometric security have grown in recent years. Technology developments in bypassing and hacking techniques are the cause of this issue. The demand for a more secure platform for identification has been prompted by this.

Our project focuses on solving this issue using fingerprint recognition using OpenCV and proposes a suitable method for this issue. Fingerprint recognition is performed by identifying the keypoints and descriptors and matching those with the test data

CHAPTER 1: INTRODUCTION

1.1. Introduction

An individual’s physical characteristics or behavioral traits can be used to securely recognize individuals and grant permission to resources like devices, networks, or digital information using biometrics.

These biometric technologies include things like fingerprints, facial characteristics, vocalizations, and typing rhythm, to name a few. Every one of these characteristics is assumed to be unique to an individual, and they can be combined to provide a more accurate identification.

Today, many aspects of daily life, including building entrance and computer login, require biometric technologies. One of the most common techniques for personal identification is fingerprint detection throughout every biometric system. A significant portion of the populace all around the world accepts fingerprint recognition as a quick, secure, and simple method of identifying an individual. The most intriguing and historically significant personal identification used to identify people is their fingerprint. Early in the 20th century, law enforcement organizations formally recognised fingerprints as reliable identification documents.

Biometrics can transform important parts of healthcare and protect individuals who are vulnerable. By using biometric authentication, healthcare can be enhanced, people and patients can be kept safe, and deception may be halted. When used in various healthcare systems, such as clinics, local hospitals, health coverage systems, these advantages can be fairly considerable and have a dramatic influence.

Medical practitioners review clinical notes for a number of reasons, such as research, specific potential treatments, insurance compensation or customer billing, and assessments. However, one of the most sensitive types of personal information is medical records, thus protecting them requires particular care.

Most organizations and healthcare facilities have strict protocols in place to ensure that HCPs can only access records that they have a legitimate reason to look at. These protocols, however, usually rely on outdated, well-known, insecure authentication methods, such as PINs and passwords. As a result, they are open to fraud, and invasion.

As many healthcare facilities attempt to improve the digitization of information, particularly medical records, the susceptibility of outdated norms is a moral, economical and ethical challenge. Medical facilities which accidentally allow unauthorized people to access records or sensitive information run the danger of being severely fined under data privacy regulations or facing legal action from parties seeking retribution. Additionally, the negative publicity that results may make patients and medical professionals lose faith in them.

Secure password-based authentication designs instead save the cryptographic hashes of the passwords associated with each username rather than the actual passwords themselves. This accepted paradigm is supported by the assumption that no user, even the administrator, might succeed to impersonate another user with less privileges.

Fingerprint biometric authentication reduces this danger because, unlike PINs and passwords, fingerprints are one-of-a-kind and can't be misplaced, lost, or duplicated. Furthermore, fingerprints cannot be duplicated or stolen since they are kept securely on smart ID cards to obtain information rather than globally on servers or in devices. Therefore, biometric authentication is the best option when security is of the highest significance.

Fingerprints are the marks impressions made on the surface by a person’s fingertip. Fingerprints reveal a lot about a person like their intelligence, personality, talents, etc. Every individual has a unique fingerprint structure and does not match with any other individual. It consists of ridges, bifurcation, ridge ending, crossover, island, core.

This project aims to deploy a biometric system that utilizes fingerprint recognition on the web to enhance accessibility and convenience. OpenCV will be used for fingerprint recognition, which will be integrated into a web application created using HTML and styled using CSS to create an intuitive and user-friendly interface.

Once the user has scanned their fingerprint on the scanner, the system will compare it with stored data to authenticate their identity. After successful authentication, the system will display the user's basic information, such as their name, photo, and other relevant details, using HTML and CSS to make it visually appealing and easy to read.

Overall, the web deployment of the biometric system using fingerprint recognition will provide a secure and convenient way for users to authenticate their identity and access relevant information. By using HTML and CSS, we can create an intuitive and user-friendly interface that makes it easy for users to interact with the system.

1. *Ridge:* Curved lines in a fingerprint.
2. *Ridge Ending:* Some ridges are not continuous curves. They terminate at some specific points. Those points are called ridge endings.
3. *Bifurcation:* Point where a ridge forks or diverges into branch ridges.
4. *Core:* This is the center part of the fingerprint.
5. *Crossover:* It is a connecting friction ridge made up of two bifurcations.

1.1.1. Fingerprint

A fingerprint pattern feature is shown in the below figure. It is an impression of the friction ridges and furrows on all the parts of a finger and these furrows and ridges present good matching in every small local window.

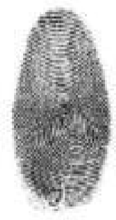


Fig 1.1 Fingerprint

Fingerprints are not distinguished by their ridges and furrows. Fingerprints are distinguished using a method called Minutia, which are some abnormal points on the ridges which are shown on Figure 2. Among the variety of minutia types reported in literatures, two are mostly significant and widely used:

1) Ridge ending - the abrupt end of a ridge

2) Ridge bifurcation - a single ridge that divides into two ridges.

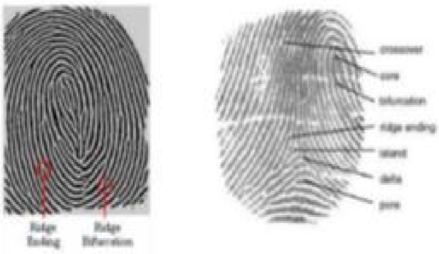


Fig 1.2 (a) Two important minutiae features (b) Other Minutiae feature

1.1.1. Fingerprint Recognition

Fingerprint recognition is the process of comparing a fingerprint against another fingerprint to determine if the impressions are from the same finger or palm. It includes two sub-domains: one is fingerprint verification and the other is fingerprint identification.

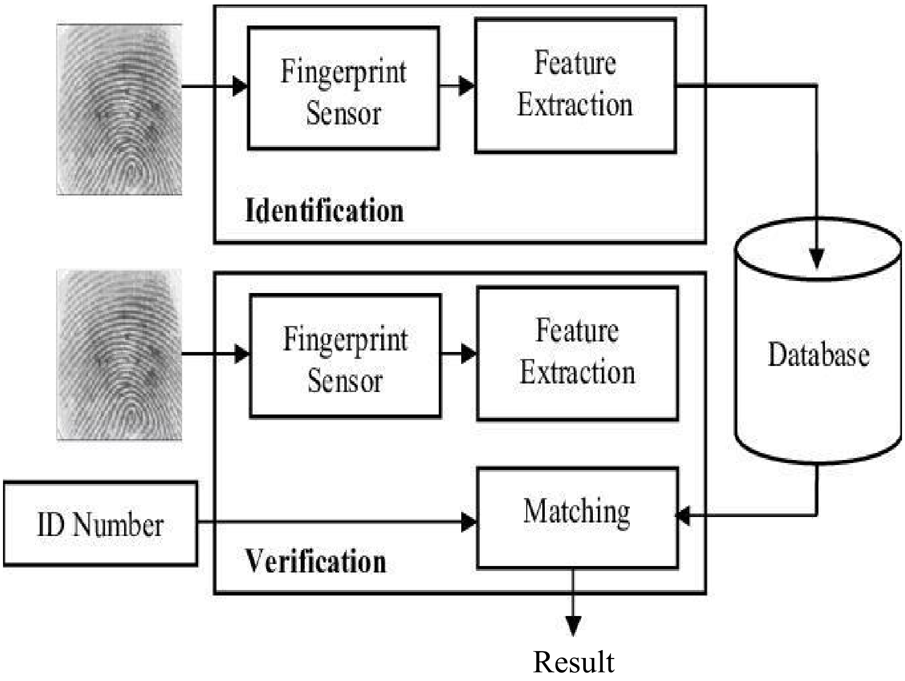


Fig 1.3 Fingerprint Verification v/s Identification

1.2. Problem Statement

Since biometrics are a key component of user identification in the healthcare industry, it follows that hospitals must have highly secure biometric authentication systems in place to verify patient identities and better manage the enormous amounts of patient data that are associated with them. Consequently, a much more secure method of fingerprint authentication is required.

Our project focuses on solving this issue using fingerprint recognition using OpenCV and proposes a suitable method for this issue. Fingerprint recognition is performed by identifying the key points and descriptors and matching those with the test data. The project aims at deploying on web using CSS and displaying the details of the patient by matching the fingerprint.

1.3. Objectives

Our project's main goal is to create a model for biometric authentication using ML Algorithms like SIFT and FLANN.

Other set of objectives that we aim to address are:

- To propose a secure architecture based on Machine Learning Algorithms for fingerprint recognition.
- To demonstrate how SIFT Algorithm and FLANN Matcher help to curate a secure model for fingerprint recognition.
- To demonstrate how the SIFT algorithm overcomes the drawbacks of the pre existing fingerprint recognition systems.

The primary goals of deploying a biometric system that utilizes fingerprint recognition on the web are:

- To establish a dependable and trustworthy method for users to verify their identity by using fingerprint recognition.
- To increase convenience and accessibility by deploying the system on the web, allowing users to access it from any location with internet access.
- To create a scalable, dependable system that can manage many users at once.

- To mitigate security concerns associated with biometric systems by building a robust and secure system that safeguards user data and credentials from potential threats, such as fraud and hacking.
- To take advantage of the latest advancements in technology, including OpenCV, machine learning, and portable devices, to design an efficient, precise, and cost-effective system.

1.4 Methodology

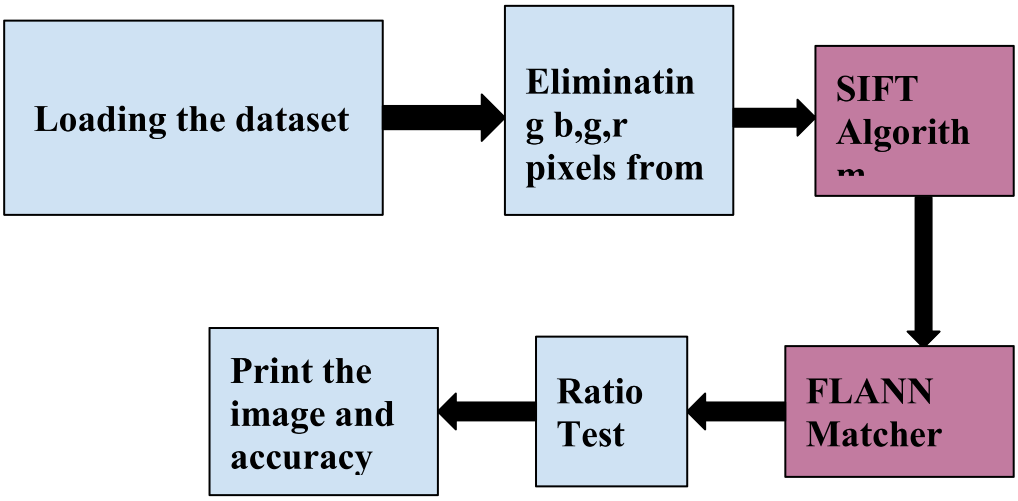


Fig 1.4 Fingerprint Authentication Model Framework

The Fast Library for Approximate Nearest Neighbours (FLANN) matcher was used to look for matches between the two photos for fingerprint recognition. This is typically applied to larger datasets.

Each of the images are initially evaluated to see if they share the same form and channels. We can divide them into three RGB channel images if they are the same size and number of channels. Now we tally the number of pixels that are not zero. We remove the image's b, g, and r pixels.

