SUNWAY UNIVERSITY

Capstone Project Report

Smartphone-based Biometric System

by

Vincent Seaw
(17028754)

BSc (Hons) In Computer Science

:

Supervisor : Associate Prof Dr Lau Sian Lun

Date : 12 November, 2020

Project title : Smartphone-based Biometric System
Date : 11th November 2020
Student : Vincent Seaw
Supervisor : Associate Prof Dr Lau Sian Lun

Abstract

This research intends to develop a mobile app with a simple biometric authentication to help Non-government Organisation's (NGO) doctors to identify their clients who are stateless people. The objective is to test the mobile app's biometric authentication's accuracy and reliability to determine if the prototype is deployable. The mobile app has a few key components, the biometric authentication, the database implementation, and lastly flow of data and screens. The app is developed in Android Studio, while the biometric authentication has 2 components; facial recognition and fingerprint authentication, whereas the database is implemented using Firebase Database. The facial recognition's face detection algorithm is implemented using Firebase MLkit, whereas the recognition algorithm of the facial recognition is a self-defined algorithm that checks for the best similarity between 2 templates for facial recognition. Moreover, the fingerprint authentication is an additional security implemented via Android Studio's API FingerprintManager to authenticate the doctor's identity and allow the only doctor to edit the client's profile in the database. The results proved the prototype to be undeployable as the accuracy and reliability of the system is inconsistent as the number of registered users grows, changing from 55% to 60% to 40% as the number of registered users are 3, 5, and 10 respectively. In conclusion, although the system is working, but it can only serve as a foundation and further refinement is required to ensure the system is more consistent and deployable.

Table of Contents

List of Figures

List of Tables

1 Introduction

Stateless people are defined as someone who is not considered as national by any state under the operation of its law, hence they do not have a proper personal identification such as an IC (Identification card) or passport. This becomes a problem as stateless people have difficulty accessing basic rights like education, healthcare, and employment. The reason for the need of a solution to this problem is due to the fact that NGO (Non-Government Organizations) whose aim is to help these stateless people, find it difficult to keep track of their clients (stateless people). Especially for NGO that provides healthcare, it is essential for them to know the medical record details of their clients and how often they have visited to ensure they provide proper healthcare equally to every client they get. In addition, this project's context will be towards helping NGO that provides healthcare to stateless people, where the doctor's smartphone will be the one having the proposed system.

1.1 Project Objectives:

Design and implement a system that is efficient for personal identification based on a common smartphone's available technology and possible biometrics. Hence, a simple database will be implemented for storing the client's data and their corresponding biometrics. Furthermore, appropriate comparisons between different types of biometrics as fingerprint and facial recognition will be performed, to help determine the best possible outcome for the final system. Thus, in short, the system is meant to be compatible on a smartphone, with the capability of performing biometric authentication, and to store the client's data on a cloud database. However, the system will not consider ways to improve the security of the biometric authentication. For example, liveness testing to catch biometric spoofing.

1.2 Overview of Proposed System:

The system's reliability and accuracy will be measured via a simple way of calculating the Equal-Error Rate (EER), whereby no actual formula will be involved but only observation from the data shown on the plotted graph (rate against security). In addition, Euclidean distance will be used to measure the distance between 2 features captures on the image, to create the biometric template for authentication. Moreover, the system will be developed in Android Studio and make use of appropriate tools to complete the system, such as MLkit for face detection to perform biometric facial recognition, Firebase Database to implement a simple database, and FingerprintManager API to perform fingerprint authentication for the doctor as the doctor intends to update the client profile. Results show that the prototype is functional and working but the performance is not good enough to be considered deployable due to the inconsistency during the facial recognition authentication. Hence, in conclusion, the prototype is not deployable, and more refinement is required on the face detection algorithm as it is the root problem for the prototype's inconsistency.

1.3 Project Integration:

This project is meant to be focused on the biometric authentication with the implementation of a simple database. However, the database is much more complicated in the real world, due to the vast amount of information stored in the database, security concerns, and how they are all associated with each other via primary keys and foreign keys relation. Hence, another project that

is related to solve the problem mentioned in the introduction is to implement a proper database that is blockchain based, where the personal information of the clients will be originating from this project, as it authenticates the user then extracts the corresponding information from the database, and it can also register new users then store their information together with their biometrics at the database.

2 Literature review

To develop a smartphone-based biometric system and resolve the issue mentioned in the introduction, it is essential to understand the technical parts of biometrics to gain an idea regarding a biometric system's reliability and security. Hence, a few papers will be reviewed in this section to provide an insight about the technicality of biometric system. Then, the strengths and weaknesses of each paper will be pointed out in their corresponding paragraphs to create a link between each paper while concluding the technical needs of the biometric system at the very end.

2.1 Measuring a Biometric System's reliability

First, the paper "Biometric Authentication for Mobile Devices" written by Clarke, Furnell, and Reynolds on January 2002, gives a very good idea about the need of biometric authentication in mobile devices. Where the number of issues of mobile devices theft are mentioned, then lists out the possible biometric authentication approaches and implementation, as well as the idea of measuring each biometrics' reliability [1]. However, as noted the paper's date is rather obsolete, but it still helps with this project as it explains the basic measurement needs for a biometric system's reliability. All biometrics work by comparing the current biometric data with a known template stored in the database, and that is the basic idea of authentication a user by using a biometric. Hence, the template stored is the biometric data that the user gives during the user's enrolment on the system. However, this simple idea creates a performance issue, where the authors use 2 variables to measure the error rates, 1 being False Acceptance Rate (FAR) and the other being False Rejection Rate (FRR). The FAR shows the rate at which an imposter is accepted by the system, while the FRR shows the rate at which the authorized user is rejected by the system, figure 1 illustrates an example of this relationship [1].

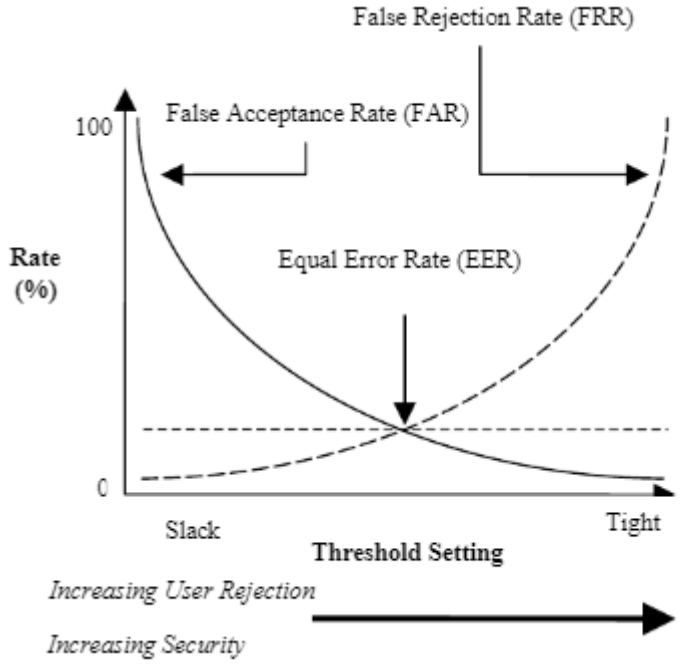

Figure 1: Mutually exclusive relationship between False Acceptance & False Rejection Rates

As shown above, a trade-off situation exists whereby the higher the security, the higher the number of times the authorized user is rejected by the system, but if the security is lowered to accommodate

this issue, then the number of times an imposter is accepted by the system will be higher [1]. Hence, a third variable known as Equal Error Rate (EER) is introduced. Whereby it is the point both error rates intersect, and this is used in the industry as a relative measurement between different biometric approaches and their reliability and security [2]. In addition, it is also important to mention that the lower the EER the better is the reliability and security of the biometric system [3]. However, even though one of this paper's strength is that it shows the basic idea about biometric systems and their security measurements, but it does not specifically explain how one increases the security of a biometric approach. It can be assumed in the case of a higher security, the system's algorithm will be more strict and check for a higher similarity when comparing between the current biometric data and the known template in the database, whereas in the case of a lower security, the system's algorithm will be more lenient and check for a lower similarity when comparing between the current biometric data and the known template in the database.

