ASHESI UNIVERSITY

LATENT FINGERPRINT IDENTIFICATION SYSTEM FOR CRIME SCENE INVESTIGATION

CAPSTONE PROJECT

B.Sc. Computer Engineering

Otitodirichukwu N. Effiong-Akpan

2021

ASHESI UNIVERSITY

LATENT FINGERPRINT IDENTIFICATION SYSTEM FOR CRIME SCENE INVESTIGATION

CAPSTONE PROJECT

Capstone Project submitted to the Department of Engineering, Ashesi University in partial fulfilment of the requirements for the award of Bachelor of Science degree in Computer Engineering.

Otitodirichukwu Effiong-Akpan

2021

DECLARATION

I hereby declare that this capstone is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

………………………………………………………………………………………………

Candidate's Name:

………………………………………………………………………………………………

Date:

………………………………………………………………………………………………

I hereby declare that preparation and presentation of this capstone were supervised in accordance with the guidelines on supervision of capstone laid down by Ashesi University.

Supervisor's Signature:

………………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………………

Date:

………………………………………………………………………………………………

Acknowledgements

Firstly, I would like to thank God for inspiring the idea for the project and enabling me to implement the idea. I want to thank my family and my friends for their financial, emotional and intellectual support.

I would like to thank my supervisor, Mr Francis Gatsi, for his direction and encouragement throughout executing this project. Finally, I would like to thank other lecturers in the Engineering Department for their feedback that influenced the project’s decisions.

Abstract

The traditional means of criminal investigation used in Nigeria is often unreliable and leads to innocent people's wrongful detention and a lack of justice for deserving offenders. The poor record-keeping and weaknesses in Nigeria's investigation process significantly contribute to the high levels of crime and insecurity in Nigeria.

To tackle these issues, this project provides an implementation of a fingerprint identification system to improve criminal investigation in Nigeria. Three image processing algorithms and a Convolutional Neural Network classification algorithm were explored for matching performance. The Convolutional Neural Network classification model performed better than the three image processing algorithms with an accuracy of 64.44%. The final system provides a web interface with database interaction to send a fingerprint image and meta data to receive match results and potential suspect (criminal) information.

Table of Content

Chapter 1: Introduction

1.1 Background

The Attorney-General of the Federation affirmed in 2006, 17.1% of cases of Nigerian prisoners were delayed for trial due to incomplete investigation. 3.7% were interminably imprisoned because of lost investigation case files. In 2010, the Minister of Interior stated that of the 46,000 prison inmates in Nigeria, 65.2% of them were awaiting court hearings for the same number of reasons as investigated in 2006 [1]. Despite the high numbers of prison inmates, many actual offenders escape punishment due to incomplete and incompetent investigations strongly linked to insufficient evidence for conviction by the court. With the inadequate investigation in Nigeria, the justice system is incapacitated, and as a result, crime continues to be a significant problem for Nigeria [1].

1.2 Problem Definition

The lack of organised criminal records, adequate forensic investigation and technology poses a threat to a nation's security. Arresting criminals without adequately documenting them and their offence robs the criminal justice system of critical information to judge cases. Additionally, it allows ex-convicts (former criminals) whose crime sentences have ended- to return to daily life activities with no trace of past criminal activity and a potential to re-offend. On the other hand, the lack of forensic technology to process biometric information makes the investigation process tedious and inaccurate. The dependence on eye-witness and forced interrogation can lead to false convictions and causes innocent people to get punished for crimes they do not commit. These criminal investigation weaknesses create insecurity, reduce the citizens' confidence in the police, and hinder the Criminal Investigation Department's technological advancement.