We employ the Scale Invariant Feature Transform (SIFT) technique in fingerprint identification to find keypoints and descriptors. The FLANN matcher was utilised in the following stage. It has a selection of algorithms that are well-suited for nearest neighbours searches in big datasets with many features. Two dictionaries, index and search parameters, are supplied inside the FLANN to specify the algorithm to be utilised. A ratio test is also used to develop effective keypoints. Based on the separation between the descriptions, effective keypoints are determined. The closer they are together, the better. Keypoints are generated using the distance restrictions. Greater than the two keypoints obtained from two images will be the total number of keypoints generated. Based on the number of keypoints obtained, keypoints are generated in both images and accuracy is obtained.

Then we sort the matches in ascending order of their distances so that the best matches with low distance come to the front. Images are stacked horizontally, and lines are drawn from the first image to the second image indicating the best matches. Initializing flags=2 means that draws two-match lines for each keypoint.

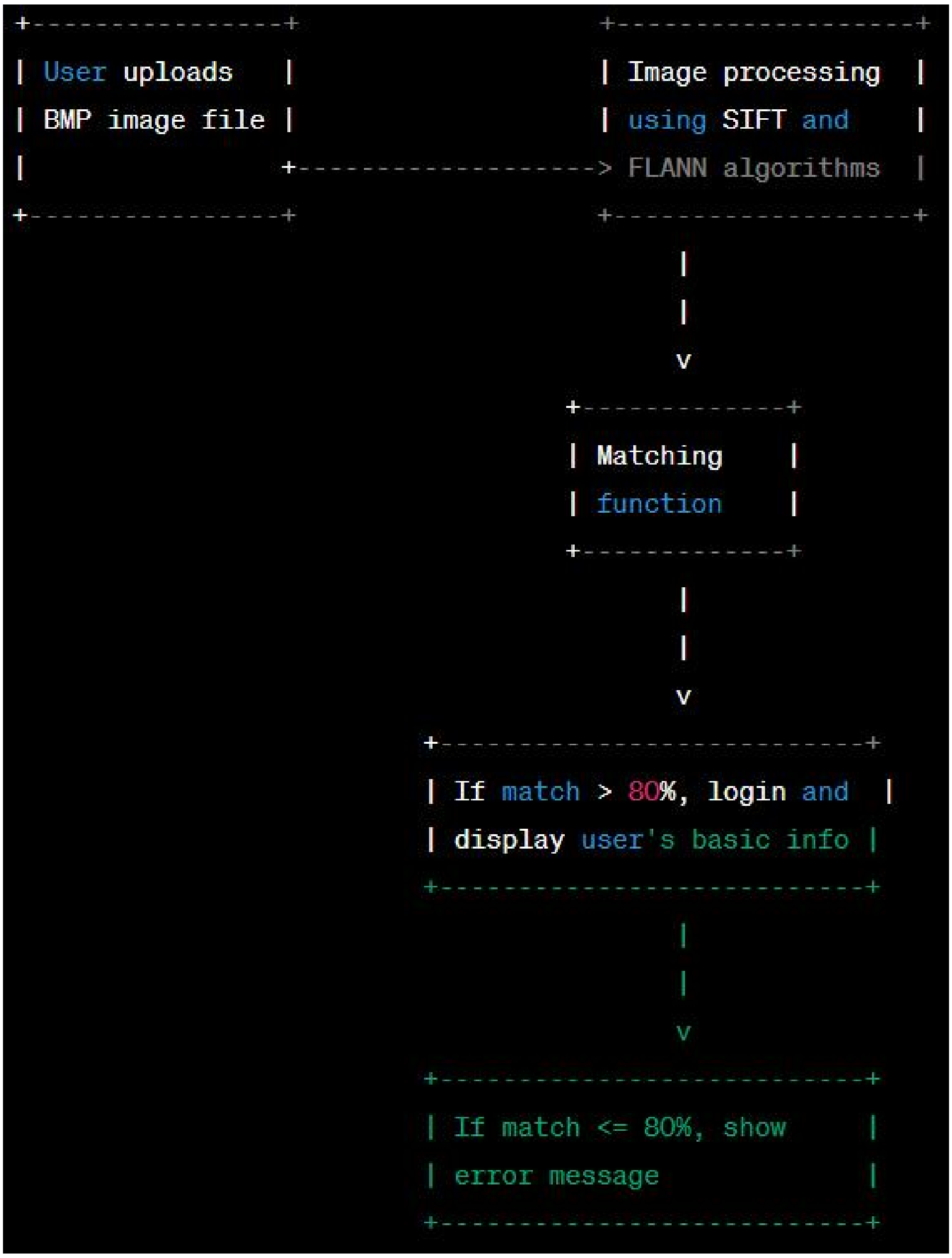
Design the user interface: This involves designing a user-friendly interface using HTML and CSS that guides users through the authentication process and displays their basic information.

Collect and preprocess the fingerprint data: This involves collecting high-quality fingerprint images from users and processing them to identify the key features or descriptors that can be used for matching.

Creating the web pages using HTML, including the pages where users can upload images of their fingerprints and where matching results are shown. The web pages can be styled with CSS to make them more aesthetically pleasing and user-friendly.

Monitor and optimize the system: This involves continuously monitoring the system to identify areas for improvement and optimize its functionality over time. It may involve implementing new features or updates based on user feedback and performance metrics.

An image is normally delivered to the server in binary file form when a user uploads it to a web application. You must change the binary image file into a buffer format that the SIFT and FLANN algorithms can read and process in order to process the image in your Django application.



```
+----------------+                    +-------------------+
| User uploads   |                    | Image processing  |
| BMP image file |                    | using SIFT and    |
|                +------------------> FLANN algorithms  |
+----------------+                    +-------------------+
                                           |
                                           |
                                           v
                                    +-------------+
                                    | Matching    |
                                    | function    |
                                    +-------------+
                                           |
                                           |
                                           v
                              +---------------------------+
                              | If match > 80%, login and  |
                              | display user's basic info |
                              +---------------------------+
                                           |
                                           |
                                           v
                              +---------------------------+
                              | If match <= 80%, show     |
                              | error message             |
                              +---------------------------+
```

Fig 1.5 Flowchart for web model

1.5. Organization

The report consists of various chapters as follows: -

Chapter 1:- The chapter gives a succinct overview of the project. Additionally, it provides an overview of the project and the fingerprint authentication system. This chapter also covers the project's overarching problem statement and objectives. The chapter includes a brief summary of the project's methodology as well as information on the steps needed in creating a fingerprint identification system utilizing machine learning and deep learning methods.

Chapter 2:- This chapter discusses the prior research on the fingerprint authentication technology. This also provides information about neural networks, machine learning, and deep learning. The journals and related papers that provide details on the earlier study are listed below. The chapter described the efforts made by several groups to create fingerprint identification systems using various models. We can choose the approach to use while creating our model or project by consulting the strategies and results included in this chapter. Readers can better grasp the advantages and disadvantages of various strategies for fingerprint authentication by looking at the methods and findings described in this chapter. The creation of new models or projects in this area can then be guided by this knowledge.

As it consolidates and summarises prior research in this field, the chapter appears to be a valuable resource for researchers or developers interested in working on fingerprint authentication systems.

Chapter 3:- This chapter provided details on the procedures we would use to construct the entire project. Both system development and model development are discussed specially for fingerprint matching using SIFT and FLANN. It certainly gives information about the system's general design, particularly how the fingerprint matching technique fits into the bigger picture. This could include details on the system's user interface, database management capabilities, and data flow between its various parts. It covers information on how to setup the data set, train and test the matching algorithm using SIFT and FLANN, and assess the results. The chapter also discusses the libraries that will be used in the project, including NumPy for numerical computation and OpenCV for image processing. The SIFT and FLANN algorithms themselves, including how they function, their benefits and drawbacks, and how they are typically used in fingerprint matching, may be covered in detail in this chapter. Understanding these concepts could help you better understand the project's guiding principles and further customise or improve the algorithm as needed. The chapter provides a thorough guide to building a complete fingerprint matching system, covering all aspects of the project from data preparation to model development to system implementation.

Chapter 4:- The entire project, including the method for developing the model and assessing its effectiveness, is thoroughly described in this chapter. It provides information on the various project levels, such as data collection and preprocessing, feature extraction, model training, and performance assessment. The specific actions conducted at each level, together with the resources and techniques employed to complete those actions, may be described. Details on the model that we created using a number of modules and packages are provided. It also contains the conclusions drawn from the many performance measures that we used during the project. It provides information on the model's accuracy as well as the predictions produced using the established model. The chapter can offer insightful information about the project's strengths and limitations and how it might be further enhanced in the future by outlining the results and performance indicators at each step.

Chapter 5:- This chapter serves as the project report's comprehensive conclusion to the effort. It offers details on how to improve the project and what may be done moving ahead in accordance with the goals of the research and its improvement.

A summary of the project's major findings and results, as well as any obstacles or limits that were faced along the way, are likely to be included in the conclusion chapter. It might also include an explanation of how these findings' implications connect to the study's goals. Ultimately, the conclusion chapter is likely to offer a plan for future work that expands on the current undertaking and fills in any gaps or restrictions that were discovered. The conclusion chapter can assist ensure that the project's findings have a long-lasting influence and contribute to continuous innovation in the field of fingerprint identification systems by clearly defining the direction for future research.

CHAPTER 2: LITERATURE SURVEY

2.1. Biometric Authentication

Numerous studies have been done on the application of neural networks for biometric authorization and verification over the years. A potent machine learning method known as neural networks may be used to identify and categorise patterns in huge datasets, including biometric information like fingerprints.

For biometric authentication, researchers have investigated a wide range of neural network topologies and methods, such as feedforward neural networks and convolutional neural networks (CNNs). The application of deep learning and neural networks for biometric authentication has promise, and more study in this field may result in the creation of systems that are more precise and dependable for user identification and verification.

A classification neural network-based biometric authentication system was the subject of a copyright submission by [4]. Building a reference order of permitted clients' fingerprint identities then verifying that used a classifying neural net are the two steps of the copyrighted biometric system.

[5] In order to authenticate biometric data, a novel directed recursive neural system has been developed. Their methods used similarity measurements of features for the purposes of ranking and clustering the fingerprint representations that were stored in their database.

Recursive neural networks (RNNs), a kind of neural network architecture that can process sequential data, are being used in this method to show the potential for biometric authentication. The system can find patterns and similarities among various fingerprints by comparing fingerprint characteristics using similarity measurements, and it can use this knowledge to authenticate individuals.

[6] analyses the use of machine learning methods to find typing patterns as a viable biometric identifier for user identification and verification.
The k-NN algorithm is a kind of non-parametric classifier that categorizes new data points based on the most prevalent class label among those k nearest neighbors. It operates by locating the k-nearest data points in a training set to a given data point. When there are several features or dimensions in the data for a classification problem, this approach is frequently utilised.

The use of Weightless Neural Networks (WNNs) as a sequential tool to categorise people's typing preferences is examined in [7] in an effort to distinguish genuine users from imposters.

In order to train a WNN-based classification model, the system required users to provide timing data for the keyboard, including key press and release times. The performance of the authors' suggested system was compared to a number of other cutting-edge authentication systems, such as conventional password-based systems and biometric systems based on keystroke dynamics and fingerprint identification. They discovered that their suggested WNN-based approach produced greater accuracy rates and was less vulnerable to assaults like replay attacks and shoulder surfing.

Emphasised the benefits of utilising WNNs for this kind of classification work, such as its capacity to handle noisy and incomplete data, resistance to adversarial assaults, and minimal processing requirements. Their work overall showed that WNNs have the potential to be a useful tool for biometric authentication across a range of application domains.