2.1.1 Additional Security for a Biometric System

However, those are assumptions. Hence, the paper "Security of Biometric Authentication Systems" by Matyas, and Zdenek will be discussed now to address the issue, as this paper researches on the various types of ways a biometric system's accuracy can be affected and how to increase the biometric system's security [4]. Based on this paper "Security of Biometric Authentication Systems", an interesting possibility that greatly affects the accuracy of a biometric system is discovered, that is the variability of biometric characteristics, the biometric characteristics itself plays a huge role in determining the reliability of the biometric system, since it depends whether the characteristic is genotypic or phenotypic. Genotypic characteristics are not subjected to change, so they do not change over time, meaning the matching algorithm does not have to adapt to changes [4]. However, the downside of this characteristic is that it cannot identify the difference between monozygotic twins, meaning that both individuals who are monozygotic twins will be accepted by the system, so the percentage of monozygotic twins in a population sets the lower limit for the FAR of the biometric system [4]. On the other hand, phenotypic characteristics is the opposite, meaning the matching algorithm have to adapt to changes, so it does not set a lower limit on the FAR, but it does set a lower limit on the FRR, and this scenario is also known as phenotypic error rate [4]. More importantly, the performance of biometric techniques that involves the variability of biometric characteristics are determined by 2 kinds of variability, that is within-subject variability and between-subject variability [4]. For within-subject variability, the results show that the biometric measurements are never the same, so the biometric system must know how to accept similar biometrics stemming from one biometric characteristic stored in the known template as a true match [4]. Hence, it shows that the matching algorithm for within-subject variability allows different input measurements, and this creates a higher false reject, making within-subject variability to set the lower limit on the FFR when within-subject variability is higher [4]. On the other hand, between-subject variability shows that when this variability is low, it will be harder to identify the difference between 2 subjects and a false accept may occur, so the lower the between-subject variability the higher the FAR. Therefore between-subject variability sets the lower limit for the FAR [4]. More importantly, the authors of this paper mentioned that "an ideal biometric characteristic impacts very high between-subject variability" [4]. This is the case since the higher the between-subject variability, the lower the FAR of the biometric system, and it is assumed that it indirectly means that it lowers the FRR of the biometric system as well. From this discussion, one of the strengths of this

paper is the elaboration on the matching algorithm based on the variability of the biometric characteristics, which helps to understand the accuracy of a biometric system. However, one weakness can be spotted from this paper is the lack of elaboration on the implementation for the corresponding matching algorithms. It creates the question how one fully implements a matching algorithm with a high between-subject variability. It can be assumed that it is done by accepting multiple templates from the new user during enrolment, and from the multiple templates find the similar points and create a new template from that, then compare the current biometric data with the new template to authenticate if the use is authorized or not, creating a lower FAR and indirectly a high between-subject variability.

Furthermore, the paper "Security of Biometric Authentication Systems" by Matyas, and Zdenek also introduced another interesting way of increasing the biometric system's security, that is by implementing liveness into the biometric system where the authors refer it to as the liveness problem [4]. The authors introduced this since it is naïve to assume that attackers will never obtain the biometric data and perform a remote authentication and access the system [4]. Hence, by implementing liveness test into the biometric system, remote attacks are not possible, since liveness ensure that the data are processing a fresh biometric data originating from the person being authenticated [4]. In general, liveness tests are a form of additional security in the biometric system that can be divided into 2 categories, one is static tests while the other is dynamic tests [4]. Static tests are some physiological characteristics measurements that helps to identify whether the biometric data is a living human or an artificial fake [4]. Whereas, dynamic tests verify the reaction a person has to an impulse, in the case of a pupil contraction due to the light intensity, the difference in light intensity is the impulse while the pupil contraction is the reaction [4]. Therefore, it is safe to assume that these liveness helps to improve a biometric system's security. More importantly, examples of liveness tests in fingerprint is measuring the temperature, pressure stimulus and other electrical properties [5], while examples of liveness tests in facial recognition systems either uses several cameras to obtain 3D, namely the depth of the picture to avoid attackers from using pictures to infiltrate the system, or request the subject to perform an action with the head, namely blinking the eyes or open the mouth [4]. Moreover, the authors even addressed the issues of liveness tests, claiming that it is weak and can be fooled by the attackers just by using materials that can pass the liveness test [4]. This is justified by the paper "Impact of Artificial "Gummy" Fingers on Fingerprint Systems" by Matsumoto, Yamada, and Hoshino, whereby it shows that it is easy to fool common fingerprint scanners with silicon or gelatine copies [6]. Hence, although it increases the security of a biometric system, but attackers who are willing to do something as trivial to fool the biometric system, then it would mean the biometric system is not entirely fool proof. However, a few biometric sensors are believed to be moderately spoof-resistant [4]. Even though this paper [4] has such a thorough discussion, showing that liveness tests to improve biometric system's security is one of this paper strengths, but the weakness to it is that there is not proper evidence in the paper itself that shows biometric sensors with moderate spoof-resistant, it is merely a belief. Thankfully, evidence shows that with the evolution of technology on liveness tests and adopting different approaches such as live facial recognition, this is no longer a concern [7].

Based on the information gathered so far, it is good to use the FAR, FRR and EER to measure the reliability of the system. Together by implementing either a genotypic or phenotypic with a high

between-subject variability matching algorithm to increase the biometric system's accuracy. In addition, due the liveness tests having hardware specifications, it seems difficult to implement a proper liveness tests, but it is still a very good idea to implement it and further increase the security of the biometric system to avoid biometric spoofing from attackers.

2.2 Fingerprint Authentication and Facial Recognition

More importantly, the 2 commonly used biometrics in majority of the system nowadays are fingerprint authentication and facial recognition[8]. Hence, the remaining part of the literature review will be focusing on the implementations of fingerprint authentication and facial recognition biometric systems.

2.2.1 Fingerprint Authentication

This research paper "Fingerprint Biometric Systems" studies the different implementations of fingerprint biometrics and compares each approach to find out the best implementation for a system [9]. The authors of this paper [9] discussed about the features that can be extracted from a fingerprint image create a template for authenticating a user, these "features combined" will create a unique template as it varies from one person to another, making each person to be different and distinguishable. These "features combined" can be considered as fingerprint patterns in a more general view, whereby there are 3 commonly known fingerprint patterns, it can be an arch, a loop, or a whorl, figure 2 shown below illustrates how each fingerprint pattern looks [9]:

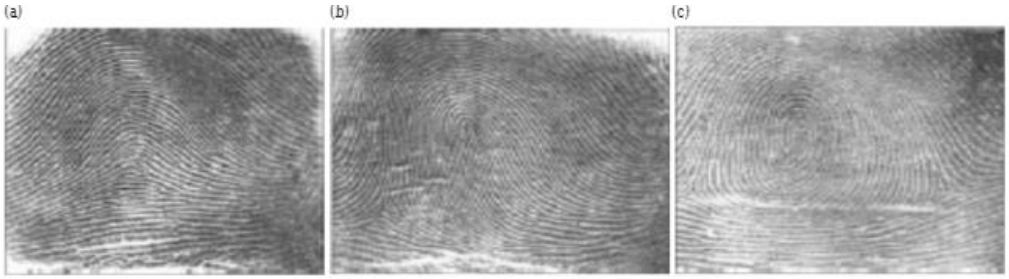

Figure 2: (a) Arch, (b) Loop, (c) Whorl

Based on the figure above, these fingerprint patterns are too broad to be taken as a feature for authentication, hence the term "Features combined" mentioned previously refer to a more specific feature that can be extracted from a fingerprint pattern, and they are commonly known as minutiae points, including a ridge ending, a bifurcation, and a short ridge, to get a clearer insight regarding each of these features, figure 3 shown below illustrates how each feature looks like [9]:

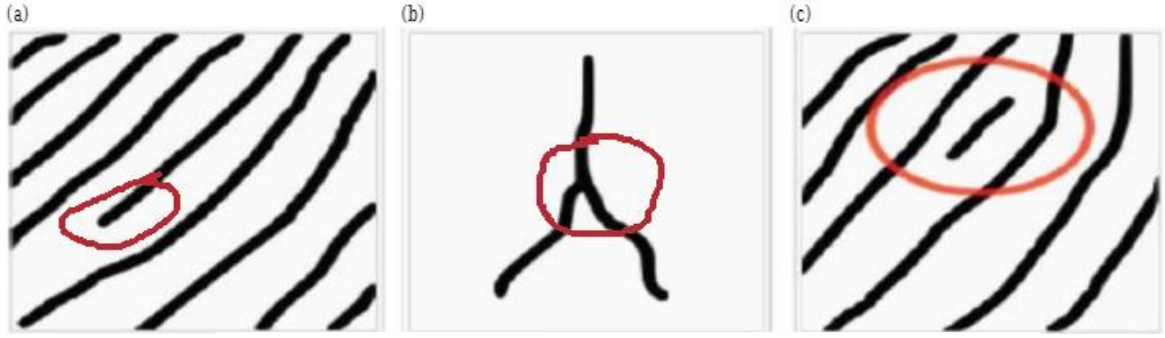

Figure 3: (a) Ridge Ending, (b) Bifurcation, (c) Short Ridge

Based on the figure above, each of these features are unique from one another, providing enough distinction to help identify a person, hence, these are the features that are commonly used in a

fingerprint template to help identify a person for fingerprint authentication [9]. The first method that the authors of [9] researched on is Components Edge Detection, this method is proposed because an issue occurred during the detection of similarities for fingerprint that are related to discontinuities, spots, and independent ridges, etc. Hence, this method intends to create connected boundary components by using the minutiae points, to produce a line drawing of an image that can be used as a fingerprint template to identify a person [9]. Moving on, another method that the authors of [9] decided to take a look at is Automated Identification System (AFIS), the algorithm that the authors took sample of is mainly about detecting the edges, storing it as a template for authentication, but before the edges are detected, the fingerprint image will undergo a few stages of pre-processing, whereby it is first turned into grayscale, then undergo grayscale normalization, then it will go through thinning process and then the final edge detection, the edge detection is done by using convolution [9]. Furthermore, another method is the Markov Model, the existing biometric system combined both algorithms known as Bayes and Henry classifier, but the system's difficulty was demonstrated, and it takes time to process data during the real time implementation stage [9]. Hence, the sampled algorithm for the Markov Model approach uses the pseudo 2D Hidden Markov Model (HMM), whereby this model captures all the different types of fingerprints separate states with different level of Markov Chain, then HMM is used during the recognition stage, whereby it checks every super states of the fingerprints to detect which types of fingerprint can be used to match the given fingerprint image with the fingerprint template stored in the database [9]. Moreover, the last method that was studied by the authors of [9] uses Discrete Fourier Transform (DFT) and Non-linear Discriminant Analysis (NDA), hence the name of the method is Discriminant Analysis, this method reconstructed the necessary images using DFT and NDA and then applied to the new image, it is also important the mention that NDA was used to extract the features for the template. However, even with all these different implementations, the authors could not reach a conclusion which implementation was better than the other, since all the implementations show similar results, but the authors did mention that the best method that can be used for the matching process in fingerprint biometric systems are minutiae-based and correlation based, whereas Euclidean distance-based is more suitable for the feature extraction [9]. Hence, this brings the next paper known as "Biometric Recognition System (Algorithm)", whereby it is a minutiae-based fingerprint recognition algorithm with the implementation of Euclidian distance based for feature extraction and for comparison during authentication [10]. There are total 20 steps in this algorithm based on the block diagram, whereby step 1 is the input image, steps 2-5 is pre-processing to better results before applying the main algorithm to the image, steps 6-16 is composed of the main algorithm, including the core-skeleton of the algorithm and a few more image processing during the main algorithm, in order words, feature extraction is performed during the main algorithm, then step 17 is the fingerprint template creation based on the extracted feature, and then stored into the fingerprint database, lastly steps 18-20 is comparing the stored template with the given template and return the matching decision [10]. The overview of the entire algorithm's steps is illustrated below in figure 4 [10]:

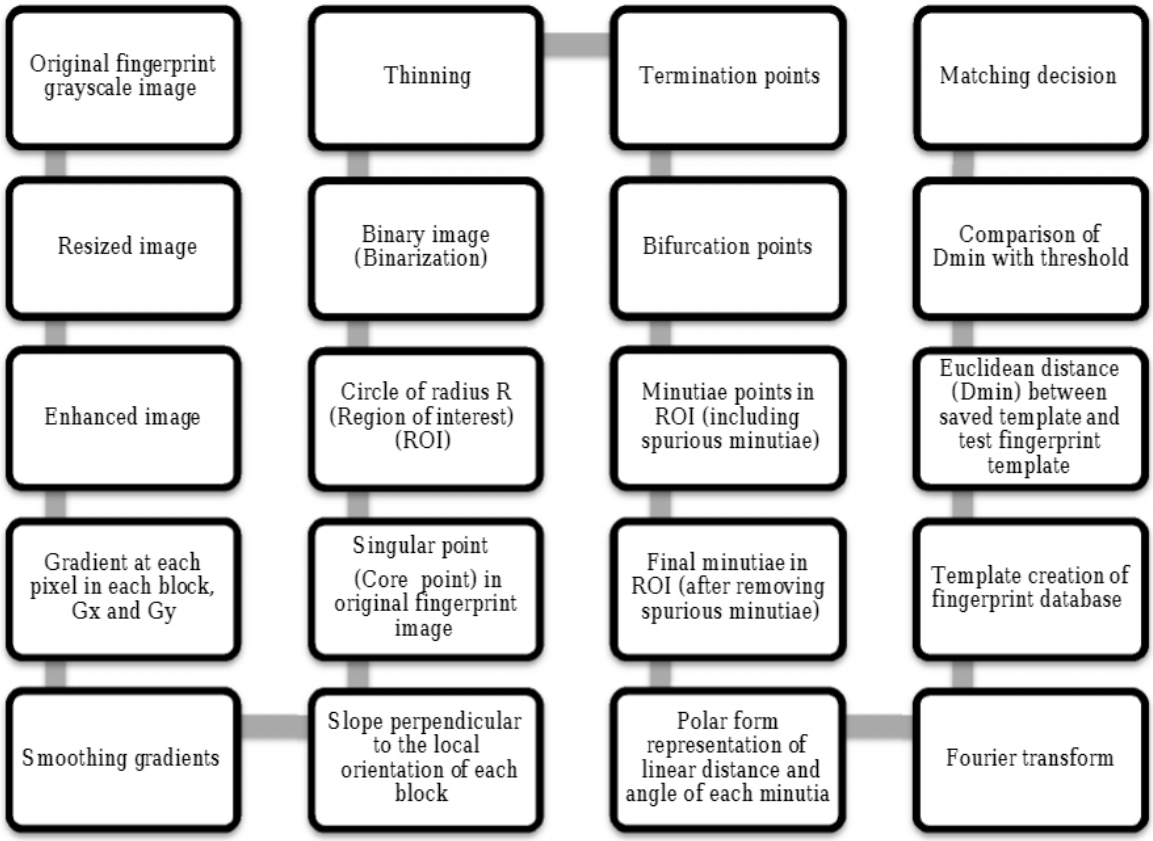

Figure 4: Overview of Minutiae-based Fingerprint Recognition Algorithm

Thus, after reviewing 5 different implementations for fingerprint biometric authentication, table 1 shown below provides a general insight regarding each method and their respective technique(s) used [9] [10]:

Method Name:	**Technique(s) Used:**
Components Edge Detection	• Digital Image Processing
Automated Identification System (AFIS)	• Digital Image Processing
Markov Model	• Bayes and Henry classifier • 1D Hidden Markov Model • 2D Hidden Markov Model
Discriminant Analysis	• Discrete Fourier Transform (DFT) • Non-linear Discriminant Analysis (NDA)
Minutiae-based Fingerprint Recognition Algorithm	• Digital Image Processing • Euclidean Distance • Discrete Fourier Transform (DFT)

Table 1: General Insight of each Fingerprint Biometric Authentication

2.2.2 Facial Recognition

On the other hand, based on the paper "A Review of Person Recognition Based on Face Model", a general overview of facial recognition process can be illustrated and is shown in figure 5 below [11]:

Figure 5: General Overview of Facial Recognition Process

However, there are a lot of different types of facial recognition algorithms, and only 2 techniques will be reviewed, one being a Model-Based technique, and another a being a machine learning-based technique. Model-Based techniques are face recognition methods that utilizes model-based strategies to develop a person's face model that can be used to extract facial features, this technique is good due to its advantage being invariant to lighting, size, and alignment [12]. In addition, another advantage of Model-Based technique is the ability to perform rapid matching and provide a compact representation of face images [13]. However, the disadvantage of Model-Based technique is the process of detecting a face is complex and not easy to be implemented [14]. Whereas according to the paper "Image-based Face Detection and Recognition", a machine learning algorithm is implemented for facial detection and recognition [15]. The first step in this algorithm is to perform face detection on the dataset face images, then from the detected faces images, background subtraction is performed to extract the face out from the images, next the extracted face from the background image will undergo pre-processing to form the dataset that is required for the training and query. Lastly, the trained classifier will then perform the facial recognition [15]. Figure 6 shown below illustrates the machine learning-based face recognition system overview [15]:

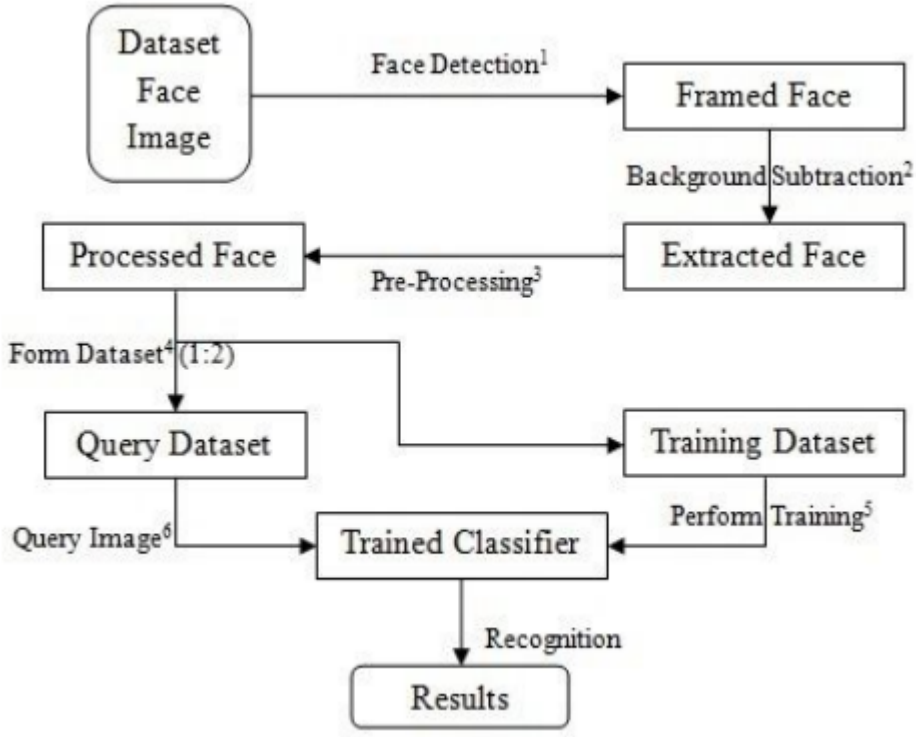

Figure 6: Machine Learning-based Face Recognition System Overview

Although it is more accurate as machine learning is utilized, but this creates a problem as machine learning algorithms require a huge amount of dataset to be utilized effectively, as this problem originates from the idea of the larger the data size, the better for the machine learning [16]. In addition, another issue that arises is the fact that machine learning algorithms are not easy to be implemented and it often requires huge computational efforts, hence the authors of [15] reduces the computational effort in the system by converting the face images into a set of basic functions which represents the principals attributes of the face images seek directions in, as it is assumed the face will always be upright and frontal for the face images.