[8] uses artificial neural networks to learn and recognize facial representations. Recent years have seen a major advancement in

representation training for biometric IDs using neural networks due of the current surge in interest in deep neural networks.

Discussed the benefits of CNNs for recognizing faces, including as its capacity to build hierarchical representations that capture both low-level and high-level information, their resistance to changes in position, illumination, and expression, and their capacity to handle enormous datasets.

A few recent developments in representation learning using neural networks for biometric identification problems were also covered in the paper. Among other biometric modalities, these developments use deep neural networks to recognize faces, fingerprints, and irises. The scientists pointed out that these techniques have substantially increased the performance of biometric identification systems in recent years and they proposed that additional study in this field might result in even more precise and dependable systems.

FingerNet has been given by [9]. They outline a brand-new approach to creating deep convolutional networks that blends domain expertise with the representational strength of deep learning. Many widely used conventional methods that were successful on wrapped biometrics are transformed into a functional approach for orientation estimation, classification, refinement, and minutiae extraction, and emerge as a single straightforward model.

The global representation component uses a deep convolutional neural network (CNN) to extract a global feature representation of the fingerprint image. Using a conventional technique known as crossing numbers, the minutiae extraction component harvests minutiae, the distinctive characteristics of a fingerprint. The refining component uses a deep CNN to improve the predicted minute locations. Another deep CNN is used by the classification component to classify the fingerprint image.

[10] proposed MENet, to provide an information driven illustration of minor details and put the problem of minutiae retrieval into the context of machine learning.

Assessed the strategy using a number of open fingerprint datasets and contrasted the outcomes with cutting-edge techniques. demonstrated that the method outperformed existing techniques in terms of minutiae extraction and detection. The outcomes proved the value of the MENet strategy and its applicability to real-world biometric systems.

[11] emphasized the issue of biometric spoofing, specifically for iris, face, and fingerprint recognition. To eliminate spoofing attempts, they suggested using a deep representation-based technique. Real and fictitious biometric data were used to train the authors' deep network, and publically available datasets were used to assess the network's performance. According to their findings, the suggested method successfully detected spoofing attempts with a high degree of accuracy. Deep neural networks were used in [12] to learn representations for fingerprint pictures. a technique known as DeepFinger was put out, which trains a deep convolutional neural network on a sizable dataset of fingerprint photos to develop a hierarchical representation of the images. Their method outperformed a number of cutting-edge techniques when the learned representations were used for fingerprint recognition.

The goal of [13] was to create a fingerprint categorization system based on the VGGNet architecture. The three possible iterations of the VGGNet structure and assessed each one's performance using a fingerprint dataset that was available to the public. Their findings demonstrated the high classification accuracy of the suggested models for fingerprints, demonstrating the promise of deep learning approaches for biometric authentication.

Convolutional neural networks (CNN) and Q-Gaussian multi support vector machines (QG-MSVM) based on different level fusion were also suggested as components of a protected multisensory authentication system [14]. They developed two authentication systems using two distinct layer fusion techniques feature - based fusion and a decision level fusion. The pattern retrieval for various modes is done using a CNN. The two CNN layers with the maximum precision were selected for this stage, and evey level acts as a unique feature descriptor. Then, in order to create the biometric templates, they fused them using the suggested internal fusion. Then, to safeguard these templates and boost the privacy of the suggested model, they used one of the cancelable biometric procedures.

Additionally, [15] performed multimodal biometric face and fingerprint identification using multilayer perceptrons and neural networks based on adaptive principal component analysis. The study emphasised the potential advantages of merging various biometric modalities for improved performance and showed how well neural networks work for multimodal biometric identification.

[16] put forth a creative solution based on neural networks for distinguishing overlapping fingerprint impressions. Overlapping fingerprint impressions can make it harder to accurately recognise a fingerprint, which lowers the efficiency of biometric devices. The suggested technique uses a convolutional neural network (CNN) to categorise every pixel of an image as either being in the foreground or background. Using a collection of fingerprint photos with varied levels of overlap, the network is trained. After being trained, the network is used to distinguish between overlapping regions in fresh fingerprint photos, enhancing the precision of fingerprint recognition. The efficiency of the neural network-based strategy was demonstrated by the experimental findings, which indicated that the

suggested method outperforms current state-of-the-art methods for overlapping fingerprint separation.

[17] used convolutional neural networks (CNNs) to assess a fingerprint's liveness.In order to identify the intricate QRS segments in ECG signals using neural networks, [18] then performed user authentication on these segments.

Neural networks are used in [19] to verify the identification of electrocardiograms (ECGs). They employed multilayer perceptrons (MLPs) and radial basis function neural networks (RBFNNs) to recognise people based on their ECG readings. For ECG-based biometric identification, classification, and verification utilising generalised recurrent unit (GRU), long short-term memory (LSTM), and vanilla RNNs, they also studied a number of recurrent neural network (RNN) topologies, such as unidirectional, bidirectional, and gated RNNs. The outcomes demonstrated that RNN-based models performed better than conventional MLP and RBFNN models in terms of accuracy and robustness, highlighting the potential of ECG biometrics and the efficacy of RNN-based techniques for this application.

An innovative technique for biometric identification and verification using electrocardiogram (ECG) data was introduced by the authors in [21]. It is called Deep-ECG. To extract important properties from one or more ECG signal leads, Deep-ECG uses a deep convolutional neural network (CNN). Then, these features are applied to verify or identify the leads as well as compare biometric templates.

Potential uses for the Deep-ECG technique exist in a number of industries, including security, finance, and the medical field. It might be applied to financial transactions or to the identification of people for the preservation of medical records.

2.2. Techniques for Fingerprint Authentication

According to a research on "Fingerprint Recognition using Robust Local Features" proposed by Madhuri and Richa Mishr (2019) [22], there are numerous human recognition approaches that are based on fingerprints currently in use. For fingerprint matching and representation, the majority of these techniques use minute points. When a person's enrolled image is matched with a rotated test image, these approaches, which are not rotation invariant, fail. They also fail when only a portion of a fingerprint image is matched. The fingerprint representation and matching method that is suggested in this paper employs local robust features.

In their suggested article on "Fingerprint Recognition Using Minutiae Extractor," Manisha Redhu and Dr. Balkishan (2019) [23] claim that the widely used biometrics are used to authenticate a person's fingerprint, which is distinct and persistent throughout the person's life.The term "fingerprint recognition" refers to automated techniques for determining if two human fingerprints match. Since more than a century ago, fingerprints have been employed often in daily life due to their practicality, distinctiveness, permanence, accuracy, reliability, and acceptability. They used the Minutiae Score matching method to project fingerprint recognition in their work.

The publication "Fingerprint Recognition using Image Segmentation" by Sangram Bana and Dr. Davinder Kaur [24] proposes a study and implementation of a fingerprint identification system based on minutiae-based matching approaches. This method primarily entails extracting minutiae points from sample fingerprint pictures and then doing fingerprint matching based on the quantity of minutiae pairings between the two fingerprints in question.

The originality, distinctiveness, and dependability of biometric fingerprints make them the finest personal identity approach, according to Ritu and Matish Garg's study "A Review on Fingerprint-Based Identification System" (2020)[25]. Fingerprints are made up of valleys or ridges on human fingertips. The most sophisticated biometric technology used for authentication may be based on fingerprints. The reliability of fingerprint authentication has been rigorously tested in numerous applications. The three pillars on which all human recognition algorithms based on fingerprints are constructed are the minutiae-based, correlation-based, and hybrid techniques. A study of several fingerprint recognition techniques is followed by a discussion of a general minutiae-based fingerprint identification system.

Priyanka Rani and Pinki Sharma (2020) [26] have suggested a paper on "Fingerprint Identification System". The most complicated biometric authentication method, according to them, is fingerprint authentication, which has been carefully verified across a range of applications. Even physical traits, such as a person's face or signature, can change with time and even be imitated or falsified. But a fingerprint is particular to one person and doesn't change over time. The different parts and procedures for the fingerprint-based identification system are described in this paper.

Gurpreet Singh and Vinod Kumar (2021) [27] write in a paper titled "Fingerprint Recognition: Minutiae Extraction and Matching Technique" that recent advancements in fingerprint identification and authentication have inspired many people to do research in this field. Fingerprint identification is developing as a new area for user authentication. Large organisations give fingerprint categorization a lot of weight when utilising fingerprint identification systems. When two fingerprints do not match, fingerprint identification, which also expedites the identifying process, can substantially improve the authenticating process. This article provides a comprehensive assessment of the classification techniques.

2.3. Privacy-Preserving Biometrics

Our mission is related to the issue of secrecy sustaining biometric authentication in that our SIFT and FLANN technique aids with some of the standard confidentiality biometrics goals, such as protecting biometric templates [3]. However, the issue of privacy-preserving biometric authentication is far more serious and concerns both the security of the authentication dataset and the security of the entire biometric authentication mechanism, from anonymous biometric verification [18] to biometric cancellability. The use of fuzzy extractors [12], safe-guard multi-party computations [8], fully homomorphic encryption [18, 30], zero-knowledge claims of knowledge [4], and fully homomorphic encryption [4] are a few of the methods employed in the literature to accomplish these various objectives.

CHAPTER 3: SYSTEM DEVELOPMENT

3.1. Resources

3.1.1. Hardware Requirements

- A physical memory (RAM) of 4 GB and above are required
- Hard disk capacity: Maximum 5 GB
- Laptop

3.1.2. Operating System

- Windows7/8/8.1/10

3.1.2. Language Requirement

- Python

3.1.2. Tools and Framework

- OpenCv

3.2. Research Survey

In this project, we have tried to come up with a more secure approach for fingerprint authentication.

But, before proceeding with this approach we conducted a survey so as to understand the adaptability of fingerprint authentication as a substitute of the conventional two factor authentication common these days.

Below are a few sets of figures depicting the majority of the answers to the proposed set of questions, followed by a brief analysis of this survey.