Now, table 2 will show the difference between the 2 techniques of facial recognition discussed earlier:

Face Recognition Technique:	**Advantages:**	**Disadvantages:**
Model-Based Technique	• Invariant to lighting, size, and alignment [11]. • Ability to perform rapid matching and provide a compact representation of face images [13].	• Process of detecting a face is complex and not easy to be implemented [14].
Machine Learning-based Technique	• More accurate	• Requires large dataset [16]. • Demands huge computational effort

Table 2: Different Facial Recognition Technique Advantages and Disadvantages

Based on Figure 7 above, both techniques have their advantages and disadvantages and it is believed that these techniques will both shine at different situations depending on how the system is used.

2.3 Literature Review Conclusion

In conclusion, the simplified way of measuring a biometrics system's reliability is through is Equal Error Rate (EER), it can be obtained from the False Reject Rate (FRR) and False Acceptance Rate (FAR) of a biometric system. FRR and FAR are obtained through a rate against security graph. It was also discovered that biometric systems can provide greater security with liveness testing, such as pressure and temperature on the sensor for fingerprint recognition to ensure it is a live person and not a fabricated spoof fingerprint. However, these greater security from biometric systems will not be implemented into the project, whereas EER, FRR, and FAR will be used to measure the project's prototype's reliability and accuracy. Then, different implementations of fingerprint biometric and facial recognition were reviewed, and it is highly involved with computer vision and digital image processing, to extract the features and help classify the respective biometrics. For example, fingerprint features such as Ridge Ending, Bifurcation, and Short Ridge to create the fingerprint biometric template. Whereas another technique will involve more processing done to the image before extracting the features directly. These reviews provide a great insight to implement fingerprint authentication. However, due to time limitation, fingerprint authentication will not be implemented. On the other hand, for facial recognition, the facial features are extracted based on the technique that was used. For example, a model-based technique utilizes an algorithm to develop a

human face and extract the facial features from it, while another technique that is more machine learning oriented will require huge datasets to extract a more unified facial features for the machine learning algorithm to be accurate at perform facial recognition. However, each technique shines differently, so the prototype is going to combine both techniques for facial recognition, whereby a machine learning algorithm will be used to perform face detection only, and from the detected face, little like model-based technique of extracting facial features, the facial features will also be extracted from the detected face to create the facial biometric template. The reason for this combination is to be practical, as a pure machine learning facial recognition system will require the doctor to capture at least 100 of photos of the client, just to register them into the system. Hence, by removing the recognition part in machine learning, and create a self-defined recognition algorithm based on the extracted facial features, the system will become much more practical and convenient to both the doctors and the clients.

3 Goals

A more general and abstraction project goal, that is long-term, and this project is meant to contribute towards part of the goals:

- Help Non-Government Organisations (NGO) to identity stateless people.
- Help NGO doctors to keep track of their clients that are stateless people.

4 Objectives

Although the introduction has provided some of the project objectives, but this section is meant to show what the project objectives are:

4.1 Objective 1

Create a mobile app prototype that has the following functionalities:

1. Perform facial recognition biometric authentication for clients.
2. Perform fingerprint biometric authentication for doctors before entering “update client profile” page.
3. View client’s profile.
4. Update client’s profile
5. Register new client’s facial biometric data along with basic information such as name, age, etc.

4.2 Objective 2

Plot the Equal Error Rate (EER) chart to measure the accuracy and reliability of the prototype’s facial recognition biometric authentication.

4.3 Objective 3

Determine if the prototype is deployable through few EER charts and some In-house validation. The EER chart will help determine if the facial recognition biometric authentication component of the prototype is deployable, whereas the In-house validation is meant to determine if the entire app in terms of database’s data insertion, data extraction, data alteration, page flows, and data flows from screens are working correctly to determine if this component of the prototype is deployable.

6 Scope of Work

This prototype is meant to be focused on solving the biometric authentication part of the problem description. However, it is a simple solution that does not entirely reflect a true solution for the problem, but part of it. Hence, this section states the scope of the work for the prototype in terms of its biometric implementation, and database implementation.

6.1 Biometric Implementation

Although both fingerprint and facial recognition is implemented. However, due to hardware limitations on common smartphones, additional security such as liveness testing to catch biometric spoofing, and iris recognition is not implemented. In addition, fingerprint authentication will only be used for authenticating the doctors to ensure that it's an authorized figure, a doctor who is altering the data in the database, whereas facial recognition will be used for authentication the client (stateless people, the problem this project is trying to solve).

6.2 Database Implementation

Since this project is meant to be focused on the biometric implementations, and another project done by another student will be focused on the database part of the problem description. So, the database implementation is done in a very simple manner, whereby no actual medication records will be implemented as part of the database, but just general information will be stored in the database, together with the client's corresponding biometric templates. Therefore, the database can perform data insertion, data alteration, and data extraction at the corresponding pages that is meant to have such features, but not much information will be stored in it that reflects a medical record used by hospitals.

7 Methodology

To develop a smartphone-based biometric system, it would only make sense to implement fingerprint biometric as one of the biometric authentications. Since it is a must for a smartphone of this current modernized age to be capable of fingerprint biometric authentication, otherwise the smartphone would lose one of its crucial security for protecting the user's data [1]. Hence, a built-in API from Android Studio known as Fingerprint Manager is used to implement fingerprint biometric into the system. Fingerprint Manager allows the developers to gain access to the hardware and perform fingerprint biometric authentication based on the fingerprint templates that are stored in the phone's built-in security's database [2]. Hence, the challenge here is to extract the fingerprint template out from the phone's built-in security's database and store the fingerprint template into the designated cloud storage database for biometric authentication via comparing the stored fingerprint template in the cloud database with the live fingerprint template undergoing authentication. Unfortunately, this built-in API from Android Studio does not allow the developers to extract and store the fingerprint templates, since it is stored by the android system within the phone at a secured location that is not accessible, to overcome this issue, the developers have to either write a new code to access the hardware and get the fingerprint template or use a third party fingerprint scanner with compatible Software Development Kits (SDKs) to get the fingerprint template [3]. Hence, due to time constraint, Fingerprint Manager's API is used as a form of security check for the doctors when they intend to update the client's profile after treating them, and no fingerprint authentication will be performed for the clients.

Moving on, facial recognition was the next implementation for this system, whereby the system will first detect a face from the live camera, and then perform facial recognition based on the facial features from the current live image and the facial features that was stored in the database, and this is possible with the implementation of Firebase MLKit. Firebase MLkit is a tool in Android Studio that utilizes Google's machine learning expertise to Android and IOS apps in an easier manner, by using the right features, Firebase MLKit is capable of machine learning activities such as text recognition, and face detection etc [4]. More importantly, by making use of additional features implemented within Firebase MLKit, face recognition is possible as Firebase MLKit is also capable of extracting landmark (facial features) from the face, and by doing so facial recognition is possible at this point but more processing is required [5].

Hence, an algorithm is created to calculate the Euclidean Distance between specific facial features, this helps create a higher accuracy biometric authentication, since now the distance is compared with a +-10% difference from the stored distance in the database, instead of comparing the x and y coordinate of the facial features directly. Furthermore, the system's biometric authentication will match the most similarly matched facial features. For example, Initially the system will only look values that fall within the +-10% difference, then it will be reduced to +-9% difference for the current match, and if the current match falls within the +-9% difference, then it will be reduced to +-8% difference, and if the current match does not fall within the +-8% difference, then it will stop shrinking the boundary, and the next match will be check if it falls within the +-8% difference, if it matches then the boundary shrinking happens again, and this is how the recognition algorithm will keep shrinking the boundary to obtain the best matched template. Then, the system will return the

id of the matched template, and with a total of 12 features, each feature will return a valid id if a matched templated is found, and if no matched templated is found, a special id will be returned to indicate unmatched template. The system will determine if the user is authorized when there are 6 or more identical id returned. However, a trial and error testing will be conducted with these factors:

- Number of identical id returned
- Number of clients stored in the database

Hence, by adjusting these few factors, the accuracy and the equal error rate (EER) for the system can be determined to help identify if the system is acceptable towards this project's goal, whereby a lower EER will be better as it creates higher security and accuracy [5]. Furthermore, the camera size is adjusted to help ensure the difference between the stored template and the current live face for authentication will not be big, providing a more consistent capture.

On the other hand, Firebase Database which another tool in Android Studio is used to act as a cloud database for the biometric authentication and extract a simplified client's profile corresponding to their biometrics.

So, by combining the implementations above, the data flow of the entire system is in figure 7, while overall flow of the entire system is shown in figure 8 below:

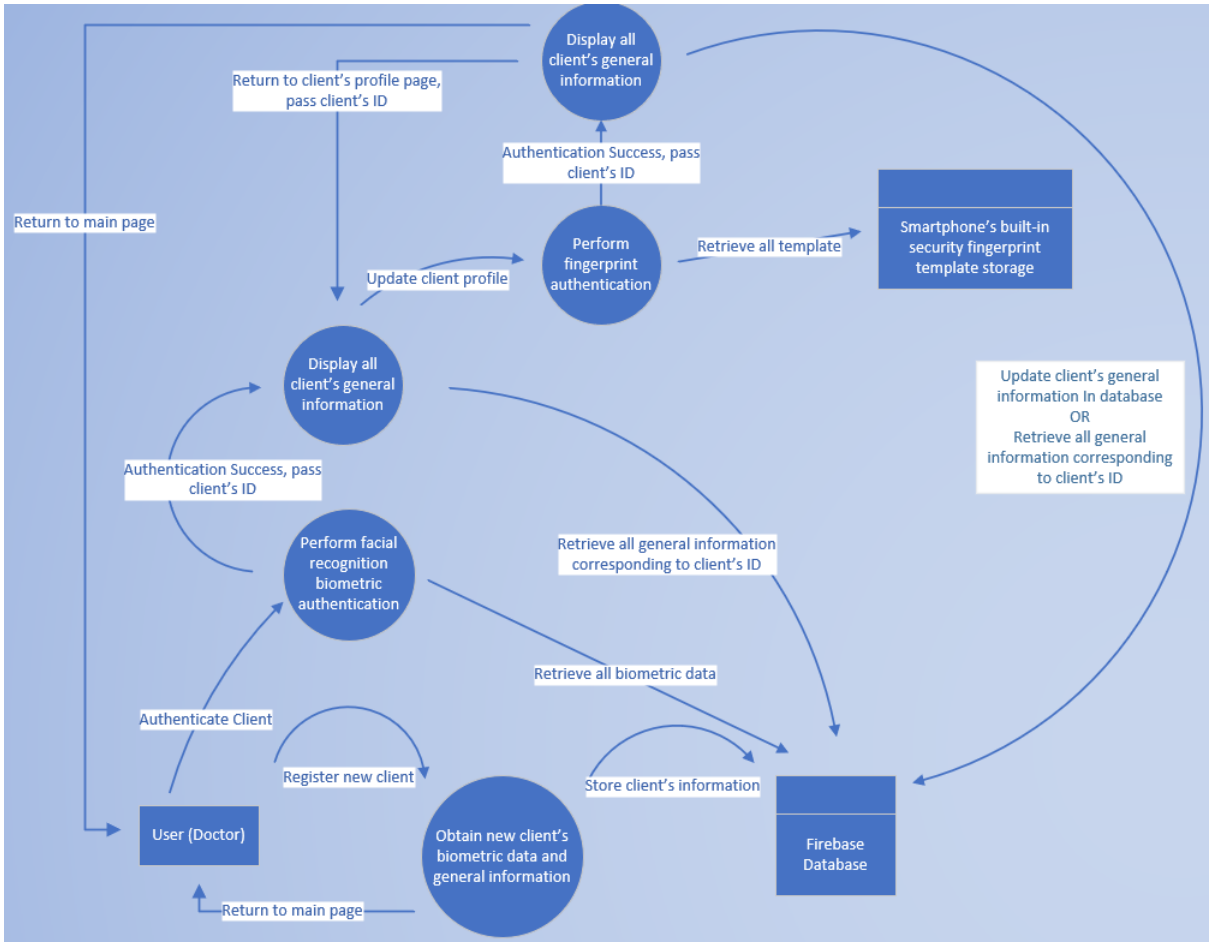

Figure 7: System's Data Flow Diagram

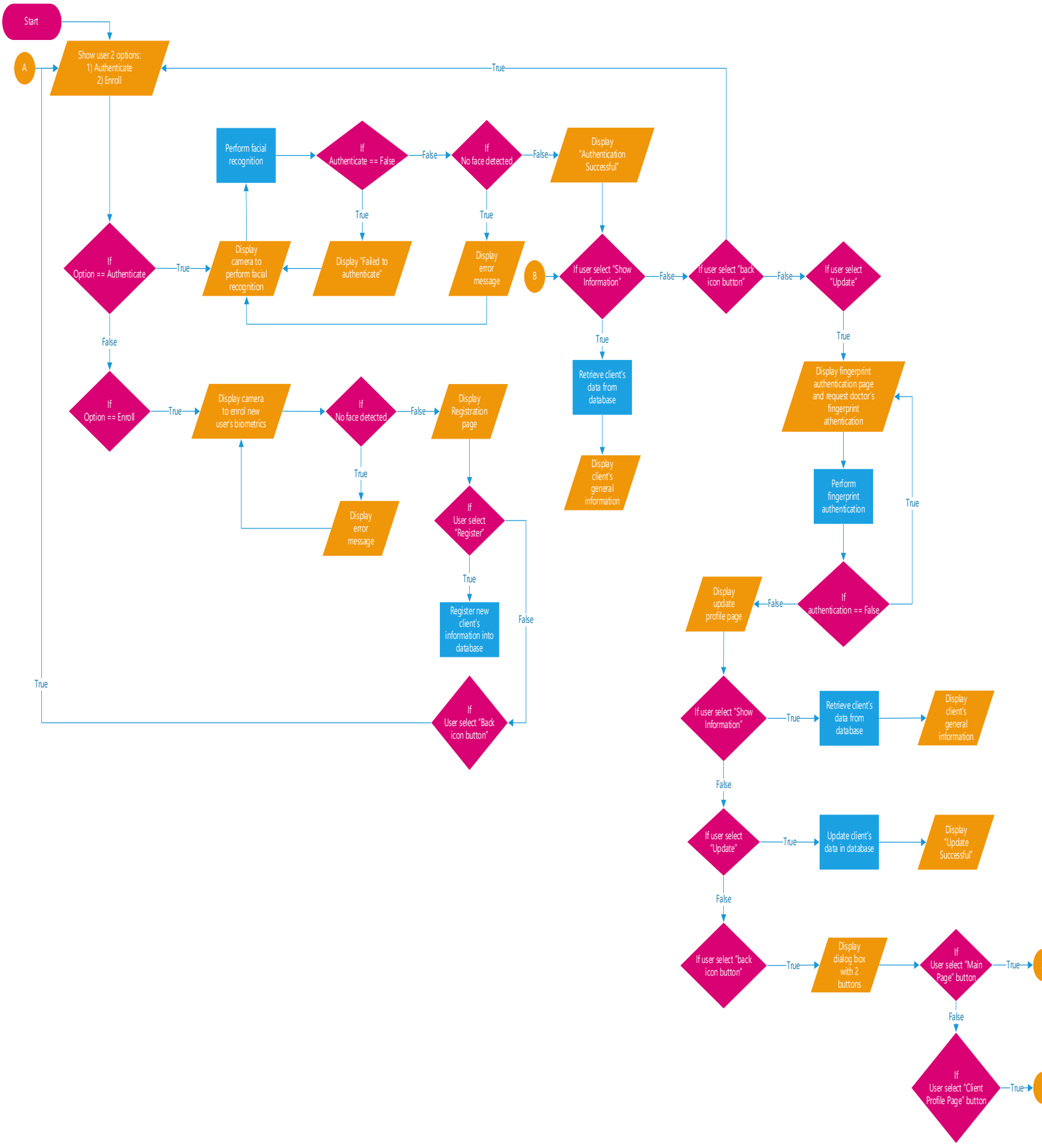

Figure 8: System's Flow Chart

7.1 System Evaluation

3 graphs of rate against security was plotted to obtain the False Rejection Rate (FRR), False Acceptance Rate (FAR), and Equal Error Rate (EER), whereby each chart will have different number of users stored in the system, to help evaluate if the system's performance will be consistent or deteriorate when the number of users increase.

In terms of rate definition in the chart, it is defined as out of 10 attempts, using a data that is not registered in the database, how many attempts are accepted, to obtain the FAR on that security level, then using a data that is registered in the database, how many attempts are rejected, to obtain the FRR on the security level, on a special note, if an authorized user is allowed to access the system after the biometric authentication but the data retrieved is false as it belongs to another user in the system, then this special scenario will be counted as 1 FAR.

Whereas for security level, it is defined as the number of matched id's, initially the system will check for 0 same id's or more and consider the data as an authorized user, then for the next security level, the system will check for 1 same id's or more and consider the data as an authorized user, and the security level will keep increasing until 12. However, if there is data redundancy, then those redundancy will be removed. For example, if security level 0 to 2 provided the exact same results, then it will be trimmed to security level 2 as the starting point for the x-axis instead of security level 0, vice versa for security level 10 to 12. On the other hand, the intersection between the FAR and the FRR on the graph will be used to obtain the EER, a variable that helps estimates the reliability of the system. In theory, a lower EER corresponds to a more reliable system, whereby the accuracy of the system is obtained via the distance of the intersection point to the maximum rate, 1. The figures below, namely figure 9, 10, and 11, illustrates how the reliability or accuracy of the system is obtained:

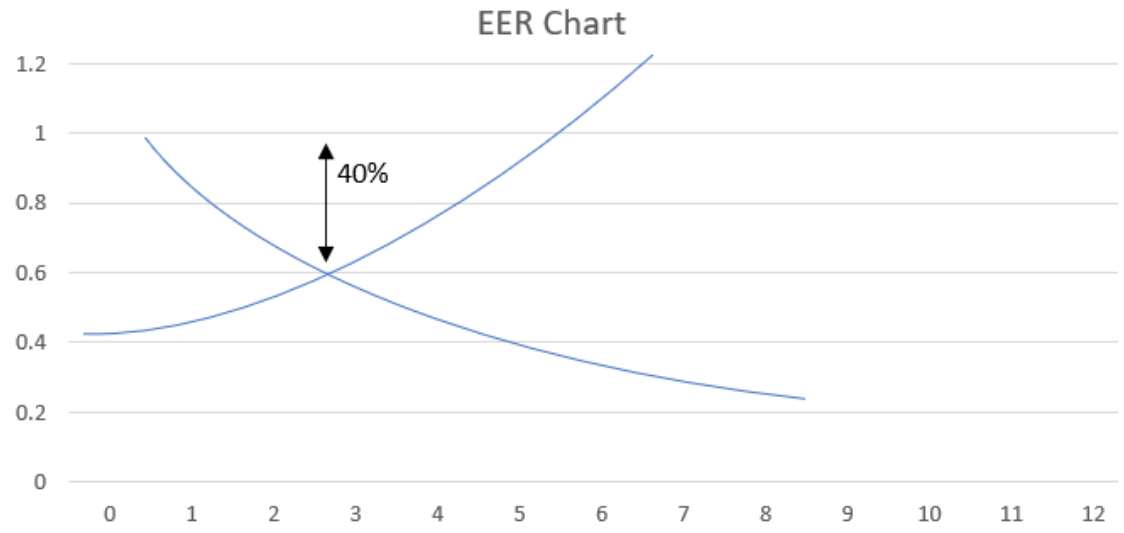

Figure 9: EER chart shows 40% accuracy

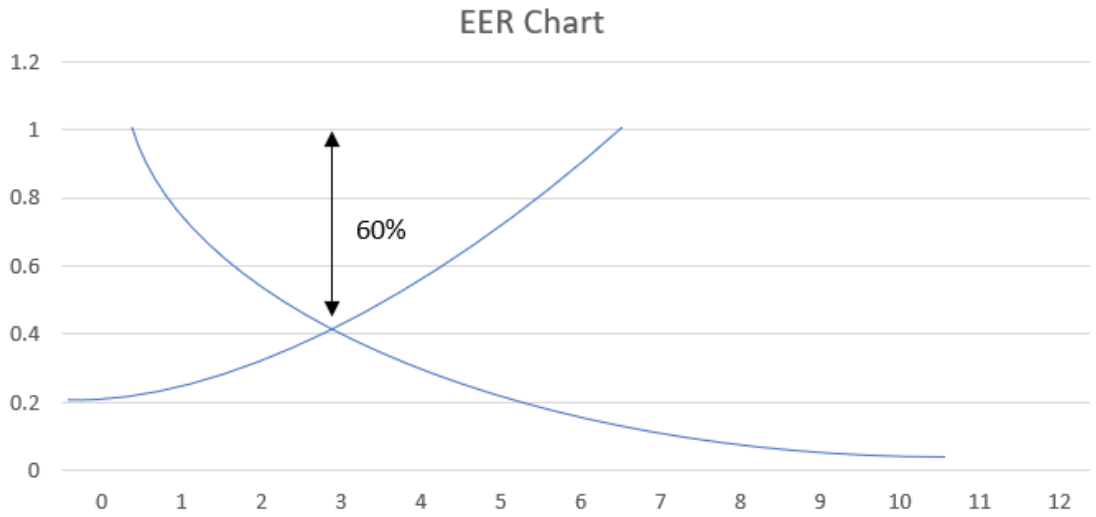

Figure 10: EER chart shows 60% accuracy

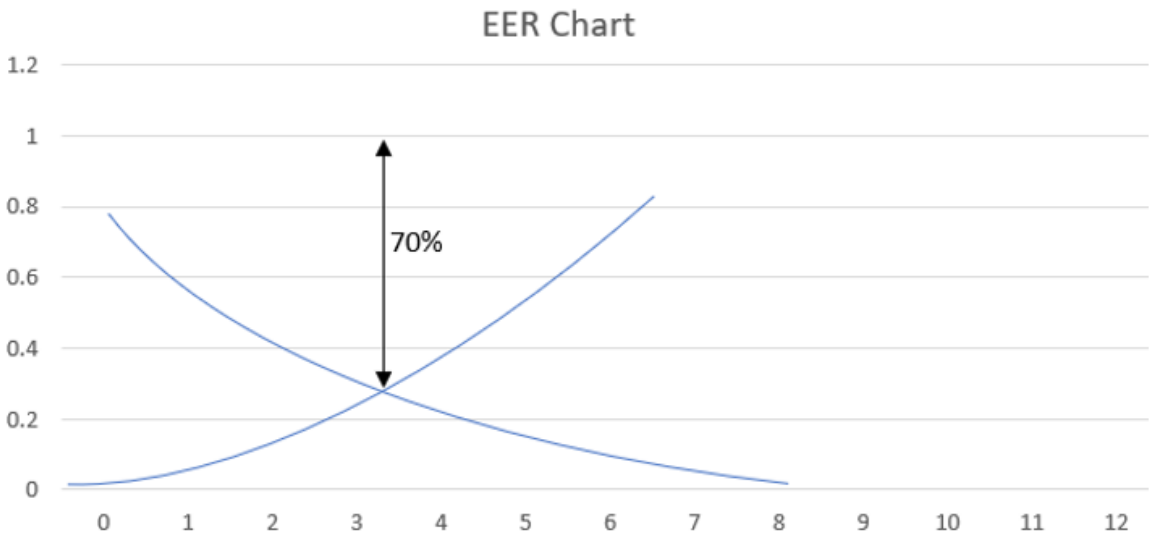

Figure 11: EER chart shows 70% accuracy

However, it is predicted that when the number of users increase, the system's performance will deteriorate, resulting in lower accuracy. Hence, the first chart will be testing with only 3 registered users, then the 2nd chart will consist of 5 registered users, lastly, the 3rd chart will consist of 10 registered users. The number of users is done in such a manner to help obtain a clearer difference in the system's performance, as a difference of 1 user in each chart will be harder to see as compared to a drastic difference in each chart. Then, a graph of performance against number of users will be charted to evaluate the performance of the system as the number of users increases, whereby performance variable data will be using the EER from each rate against security chart. In addition, a testing using 3 different pictures of the same authorized user will be performed to test if the system works in this manner, refer to exhibit 1, exhibit 2, and exhibit 3 in the appendix to see the different pictures used. Lastly, a basic system (In house validation) testing was carried out to examine the entire functionality of the prototype.