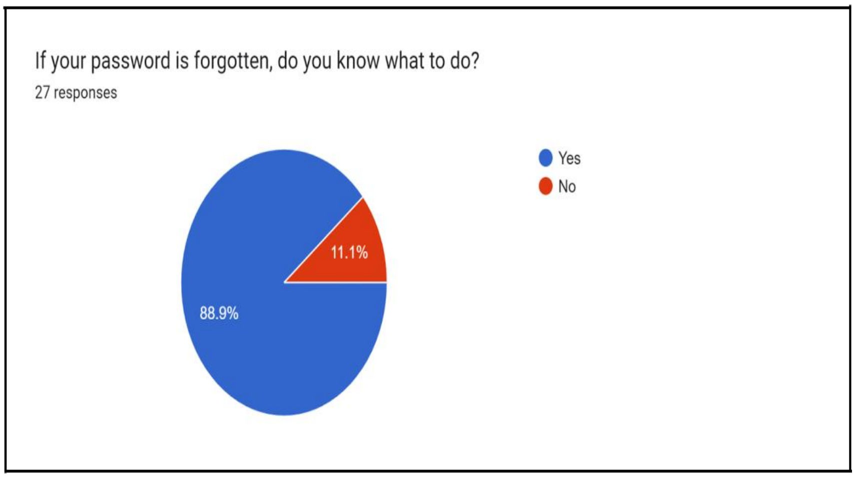
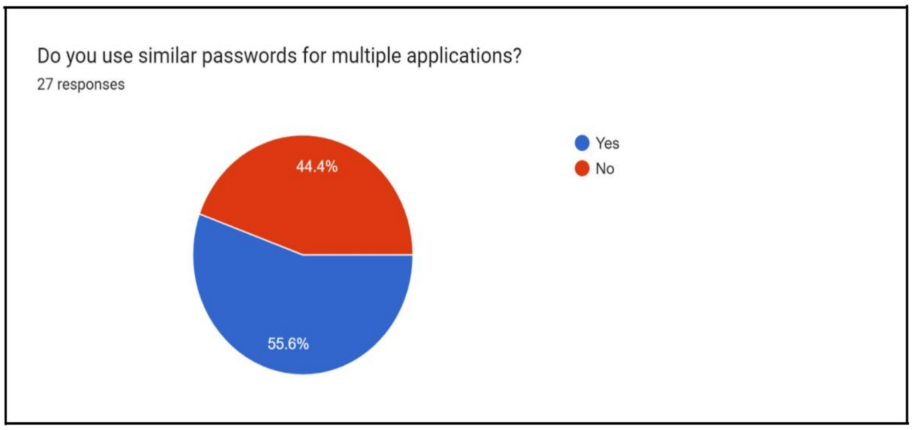


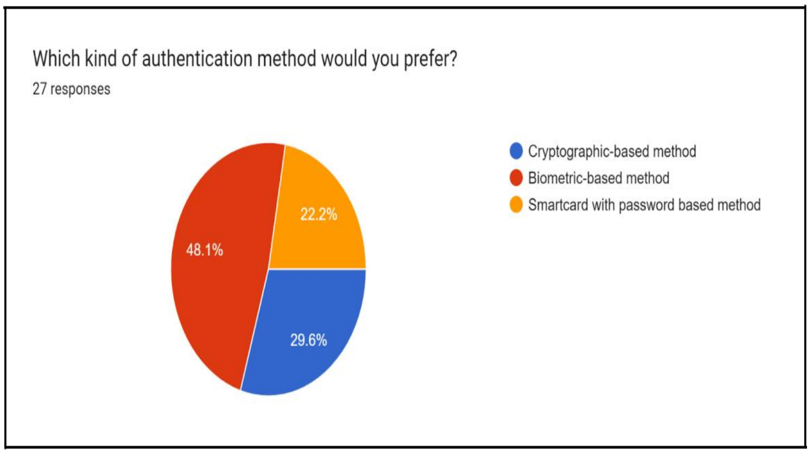
Fig 3.1 Survey Question 1



Do you use similar passwords for multiple applications?

27 responses

Fig 3.2 Survey Question 2



Which kind of authentication method would you prefer?

27 responses

Fig 3.3 Survey Question 3

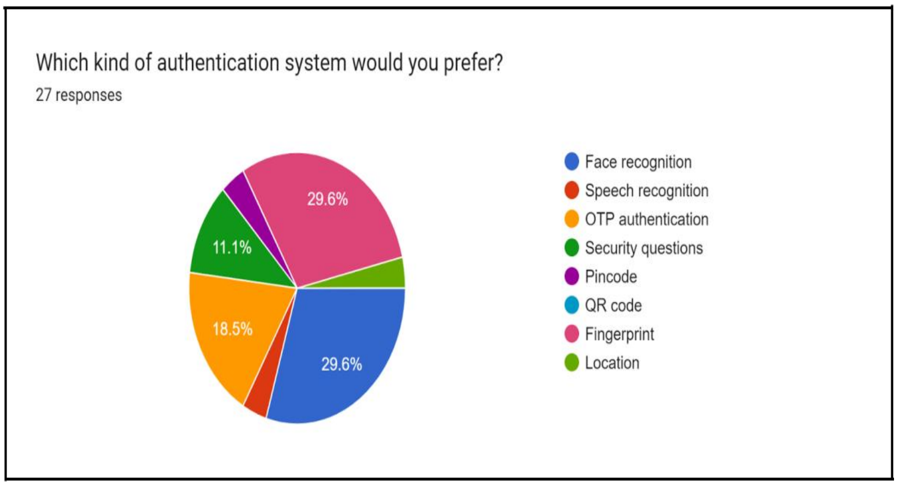


Fig 3.4 Survey Question 4

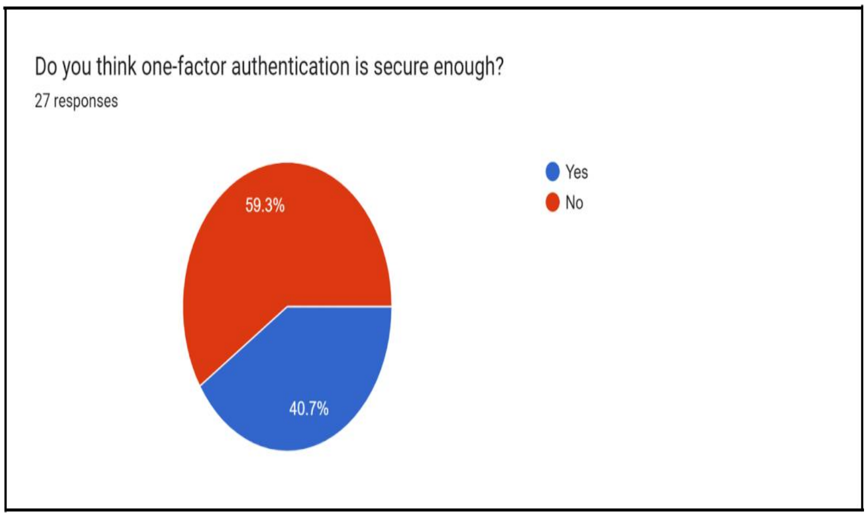


Fig 3.5 Survey Question 5

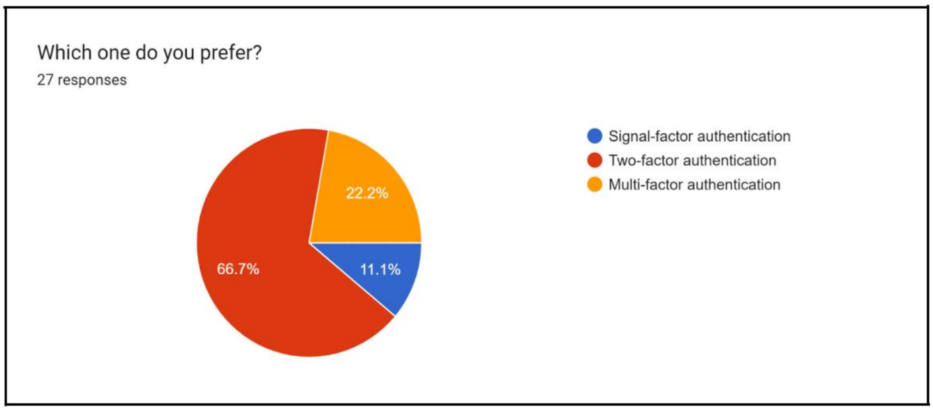


Fig 3.6 Survey Question 6

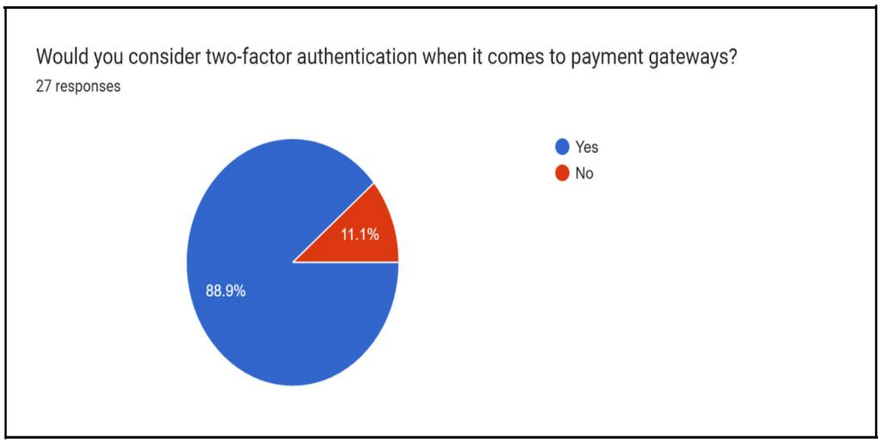


Fig 3.7 Survey Question 7

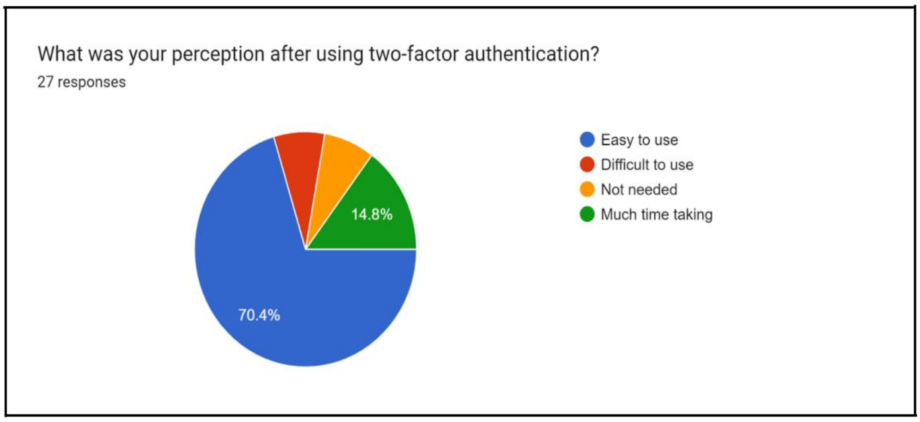


Fig 3.8 Survey Question 8

Furthermore, the following analysis could be inferred from this survey:

- A small proportion of people become uneasy in the event that their password is compromised. They are unable to modify their credentials in order to restore control of their account and stop the hacker from using it for a prolonged period of time to carry out other fraudulent operations.

- The majority of people use the same passwords on all platforms. As a result, if one of their accounts is compromised, it is quite likely that other accounts will also be attacked, and they run the danger of losing a lot of sensitive information. When biometric components are included, password theft becomes much more difficult.

- Since biometric authentication is more dependable and difficult to forge or steal, the majority of people preferred it. The most popular and often used method is the one-time password sent over SMS. Additionally, it was discovered that people preferred biometric technology because it is secure and more difficult to copy or steal.

- Most people think that biometric security, which is unique to each and every person, is the best kind of protection that is currently available and is adequate to protect their information.

- Only a small percentage of people believe that multi-factor authentication is time consuming and negatively affects user experience. One factor authentication, on the other hand, is not frequently used because of the high dangers involved; this shows that better solutions must be incorporated.

- Many people want better security measures to be added to online financial consultancies, digital transaction applications, and other internet businesses where money is traded.

- Few people actually think that utilising 2FA is simple. A small proportion of participants stated they found it difficult, and a nearly equal number said they found it extremely tedious and time-consuming. These findings suggest that, despite the fact that the vast majority of people are familiar with the technology and how it works, a sizable portion of the population is still utterly ignorant of its application.

3.3. SIFT

Scale-Invariant Feature Transform, or SIFT, was introduced for the first time by D. Lowe of the University of British Columbia in 2004. Image scale and rotation invariance is known as SIFT. This algorithm is included in the Non-free module of OpenCV since it is patented.

General-purpose object recognition was the initial reason for which Scale Invariant Feature Transformation (SIFT) [4] was created. Using the descriptors for each feature point, SIFT finds stable feature points in a picture and matches them.

Major advantages of SIFT are:

- **Locality:** features are local, so robust to occlusion and clutter (no prior segmentation).
- **Distinctiveness:** individual features can be matched to a large database of objects.
- **Quantity:** many features can be generated for even small objects.
- **Efficiency:** close to real-time performance.
- **Extensibility:** can easily be extended to a wide range of different feature types, with each adding robustness.