8 Results and Discussion

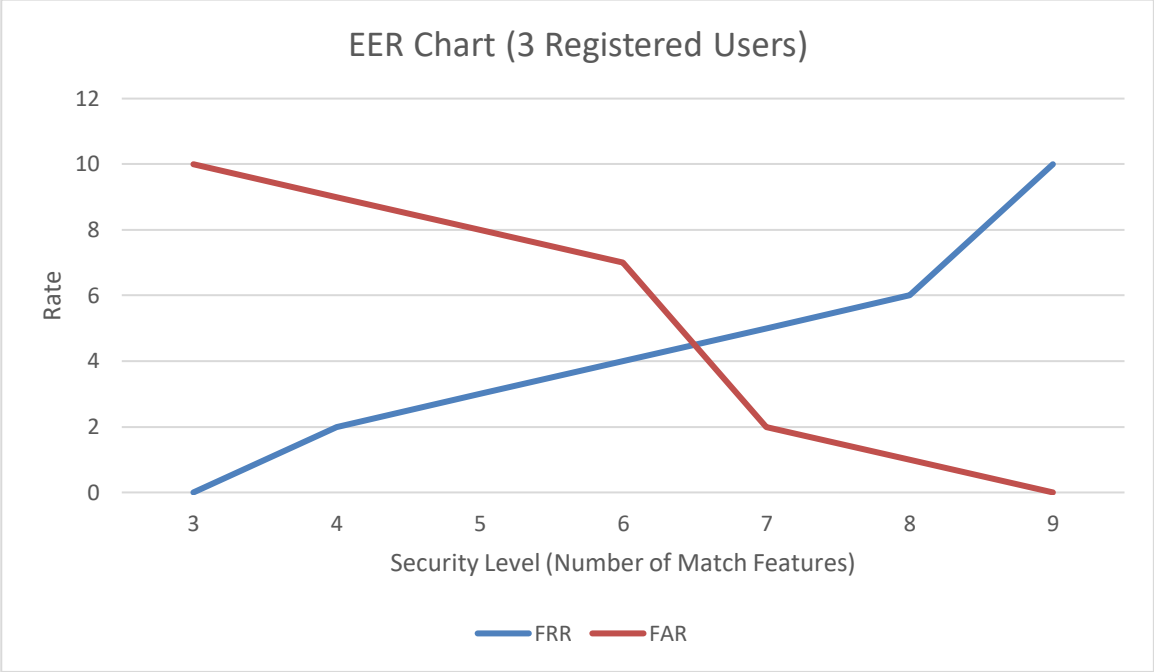

Figure 12: EER Chart with 3 registered users

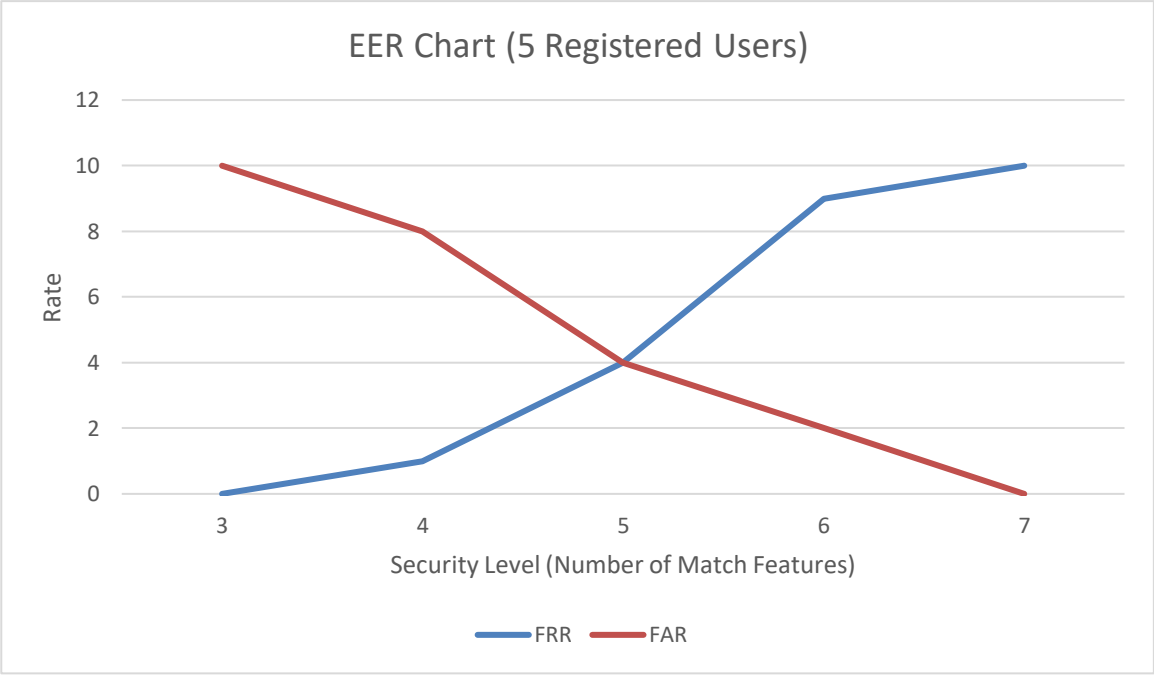

Figure 13: EER Chart with 5 registered users

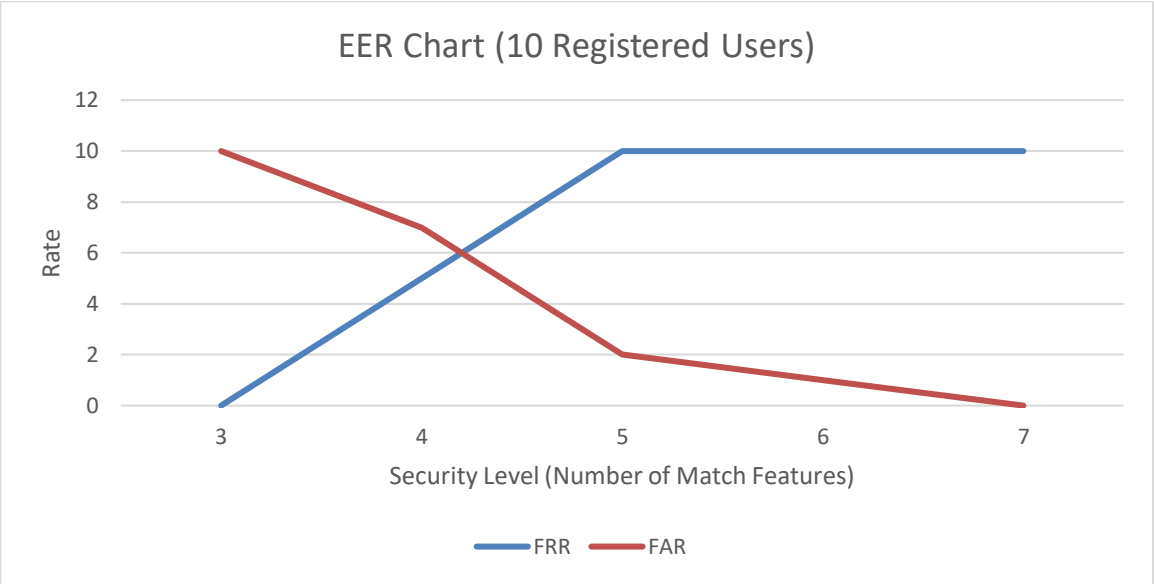

Figure 14: EER Chart with 10 registered users

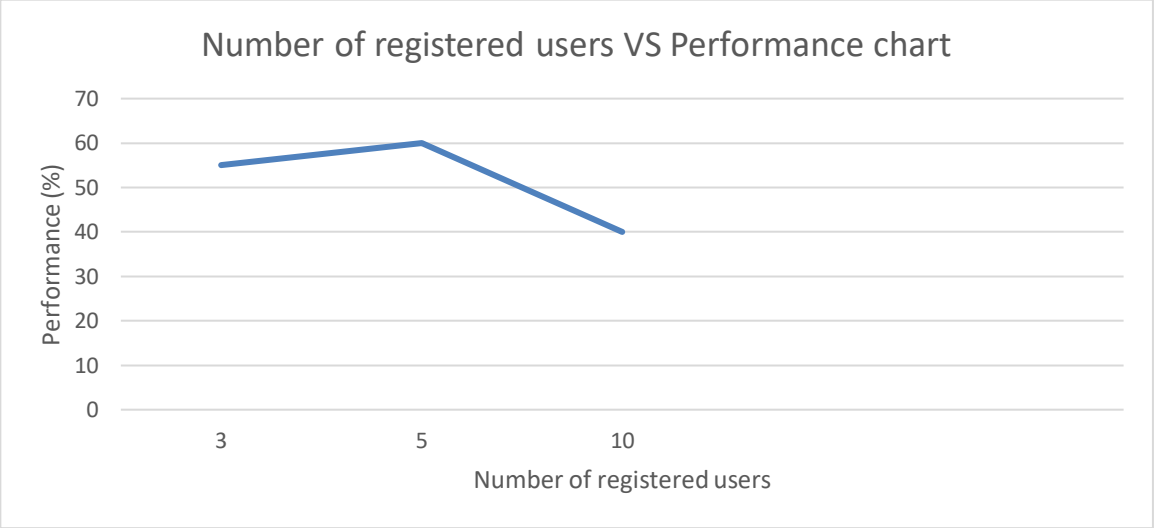

Figure 15: Number of registered users VS Performance Chart

	Exhibit 1 (Used for registration)	Exhibit 2	Exhibit 3
Number of successful attempts:	7	4	0
Number of failed attempts:	3	6	10

Table 3: Test case for same user but different picture

Table 4 below shows the system testing (In-house Validation) of the prototype:

No.	Test Scenario	Test Case	Pre-conditions	Steps used for testing	Data required	Anticipated Results	Post-conditions	Actual test results	Status of the test
1.	Enroll a new user into the system's database	• Page flow • Biometric template capture (facial) • Data insertion into system's database	• Wi-Fi is required to connect to the database • Permission for app to access camera needs to be granted.	1. Press "Enroll" button 2. Press "Detect" button and capture biometric template 3. Insert general information into the specified fields on the 4. Press "Register" button	• Biometric Template (facial) • All general information stated in the registration page.	Notifies user that registration is successful.	-	Notifies user that registration is successful.	Pass
2.	Authenticate an authorized user and update the user's profile	• Page flow • Data extraction from system's database • Biometric authentication (facial, and fingerprint) • Altering the right data in the system's database	• Wi-Fi is required to connect to the database • Permission for app to access camera needs to be granted.	1. Press "Authenticate" button 2. Press "Detect" button and perform biometric authentication 3. Press "Show information" button 4. Press "Update button" 5. Perform fingerprint authentication 6. Press "Show information" button 7. Change any one of the information 8. Press "Update" button.	• Biometric Template (facial, fingerprint) • All general information stated in the registration page.	Notifies user that update on client's profile is successful.	-	Notifies user that update on client's profile is successful.	Pass

Table 4: System's In-house Validation

According to the logic of how the system's recognition component is implemented, using a +- difference and slowly reducing the upper and lower bound to find the highest similarity template. It is still not enough to implement a very reliable system. This is because, in theory the system will have a higher FAR as the number of users increase in the database. Since, more users meaning there will be more diverse templates, and more chance for one of these templates to be matched wrongly. However, this will not be a problem given that the face detection algorithm provides a very consistent template for each capture., which in turn will provide a reliable and high accuracy system as the EER stays consistent as the number of users increases [1]. Unfortunately, that is not the case.

Based on the figure 12, the chart shows a performance of 55% accuracy at the security level of 6.5 with 3 registered users in the database. Whereas, figure 13 chart shows a performance of 60% accuracy at the security level of 5 with 5 registered users in the database. While, figure 14 chart shows a performance of 40% accuracy at the security level of 4 with 10 registered users in the database. Table 5 below provides a clearer view regarding the system's performance with different security level and different number of registered users:

Figure	Number of registered users	Security level (EER)	Performance (accuracy in percentage)
13	3	6.5	55
14	5	5	60
15	10	4	40

Table 5: Overview of System's performance

These data inevitably show the inconsistency of the system, whereby the system's performance is slowly increasing but with the compromise of having a lower security level. The inconsistency of the system as the number of users increase is much obvious in figure 15, where performance against number of users in the database is plotted.

The reason for this occurrence is due to MLkit's face detection algorithm's inconsistency, whereby this component relies heavily on the centre point of the capture, if it just so happens that both captures share a very high similar centre point, then it is very likely that the system will deem these 2 captures as the same template [20]. In an attempt to try and nullify the centre point's effect, the camera size shrank to a small box in the middle, to force the users to fit the face into the box for the capture's centre point to be more consistent. However, it is still insufficient to resolve the issue. Moreover, another huge factor is the distance of the camera to the object, whereby even if both captures share the same centre point, but their distance has a drastic difference, then the system will not deem these 2 templates as the same, since the facial features used to create the template is based on the distance between specific facial points, such as left-mouth and right-mouth [21]. Hence, when the distance is drastically different, the distance between these specific facial points also changes drastically, resulting in a very different facial feature for the template even if both templates share the same centre point.