SIFT is quite an involved algorithm. There are mainly four steps involved in the SIFT algorithm. We will see them one-by-one.

- **Scale-space peak selection:** Potential location for finding features.
- **Keypoint Localization:** Accurately locating the feature keypoints.
- **Orientation Assignment:** Assigning orientation to keypoints.
- **Keypoint descriptor:** Describing the keypoints as a high dimensional vector.
- **Keypoint Matching :** Finding corresponding keypoints between two images is the process of keypoint matching in SIFT, which is useful for image alignment, object detection, and tracking.

Scale-space peak Selection

Scale-space

Only at a certain scale do real-world items have any significance. A sugar cube may appear precisely placed on a surface. But if you take a broad view of the Milky Way, it is plainly impossible.

It's extremely typical for items in nature to have multiple scales. And a scale space makes an attempt to represent this idea on digital photographs.

The scale space of an image is a function L(x,y,) that is created via the convolution of a Gaussian kernel (Blurring) at various scales with the input picture.

The number of octaves and scale in scale-space depend on the size of the source image. As a result, we produce the original image in several octaves. The size of the image decreases by half for each octave.

Applying a variable scale Gaussian operator to an input image creates a scale space. By deducting the following scales in each octave, Difference of Gaussian (DOG) images are produced. An octave is the collection of DOG and Gaussian-smoothed pictures. By repeatedly down sampling the original image, a set of these octaves is created.

For SIFT operation, 5 and 6 octaves, respectively, are normal numbers of scales. The graphic below displays the relevant difference images for each of the four subsequent octaves with five scales.

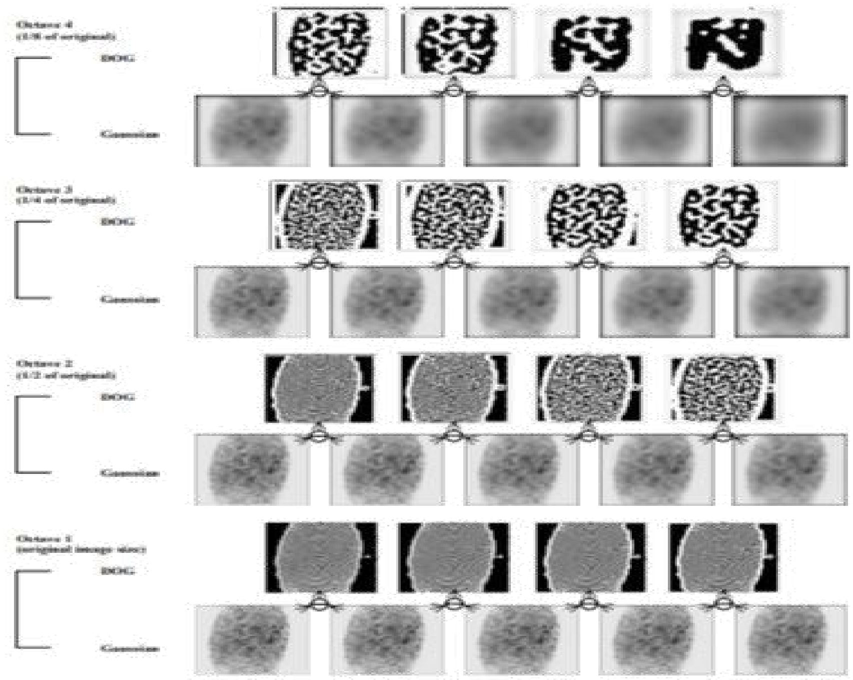


Fig 3.9 Scale Space Recognition for SIFT Operations

Blurring

Within an octave, images are progressively blurred using the Gaussian Blur operator. Mathematically, “blurring” is referred to as the convolution of the Gaussian operator and the image. Gaussian blur has a particular expression or “operator” that is applied to each pixel.

What results is the blurred image.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$$

Fig 3.10 Blurred image

G is the Gaussian Blur operator and I is an image. While x,y are the location coordinates and σ is the “scale” parameter. Think of it as the amount of blur. Greater the value, greater the blur.

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}$$

Fig 3.11 Gaussian Blur operator

DOG (Difference of Gaussian kernel)

The Difference of Gaussians (DoG) images are now created using the blurred images as a starting point. Finding intriguing focal areas in these DoG photos is quite helpful. When a picture is blurred using two different, let's say k and, the difference of Gaussian is obtained. The image in the Gaussian Pyramid is processed in this way for various octaves. The following picture illustrates it:

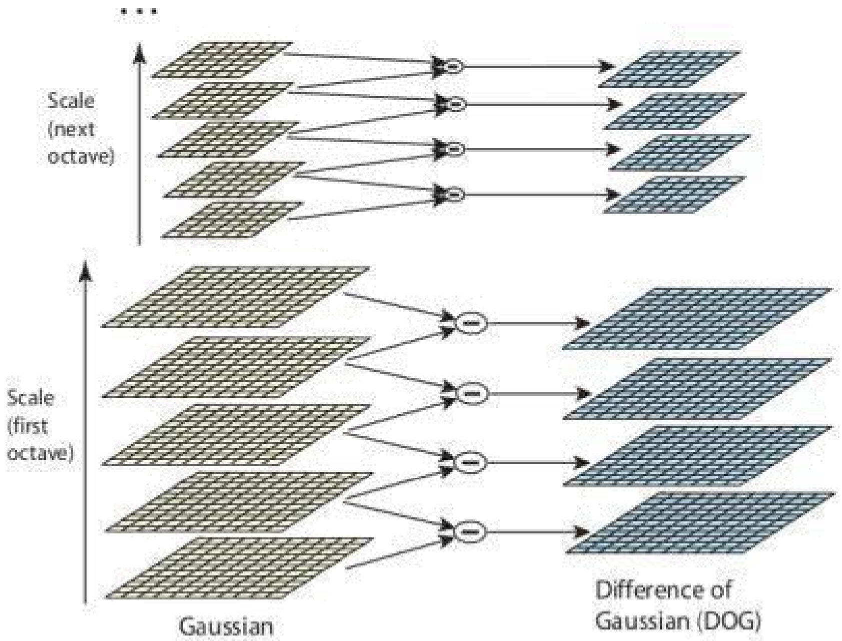


Fig 3.12 Difference of Gaussian set of images

Finding keypoints

Up till now, we have generated a scale space and used the scale space to calculate the Difference of Gaussians. Those are then used to calculate Laplacian of Gaussian approximations that are scale invariant.

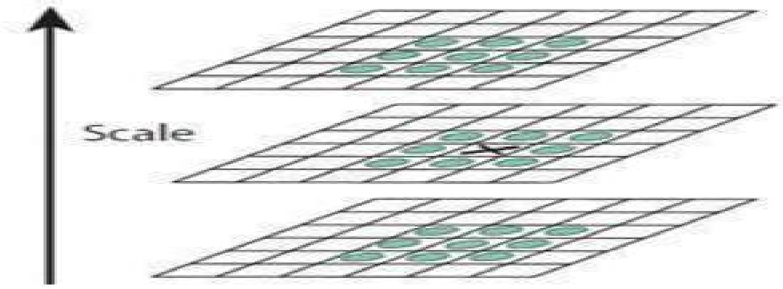


Fig 3.13 Scale Invariant Laplacian of Gaussian Approximations

A pixel in a picture is compared to 8 of its neighbours, 9 of the scale below it, and 9 of the scales above it. 26 inspections are performed in total this way. A possible keypoint exists if it is a regional extremum. It basically says that scale best captures that keypoint.

Keypoint Localization

The keypoints produced in the preceding phase result in a large number of keypoints. Some of them have insufficient contrast or are located on edges. They are less beneficial than features in both situations. Therefore, we eliminate them. The method is comparable to that employed in the Harris Corner Detector to get rid of edge features. In the case of low contrast features, we merely examine their intensities.

Orientation Assignment

Now that we have solid key points. Their stability has been tested. As the scale of the blurred image is the same as the scale at which the keypoint was detected, we already know this scale. Thus, scale invariance exists. To make each keypoint rotation invariant, the next step is to assign an orientation to it.

Keypoint descriptor

Each keypoint at this point has a location, a scale, and an orientation. Next, compute a descriptor for the local image region surrounding each keypoint that is highly distinctive and as resistant to variations as possible, such as shifts in viewpoint and lighting.

To do this, a 16x16 window is taken around the keypoint. It is separated into 16 separate 4x4 blocks.

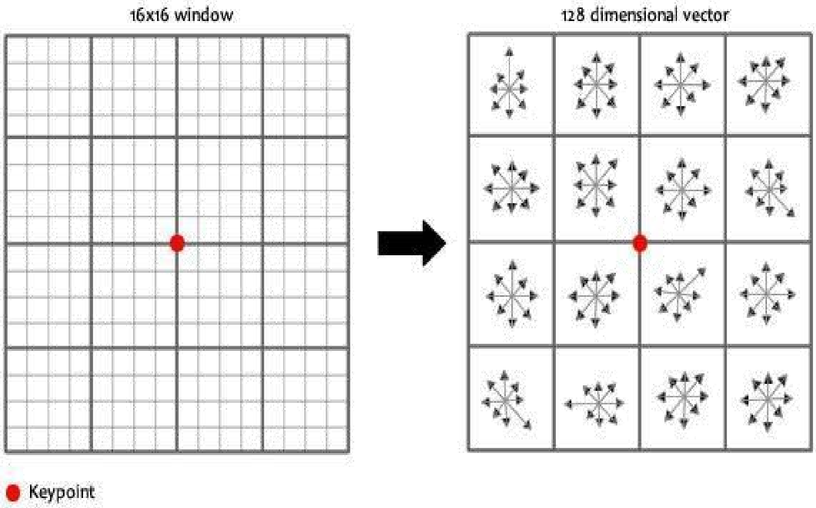


Fig 3.14 A 16x16 window around the keypoint

For each sub-block, 8 bin orientation histogram is created.

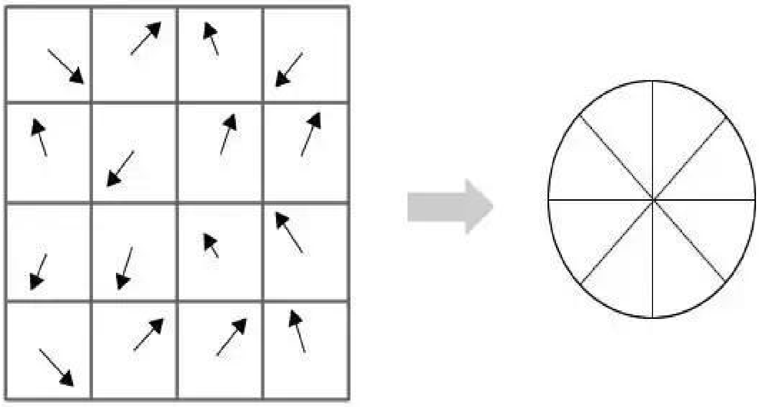


Fig 3.15 8 Bin Orientation Histogram

So 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 X 8 directions give 128 bin values. It is represented as a feature vector to form keypoint descriptor. This feature vector introduces a few complications. We need to get rid of them before finalizing the fingerprint.

1. **Rotation dependence** : In the feature vector, gradient orientations are used. It is obvious that everything changes if you rotate the image. The directions of every gradient also shift. The keypoint's rotation is subtracted from each orientation in order to achieve rotation independence. As a result, the orientation of the keypoint is relative to each gradient orientation.