One recommendation but also the hardest solution to resolve this issue it so implement a self-developed machine learning face detection algorithm, to specify specific facial points without the use of x/y-coordinates, which gives a rather inconsistent and unreliable capture. Besides that, another

rather difficult solution is to implement a re-scaling after the capture via a self-developed formula, to nullify the effects of centre point and distance of the camera and the object. On the other hand, another much easier but not so reliable solution is to implement the basic idea of average, whereby more capture creates a better reliable and capture to be stored in the system, meaning during the enrolment of the facial biometric, the client is required to perform at least 3 or more captures to provide an average and a more consistent capture, so the template stored in the database is more accurate. This average method is more generic, and it can be seen implemented in many different systems, such as signature recognition, and fingerprint biometric registration [22].

Another limitation of the system is the lack of obscurity, this is due to the method of trying to nullify the centre point's difference via a small camera box in the middle. It is because of this box that the user's phone will be rather close to the client's face to fill in the box and provide a more consistent capture. Although it does not mean much in terms of the system itself but for non-functional requirement, namely comfort, then sadly it is quite intrusive and does not really fulfil that non-function requirement [23]. Moreover, another limitation of this box method is the inability to detect a face during the facial biometric enrolment. Since, the camera shrank to a small box, sometimes MLkit's face detection algorithm is unable to detect the face.

In addition, as shown in figure 17 (*Test case for same user but different picture).* It appears that although exhibit 1 is the picture that was used for the biometric template during the registration process. The prototype can still detect and grant access at a decent rate, specifically 4 successful attempts out of 10 for exhibit 2. Whereas exhibit 1, the original picture that was used for registration has 7 successful attempts out of 10. One explanation behind this is the fact that although exhibit 1 and exhibit 2 both pictures are different in terms of angle, and hair, but due to the facial features similarity, the system can still detect them as the same person [24]. For example, referring to exhibit 1 and exhibit 2, both pictures have the person smiling with their teeth, so the cheeks facial feature will be similar, and both pictures have all the facial features required shown in a clear manner, so all the facial features can be extracted to create the biometric template required. However, when it comes to exhibit 3, the results varied greatly, whereby 0 out of 10 attempts were authenticated successfully. This is because, exhibit 3 is a picture that is more different is more different in terms of not smiling with their teeth, or one of the ears is not shown in the picture [24]. This makes the biometric template to be drastically different, so the system is unable to detect exhibit 3 as the same person as exhibit 1.

On the other hand, based on table 4, specifying unique test cases to examine the system functionality, the system's database implementation, page flow, data insertion, data extraction, data alteration, and biometric authentication (facial and fingerprint) all these features seem to be working without any error, but not very consistent on the facial authentication. Hence, the only problem that needs to be fixed for this system to be deployable is the face detection' algorithm or making each capture more consistent regardless of distance and centre point.

9 Conclusion and Future Work

In conclusion, the system is functional and working with an accuracy of 55% at the security level of 6 as it is the middle point between the 3 charts plotted that shows a rather acceptable security level. However, the prototype is not considered deployable due to its inconsistency in accuracy and reliability. Hence, this project serves as a stepping stone to solve the real-life problem, and future research is required. Since the system needs refinement for the face detection component, either by developing a new machine learning face detection algorithm to capture facial points consistently, or the use of a rescaling method. The system can also be further refined for a more consistent capture with the implementation of an average, whereby the system will request at least 3 or more captures during the biometric enrolment phase. These are the suggested areas that can be explored to make prototype more reliability with a higher accuracy that is deployable and capable of solving a real-life problem consistently.

References

[1] L. N. Clarke, S. Furnell and P. Reynolds, "Biometric Authentication for Mobile Devices," ResearchGate, 1 January 2002. [Online]. Available: https://www.researchgate.net/publication/253717880_Biometric_Authentication_for_Mobile_Devices. [Accessed 2 May 2020].

[2] J. Ashbourn, "Springer," 2000. [Online]. Available: https://www.springer.com/gp/book/9781852332433#aboutBook. [Accessed 5 May 2020].

[3] J. D. Cook, "Biometric security and hypothesis testing," John D. Cook Consulting, 31 October 2018. [Online]. Available: https://www.johndcook.com/blog/2018/10/31/biometric-security-error/. [Accessed 2 May 2020].

[4] V. Matyas and R. Zdenek, "Security of Biometric Authentication Systems," ResearchGate, 1 January 2010. [Online]. Available: https://www.researchgate.net/publication/251971395_Security_of_Biometric_Authentication_Systems. [Accessed 2 May 2020].

[5] P. Kallo, I. Kiss and A. Podmaniczky, "Official Gazette of the United States Patent and ..., Volume 1242, Issue 3," United States. Patent and Trademark Office, 1 January 2001. [Online]. Available: https://books.google.com.my/books?id=NfzQAAAAMAAJ&pg=PA2768&lpg=PA2768&dq=%E2%80%98Detector+for+recognizing+the+living+character+of+a+finger+in+a+fingerprint+recognizing+apparatus%E2%80%99,+US+patent+6175641,+January+2001.&source=bl&ots=OuKnTHjPFE&sig=ACf. [Accessed 2 May 2020].

[6] T. Matsumoto, H. Matsumoto, K. Yamada and S. Hoshino, "Impact of Artificial "Gummy" Fingers on Fingerprint Systems," 24 January 2002. [Online]. Available: https://cryptome.org/gummy.htm. [Accessed 2 May 2020].

[7] PYMNTS, "Deep Dive: The Fight To Stop Biometric Spoofing," PYMNTS.com, 5 December 2019. [Online]. Available: https://www.pymnts.com/authentication/2019/deep-dive-the-fight-to-stop-biometric-spoofing/. [Accessed 2 May 2020].

[8] THALES, "Biometrics: authentication & identification (definition, trends, use cases, laws and latest news) - 2020 review," 15 May 2020. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/inspired/biometrics. [Accessed June 19 2020].

[9] Y. Faridah, Haidawati Nasir, A.K. Kushsairy, Sairul I. Safie, S. Khan and T. S. Gunawan, "Fingerprint Biometric Systems," ResearchGate, 15 September 2016. [Online]. Available: https://www.researchgate.net/publication/309961589_Fingerprint_Biometric_Systems. [Accessed 19 June 2020].

[10] K. R. Jaiswal and G. Saxena, "Biometric Recognition System (Algorithm)," ResearchGate, 1 June 2017. [Online]. Available:

https://www.researchgate.net/publication/310022120_Biometric_Recognition_System_Algorithm. [Accessed 19 June 2020].

[11] F. M. Mustafa, S. F. Kak and P. Valente, "A Review of Person Recognition Based on Face Model," ResearchGate, 1 January 2018. [Online]. Available: https://www.researchgate.net/publication/323611486_A_Review_of_Person_Recognition_Based_on_Face_Model. [Accessed 6 June 2020].

[12] L. Wiskott, J. M. Fellous, N. Kruger and C. D. V. Malsburg, "Face recognition by elastic bunch graph matching," Arizona Commerce Authority, 1 December 1997. [Online]. Available: https://arizona.pure.elsevier.com/en/publications/face-recognition-by-elastic-bunch-graph-matching. [Accessed 19 June 2020].

[13] U. S. Kurmi, D. Agrawal and R. Baghel, "Study of Different Face Recognition Algorithms and Challenges," ResearchGate, 1 February 2014. [Online]. Available: https://www.researchgate.net/publication/290150010_Study_of_Different_Face_Recognition_Algorithms_and_Challenges. [Accessed 19 June 2020].

[14] N. H. Barnouti, s. S. Mahmood Al-Dabbagh and W. E. Matti, "Face Recognition: A Literature Review," International Journal of Applied Information Systems, 1 September 2016. [Online]. Available: https://www.ijais.org/archives/volume11/number4/935-2016451597. [Accessed 19 June 2020].

[15] F. Ahmad, A. Najam and Z. Ahmed, "Image-based Face Detection and Recognition," [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1302/1302.6379.pdf. [Accessed 19 June 2020].

[16] J. BrownLee, "How Much Training Data is Required for Machine Learning?," Machine Learning Process, 23 May 2019. [Online]. Available: https://machinelearningmastery.com/much-training-data-required-machine-learning/. [Accessed 19 June 202].

[17] M. Hassan, "How Biometrics on Smartphones is Changing our Lives," M2SYS, 1 January 2019. [Online]. Available: https://www.m2sys.com/blog/biometric-resources/biometrics-on-smartphones/. [Accessed 6 June 2020].

[18] O. Oztekin, "Integrate fingerprint authentication into your Android apps," Medium, 12 June 12017. [Online]. Available: https://medium.com/commencis/integrate-fingerprint-authentication-into-your-android-apps-dcb977b2e846. [Accessed 6 June 2020].

[19] Abhi, "Get fingerprint templates from fingerprint scanner [closed]," Stackoverflow, 30 June 2017. [Online]. Available: https://stackoverflow.com/questions/44841963/get-fingerprint-templates-from-fingerprint-scanner. [Accessed 6 June 2020].

[20] A. Studio, "ML Kit for Firebase," Google Developers, 2 June 2020. [Online]. Available: https://firebase.google.com/docs/ml-kit. [Accessed 6 June 2020].

[21] J. Birch, "Exploring Firebase MLKit on Android: Face Detection (Part Two)," Google Developers Experts, 31 May 2018. [Online]. Available: https://medium.com/google-developer-experts/exploring-firebase-mlkit-on-android-face-detection-part-two-de7e307c52e0. [Accessed 6 June 2020].

Appendix

Exhibit 1

Exhibit 2

Exhibit 3