2. **Illumination dependence** : We can attain illumination independence if we threshold large numbers. This means that any number (of the 128) that is bigger than 0.2 is converted to 0.2. The resulting feature vector is once more normalised. You now have a feature vector that is independent of illumination.

Keypoint Matching

By locating their closest neighbours, keypoints between two photos are matched. The second-closest match, though, may occasionally be quite close to the first. Noise or other factors can be at blame. The closest-distance to second-closest distance ratio is used in that situation. They are rejected if it is higher than 0.8. Only 5% of the true matches are discarded, while about 90% of the false matches are removed.

3.4. SIFT on Fingerprint Images

3.4.1. Characteristic feature points in fingerprints

The ridge ending and bifurcation points are the only locations that firmly define minutiae points. This means that a fingerprint image can only contain a modest (100) number of minute points. SIFT points, however, are only constrained by the equirement of local minima or maxima in a certain scale space, leading to a vast number of feature points. The number of octaves and scales, among other factors, have an impact on the amount of SIFT feature points. The number of SIFT feature points in a typical fingerprint might reach a few thousand. Figure following provides an illustration of SIFT feature points and minute points on the same fingerprint image. The number of SIFT feature points is seen to be 2,020 despite their being only 36 minutiae points.

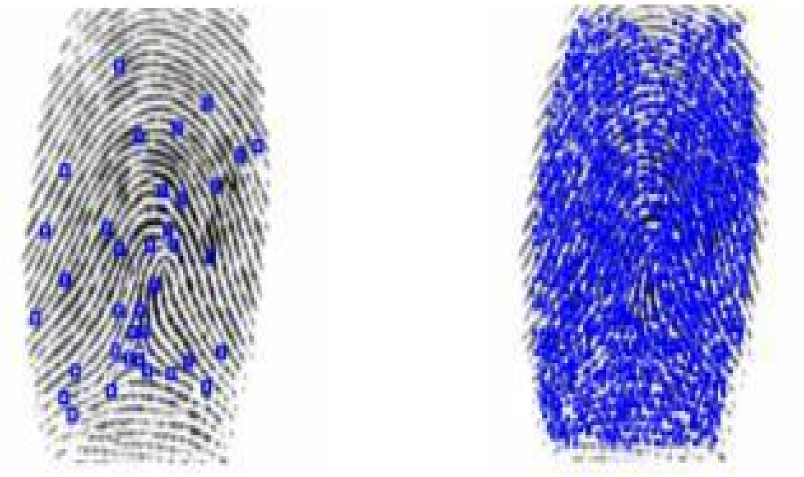


Fig 3.16 Minutiae and SIFT feature points extracted from the same image

(a) minutiae points (b) SIFT feature points

3.4.2. Fingerprint verification using SIFT

3.4.2.1. Preprocessing

Despite the fact that SIFT was initially designed for general-purpose object recognition and does not need picture pretreatment, we have added a few preprocessing procedures to fingerprint images in order to improve matching performance. Graylevel distribution adjustment and noisy SIFT feature point removal are the two steps in the preprocessing procedure. Because SIFT uses texture information for both extracting feature points and matching, it is anticipated that performance will be improved when the fingerprint images have similar textures. The same logic also applies to the removal of noisy SIFT feature points to improve matching performance. First, we adjust the histogram by measuring the image intensity in the fingerprint's central region in order to correct some apparent differences in grey level distributions. Second, since local extrema, a fingerprint's boundary region always results in the detection of some feature points. The boundary region varies, even for the same finger, with each fingerprint impression. This means that feature points on the fingerprint boundary typically provide erroneous matches.

.

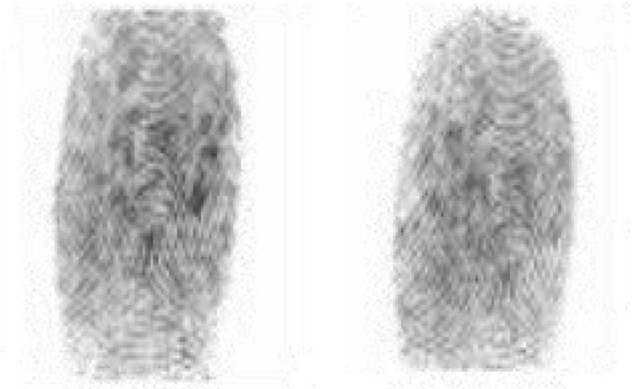


Fig 3.17 Input pair of fingerprints

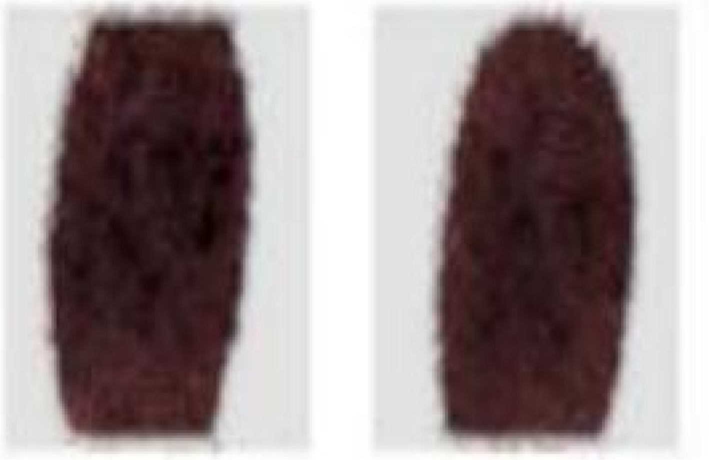


Fig 3.18 Preprocessing generate masks to remove SIFT points on the boundary of fingerprint

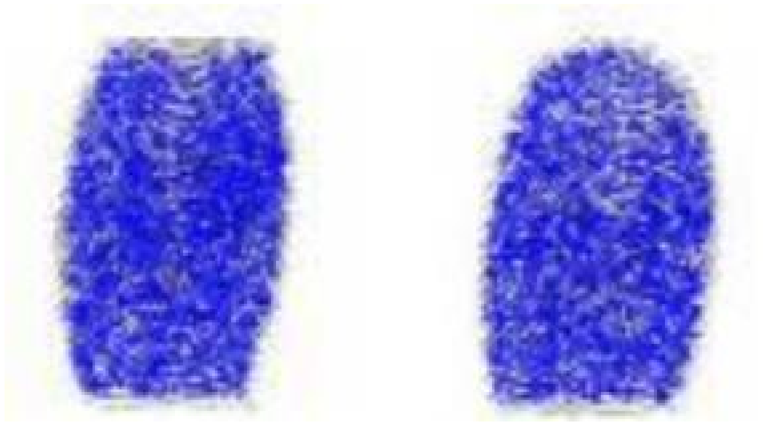


Fig 3.19 Local Extrema & Descriptor Extraction

3.4.2.2. Point Wise Matching

Each feature point is directly compared using the Euclidean distance metric based on the descriptor as the first step in matching.

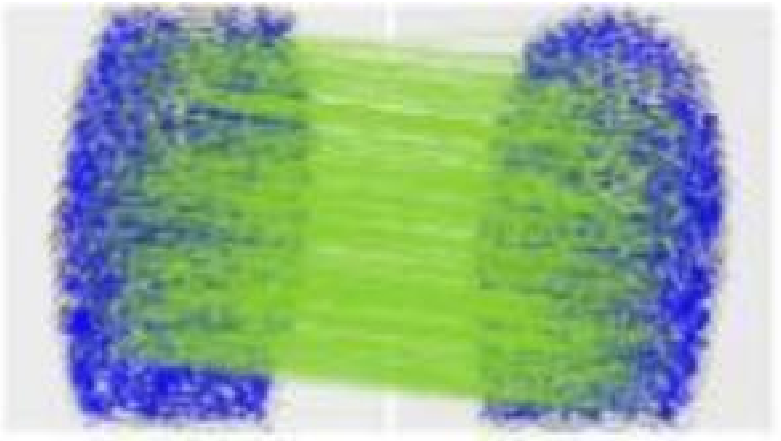


Fig 3.20 Point wise matching (580): different shades in the background is the effect of brightness adjustments

3.5. FLANN

The closest neighbour search (NSS) problem can be described as follows: It is necessary to preprocess a group of points in a metric space X so that, given a new query point q X, finding the point in P that is closest to q can be done efficiently. A number of applications, including image recognition, data compression, pattern recognition and classification, machine learning, document retrieval systems, statistics, and data analysis, all heavily rely on the problem of closest neighbour search. There is no technique that outperforms the traditional brute-force search, making it an extremely challenging undertaking to solve this problem in high dimensional environments. A class of algorithms that do approximate closest neighbour searches, which have shown to be a good enough approximation in most practical situations and in most cases, orders of magnitude faster than the algorithms performing the precise searches, have seen an increase in interest as a result.

In high dimensional spaces, FLANN is a library for quick approximative closest neighbour searches. It includes a selection of the nearest neighbour search algorithms we have found to be most effective as well as a system for automatically selecting the most effective algorithm and optimal parameters for each dataset. In addition to having bindings for C, MATLAB, Python, and Ruby, FLANN is developed in the C++ programming language.

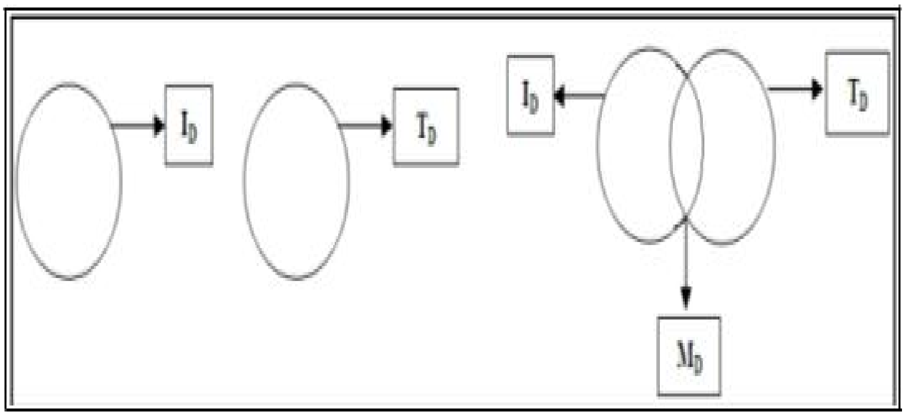


Fig 3.21 Descriptor Matching using FLANN

Descriptor matching is carried out using the FLANN algorithm (shown in Fig. 25) by initially Considering that I and T, respectively, stand in for the Query and Gallery image. Additionally, = 1, 2, 3, etc., and = 1, 2, 3, etc.

Indicates a collection of Test Image (I) and Template Image (T) critical points. Consider the following equations: = 1, 2, 3, etc., where ID is a set of descriptors (feature vectors) for image I and = 1, 2, 3, etc., where TD is a set of descriptors for image T.

Then, "11, 12, 13,..." stands for the group of matched descriptors.

3.6 Web Deployment

3.6.1 Django

Django is a high-level Python web framework that is used for the speedy development of web applications. It is built on the Model-View-Controller (MVC) architectural pattern and offers a robust toolset for creating web applications rapidly and effectively.

Django is used in the fingerprint matching project to deploy the project on a web server and offer an intuitive user interface for the user authentication system. The Django framework has built-in security features and a sizable support community.

Features of Django:

Among the main aspects of Django that make it a well-liked option for web development are the following:

- Object-relational mapping (ORM): Django offers a high-level ORM that enables programmers to communicate with databases using Python classes and objects as opposed to writing direct SQL queries.

- Web-based data management is made possible through Django's integrated admin interface, which is available to developers.

- URL mapping: Django offers a robust framework for URL mapping that enables developers to map URLs to views and manage the flow of data within the application.

3.6.2 Django in fingerprint matching

Django is utilised in the fingerprint matching project to provide the web interface for the programme and to manage user authentication. The main actions needed to use Django for this project are listed below:

- Model definition : The Django model is intended to hold the user's fundamental information as well as fingerprint information.

- Watch definition : The user login and authentication process is managed by the Django view definition. The user's fingerprint image is input into the view in buffer format, and it then uses the SIFT and FLANN algorithms to compare it to the fingerprint database that already exists.

- Template explanation : The Django template is set up to show the user's fundamental information if the fingerprint match is successful or an error message if it is not.

3.6.3 Advantages of Django:

For web development, Django offers a number of benefits, including:

- Rapid development: Django offers a strong toolbox for quickly and effectively creating web apps.

- Scalability: Django is very scalable and can manage websites with significant traffic.

- Security: Built-in security measures in Django include defenses against cross-site scripting (XSS) and SQL injection, two typical web threats.

- Support from the community: Django has a sizable and vibrant developer community that contributes to its growth and offers assistance through forums and documentation.

3.6.4 Django implementation

The images are submitted in BMP format for the project that compares fingerprints, and they are afterwards transformed to binary bytes buffer format. Following that, SIFT and FLANN algorithms are used to match the photos with the pre-trained model, which is kept in the model.h5 file. The user ID and the proportion of the query image's fingerprints that match those in the pre-existing database are returned by the matching function.

If the match rate is greater than 80%, the user is signed in and all of their fundamental information is displayed. An error message informing the user that no suitable match was found is displayed if the match percentage is less than 80%.

CHAPTER 4: PERFORMANCE ANALYSIS

4.1. Dataset

600 African individuals' 6,000 fingerprints are included in the SOCOFing collection. Each subject has been fingerprinted ten times, and they are all at least 18 years old. Labels for gender, names for the hands and fingers, and other distinctive characteristics are all part of SOCOFing. Additionally, using the STRANGE toolbox, synthetically altered versions of these fingerprints are offered with three distinct levels of alteration for obliteration, central rotation, and z-cut. STRANGE is a brand-new framework for creating realistic synthetic changes to fingerprint scans. Over 500 dbi resolution images were changed using the STRANGE toolbox's easy, medium, and hard parameter settings.

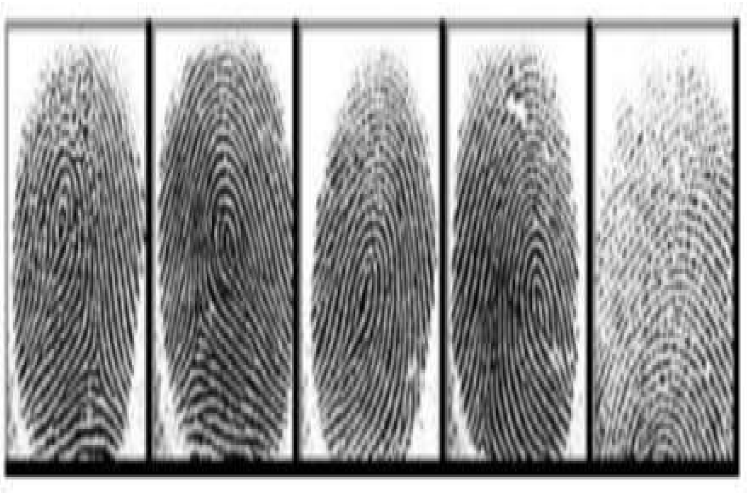


Fig 4.1 Sample illustration of five left hand fingerprints belonging to the same subject.

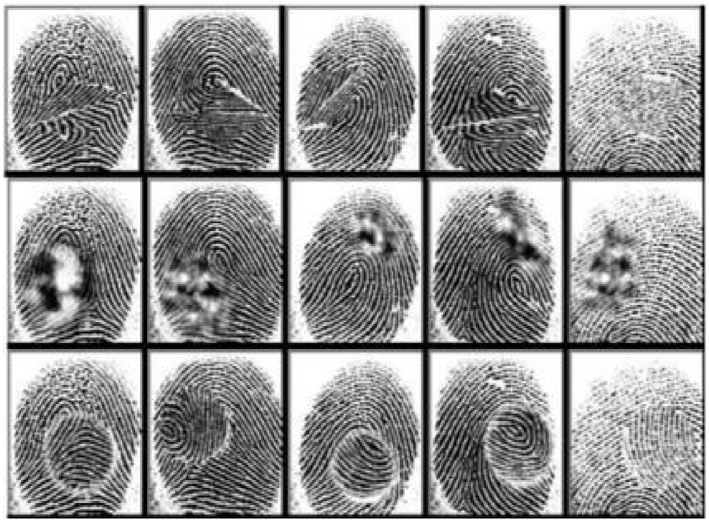
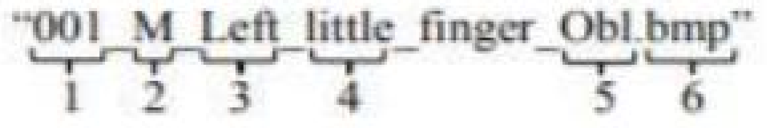


Fig 4.2 Images after being altered into z-cut, obliteration and central rotation.

Real, or original, and changed photos are separated into two subfolders in the collection. The revised folder is further broken down into three categories: easy, medium, and hard.

The file format provides the labels for each individual image and has the naming convention of:



"001_M_Left_little_finger_Obl.bmp"

Fig 4.3 Format of image stored

where:

1. Identifies the number of the subject: 001 to 600.

2. Indicates the gender of the subject: M – male, F – female.

3. Denotes the hand: Left or Right.

4. Indicates the finger name: little, ring, middle, index, or thumb.

5. Indicates the type of alteration type (altered images only): Obl – obliteration, CR – central rotation, or Zcut.

6. File extension: ".bmp" for all images.

4.2. Outputs

The output of the matched fingerprint is created when the required ML algorithm is applied to the input fingerprint, as illustrated below.

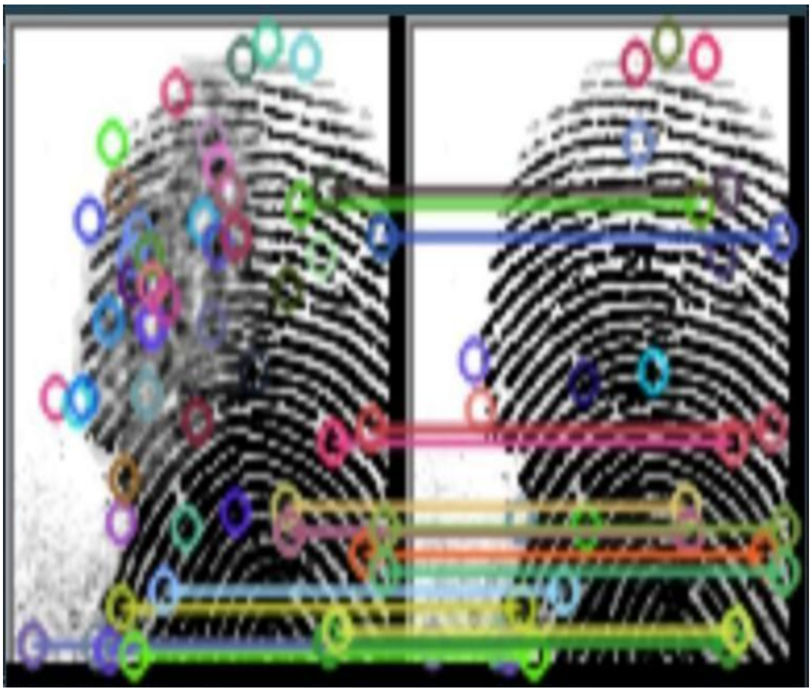
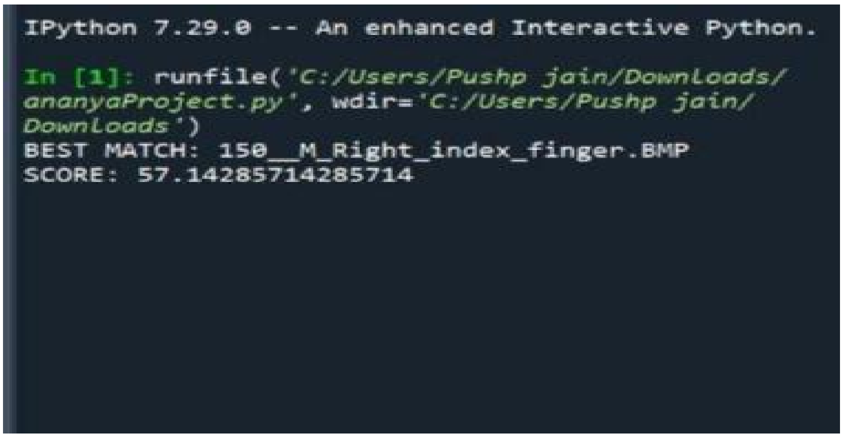


Fig 4.4 Point wise Matching for the input finger with the best match found



```
IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/Pushp jain/Downloads/
ananyaProject.py', wdir='C:/Users/Pushp jain/
Downloads')
BEST MATCH: 150__M_Right_index_finger.BMP
SCORE: 57.142857142857 14
```

Fig 4.5 Best Fingerprint Match with the respective score

CHAPTER 5: CONCLUSION

5.1. Conclusion

In this study, we proposed a secure architecture for biometric user authentication (or identification) that forbids the retention of data (like neural network weights or specific biometric data) that could be used by adversarial organisations to create entirely fictitious biometric inputs that can pass the authentication procedure. The suggested approach complies with the constraints on user credential maintenance imposed by that paradigm.

A strong and dependable system for user authentication is provided by the fingerprint matching project using the SIFT and FLANN algorithms with Django. The fingerprint matching procedure is accurate and effective thanks to the usage of SIFT and FLANN, and the Django framework gives developers a robust toolkit for creating aesthetically pleasing web apps.

The fingerprint matching algorithm can process the images quickly and compare them with the pre-existing fingerprint database by transforming the images from BMP format to binary byte format. The usage of Django for online deployment offers a scalable and safe platform for user authentication, with built-in security features and a big community for support. The project's overall results show the potential of machine learning algorithms and web frameworks for creating reliable user authentication systems.

REFERENCES

[1] Y. Dodis, L. Reyzin, and A. Smith, “Fuzzy extractors: How to generate strong keys from biometrics and other noisy data,” in International conference on the theory and applications of cryptographic techniques. Springer, 2004, pp. 523–540.

[2] S. Gootman, “Opm hack: The most dangerous threat to the federal government today,” Journal of Applied Security Research, vol. 11, no. 4, pp. 517–525, 2016.

[3] J. Taylor, “Major breach found in biometrics system used by banks, uk police and defence firms,” The Guardian, August 2019.

[4] M. J. Brady, “Biometric recognition using a classification neural network,” Apr. 6 1999, uS Patent 5,892,838.

[5] M. M. A. Allah, “Artificial neural networks based fingerprint authentication with clusters algorithm,” Informatica, vol. 29, no. 3, 2005.

[6] F. W. M. H. Wong, A. S. M. Supian, A. F. Ismail, L. W. Kin, and O. C. Soon, “Enhanced user authentication through typing biometrics with artificial neural networks and k-nearest neighbor algorithm,” in Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat. No. 01CH37256), vol. 2. IEEE, 2001, pp. 911–915.

[7] S. Yong, W. K. Lai, and G. Goghill, “Weightless neural networks for typing biometrics authentication,” in International Conference on Knowledge-Based and Intelligent Information and Engineering Systems. Springer, 2004, pp. 284–293.

[8] S. A. Nazeer, N. Omar, and M. Khalid, “Face recognition system using artificial neural networks approach,” in 2007 International conference on

Signal Processing, Communications and Networking. IEEE, 2007, pp. 420–425.

[9] Y. Tang, F. Gao, J. Feng, and Y. Liu, “Fingernet: An unified deep network for fingerprint minutiae extraction,” in 2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 2017, pp. 108–116.

[10] L. N. Darlow and B. Rosman, “Fingerprint minutiae extraction using deep learning,” in

2017 IEEE International Joint Conference on Biometrics (IJCB). IEEE, 2017, pp. 22–30.

[11] D. Menotti, G. Chiachia, A. Pinto, W. R. Schwartz, H. Pedrini, A. X. Falcao, and A. Rocha, “Deep representations for iris, face, and fingerprint spoofing detection,” IEEE Transactions on Information Forensics and Security, vol. 10, no. 4, pp. 864–879, 2015.

[12] J. J. Engelsma, K. Cao, and A. K. Jain, “Fingerprints: Fixed length representation via deep networks and domain knowledge,” arXiv preprint arXiv:1904.01099, 2019.

[13] W.-S. Jeon and S.-Y. Rhee, “Fingerprint pattern classification using convolution neural network,” International Journal of Fuzzy Logic and Intelligent Systems, vol. 17, no. 3, pp. 170–176, 2017.

[14] M. Hammad, Y. Liu, and K. Wang, “Multimodal biometric authentication systems using convolution neural network based on different level fusion of ecg and fingerprint,” IEEE Access, vol. 7, pp. 26 527–26 542, 2018.

[15] P. K. Nayak and D. Narayan, “Multimodal biometric face and fingerprint recognition using neural network,” International Journal of Engineering, vol. 1, no. 10, 2012.

[16] B. Stojanovic, A. Ne ´ skovi ˇ c, and O. Marques, “A novel neural network ´ based approach to latent overlapped fingerprints separation,” Multimedia Tools and Applications, vol. 76, no. 10, pp. 12 775–12 799, 2017.

[17] R. F. Nogueira, R. de Alencar Lotufo, and R. C. Machado, “Fingerprint liveness detection using convolutional neural networks,” IEEE transactions on information forensics and security, vol. 11, no. 6, pp. 1206–1213, 2016.

[18] A. Page, A. Kulkarni, and T. Mohsenin, “Utilizing deep neural nets for an embedded ecg-based biometric authentication system,” in 2015 IEEE Biomedical Circuits and Systems Conference (BioCAS). IEEE, 2015, pp. 1–4.

[19] V. Mai, I. Khalil, and C. Meli, “Ecg biometric using multilayer perceptron and radial basis function neural networks,” in 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2011, pp. 2745–2748.

[20] R. Salloum and C.-C. J. Kuo, “Ecg-based biometrics using recurrent neural networks,” in 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017, pp. 2062–2066.

[21] R. D. Labati, E. Munoz, V. Piuri, R. Sassi, and F. Scotti, “Deep-ecg: ˜ convolutional neural networks for ecg biometric recognition,” Pattern Recognition Letters, vol. 126, pp. 78–85, 2019.

[22] Madhuri and RichaMishr,” Fingerprint Recognition using Robust Local Features”,IJARSSE,Volume 2, Issue 6, June 2019, INDIA.,

[23] ManishaRedhu and Dr.Balkishan,Department,” Fingerprint Recognition Using Minutiae Extractar “,Applications (IJERA),Vol. 3, Issue

4,pp .2488-2497, Jul-Aug 2019,MaharshiDayanandUniversity , Rohtak, India.

[24] Sangram Bana1 and Dr.Davinder Kaur2," Fingerprint Recognition using Image Segmentation", IJAEST,Vol No. 5, IIT Roorkee, Roorkee.

[25] Ritu1 and Matish Garg2," A Review on Fingerprint-Based Identification System",Student, SBIET College, Pundri, Kaithal, India1Assistant Professor, SBIET College, Pundri, Kaithal, IJARCCE,Vol. 3, Issue 3, March 2020, India.

[26] 1Priyanka rani,2Pinki Sharma,"A Review Paper on Fingerprint Identification System ",IJARCST, Vol. 2, Issue 3 (July - Sept. 2021)Kaithal, Haryana, India.

[27] Gurpreet Singh1 and Vinod Kumar2," Review On Fingerprint Recognition: Minutiae Extraction and Matching Technique",IJISR,Vol. 10 No. 1,pp. 64-70, Oct. 2021, Punjab, India.

APPENDICES

1. Install the libraries

Cv2 refers to OpenCv2, useful for image manipulation, while OS is the library for accessing the user's operating system for directories and files.

```
import os

import cv2

#import eval #import metrics
```

2. Read a sample altered fingerprint image from the dataset

Here, we use the SOCOFing dataset which contains unique attributes for gender, hand and finger name. In addition altered fingerprint versions are provided with three levels of alteration: obliteration, central rotation, and z-cut. But note that not all fingerprints have synthetically altered versions.

Moreover, the file format consists of label for each image with a naming convention of:

"001_M_Left_little_finger_Obl.bmp"

To read the dataset, we need to use imread() from OpenCV.

```
sample=cv2.imread("C:/Users/Pushpjain/Downloads/archive

(5)/SOCOFing/Altered/Altered-
Hard/150__M_Right_index_finger_Obl.BMP")

# sample = cv2.resize(sample, None, fx = 2.5, fy = 2.5)

# cv2.imshow("Sample", sample)

# cv2.waitKey(0)

# cv2.destroyAllWindows()
```

3. Read the real fingerprint images

In this step, we use os.listdir() to scroll through the “real” images in the fingerprint dataset.

Real images are the unaltered fingerprint images in the dataset.

We used a for loop to read through the real images in the dataset. So, the purpose is to eventually compare the sample altered image with real fingerprint images in the fingerprint image variable. Hence, it looks for a match among the real fingerprint images.

```
best_score = 0

filename = None

image = None

kp1, kp2, mp = None, None, None


counter = 0

for file in [file for file in os.listdir("C:/Users/Pushp
      jain/Downloads/archive

(5)/SOCOFing/Real")][:1000]:

          fingerprint_image = cv2.imread("C:/Users/Pushp

jain/Downloads/archive (5)/SOCOFing/Real/" + file)
```

4. Creating a SIFT object

SIFT object is a Scale Invariant Feature Transform object. It is a feature algorithm that finds the image keypoints. In this case, the SIFT object detects the ridges in the fingerprints. According to the OpenCV docs, each keypoint is a special structure with many attributes like its (x,y) coordinates, size of the relevant neighborhood, the angle which specifies its orientation, and response that specifies the strength of keypoints, etc.

```
sift = cv2.SIFT_create()
```

5. Detect the kepoints and compute the descriptors

The SIFT object consists of a function that identifies and detects the descriptors and keypoints in any image. This is called detectAndCompute(). Pass two arguments: the image to read and None for keypoints as we want it to detect keypoints–not provide them. So, the keypoints and descriptors in both the fingerprint images (original and altered) are identified and computed. These features are scale and rotation invariant.

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

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

6. Finding the best match of distance between keypoints and descriptors

OpenCV contains tons of effective and efficient functions and libraries. One of these is the Flann Based Matcher. This matcher function performs a fast local approximate nearest neighbors (FLANN) calculation between two sets of feature vectors. The result is two NumPy arrays. The first one is a list of indexes of the matches, while the second one contains the values of match distances.

In this case, k=2 as we only compare two images at any time. Hence, it is a great function to speed up the process. An argument specifying the algorithm to use is passed as a dictionary. One specifies k-d trees, which organize points in k-dimensional space.

```
matches = cv2.FlannBasedMatcher({'algorithm': 1, 'trees': 10},

                    {}).knnMatch(descriptors_1, descriptors_2, k
                    = 2)
```

7. Matching the keypoints for finger authentication in python

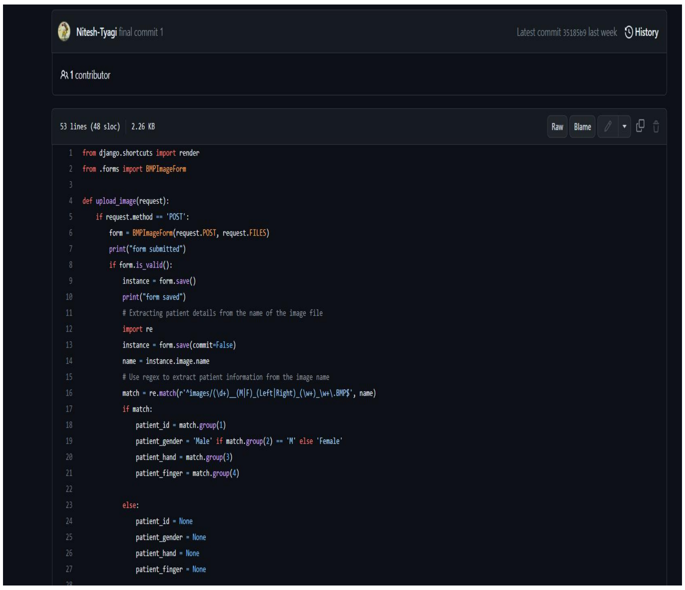
Matching the keypoints is perhaps the most essential step of this algorithm. Here, first, define an array to hold the matched points. Then, compare the distance between the matched points. After that, use conditional statements to assign keypoints to the length of the larger one among the two sets of keypoints.

```
match_points = []

for p, q in matches:
    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)

if len(match_points) / keypoints
   * 100 > best_score: best_score
   =        len(match_points)        /
   keypoints * 100 filename = file
   image = fingerprint_image
   kp1, kp2, mp = keypoints_1, keypoints_2, match_points
```

8. Viewing the fingerprint with drawn match points and the best match and score

This is the last step for this algorithm. Here, I have printed the file name that results in the best match and the best score. Next, apply a condition that if the match points are greater than 0 to see if the images have been read, then draw the match points using the drawMatches() method in cv2. After that, use imshow() to display the resultant image. The next two lines ensure the image window doesn't close instantly.

```
#print("BEST MATCH: " + filename)
print(f'BEST MATCH: {filename}')
print("SCORE: " + str(best_score))

result = cv2.drawMatches(sample, kp1, image, kp2, mp, None)
result = cv2.resize(result, None, fx = 4, fy = 4)
cv2.imshow("Result", result)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



```
from django.shortcuts import render
from .forms import BMPImageForm

def upload_image(request):
    if request.method == 'POST':
        form = BMPImageForm(request.POST, request.FILES)
        print("form submitted")
        if form.is_valid():
            instance = form.save()
            print("form saved")
            # Extracting patient details from the name of the image file
            import re
            instance = form.save(commit=False)
            name = instance.image.name
            # Use regex to extract patient information from the image name
            match = re.match(r'^images/(\d+)__(M|F)_(Left|Right)_(\w+)_\w+\.BMP$', name)
            if match:
                patient_id = match.group(1)
                patient_gender = 'Male' if match.group(2) == 'M' else 'Female'
                patient_hand = match.group(3)
                patient_finger = match.group(4)

            else:
                patient_id = None
                patient_gender = None
                patient_hand = None
                patient_finger = None
```