An interview with the Commissioner of Police, Mr Abiodun Alabi, revealed that currently, there are no electronic criminal records that efficiently track criminal activities in Nigeria. All existing criminal records are documented on paper and in physical files. These files can easily be lost or ruined by mishandling, fires, carelessness and inadequate storage. Also, there is very little use of biometrics, i.e. (DNA, fingerprint, face recognition and iris recognition) during an investigation. The police collect blood samples from crime scenes, and occasionally, places equipped with CCTV cameras capture people's faces. Recently, DNA labs have been built to process blood samples. However, there is no official face recognition technology to process images from CCTV cameras. Despite fingerprints being one of the most common forms of evidence used in a criminal investigation, the police do not collect them. Hence, there is no technology to process them. However, the Criminal Investigation Department in Lagos is taking more interest in forensic investigation. On Sept 28 2020, the Nigerian Air Force began a month-long Advanced Crime Scene Investigation Course for the Air Provost personnel. The training aims to equip the personnel with forensic investigation skills to assist forensic technology employment for crime-fighting in Nigeria [2]. The new interest in forensic technology and traditional file storage problems inspire a need to design an identification system for crime scene investigation based on fingerprint biometrics.

This project aims to design an on-site fingerprint identification system to process fingerprint information from crime scenes accurately. It aims to bridge the division between fingerprint evidence capturing, fingerprint evidence storage, fingerprint evidence processing, and database keeping. The system presents a novel approach to crime scene investigation that will increase criminal investigation reliability in Lagos, Nigeria. The objectives to achieve the project aim are outlined in the next section.

1.3 Project Objectives

- Develop an on-site fingerprint identification system
- Develop a front-end web application
- Develop a prototype database system for the criminal department

The remaining parts of this chapter summarises some concepts and processes associated with forensic science and fingerprint biometrics necessary to understand and execute this project.

1.4.1 Forensics science

According to the National Institute of Justice [3], "Forensic science is applying sciences such as physics, chemistry, biology, computer science and engineering to matters of law." Forensic science dates to the 16th century till date and has been used to solve crimes and convict suspects of crimes [4]. The simple approach forensic scientists use to investigate a crime is: examine for physical evidence, administer tests, interpret the data, make clear reports and make a truthful testimony to add to the scientific integrity of the investigation [4]. Various technologies are used in forensics, such as DNA fingerprinting, facial reconstruction, alternative light photography, and digital forensics. Each of these technologies has various modes of operations and applications that can be used in the investigation process. For instance, alternate light photography aids in the collection of biological samples by taking advantage of a sample's light absorption properties or fluorescence effect [5].

1.4.2 Biometrics technology

Biometrics technology measure a person's physical characteristics, e.g. (iris, palm print, footprint, fingerprint etc.) for identification verification. Biometrics must be unique, collectable, and permanent to be practical or useful [6]. Biometrics technology is applied in our everyday lives, such as mobile phone security locks, National Identification Registration, Election registration, and many other processes requiring authentication and verification. Some standard biometrics techniques are fingerprint identification, iris recognition, facial recognition, finger and vein pattern recognition. Each technique has both advantages and disadvantages, and neither guarantees a full-proof identification as they are all prone to spoofing. Existing fingerprint recognition devices are relatively affordable and easy to use but offer varying quality in their false acceptance and false rejection rates [7]. As defined in [8], false rejection is a situation where a biometric system rejects attempted access by an authorised user. This error is referred to as a type I error.

On the other hand, false acceptance is when a biometric system will accept a wrong access attempt by an unauthorised user. This error is typically referred to as a type II error. However, fingerprint images can be manipulated and replicated, which compromises the integrity of this technology. Additionally, people have skin conditions that make it challenging to obtain fingerprints, and some people do not even have fingerprints at all [9]. Despite these concerns, fingerprint recognition is the most widely used biometric technology.

1.4.3 Fingerprint identification

Fingerprints are friction ridges (represented on images as lines) on the surface of a person's finger that is unique to the person. Fingerprints as a biometrics technique are captured as images with dark lines and special distinguishing features (arches, whorls, ridges, etc.) [9]. Thermal, optical, ultrasound and capacitive sensors can be used to collect a fingerprint image. Fingerprint scanners that use optical sensors are the oldest sensor devices and can easily be integrated into existing systems such as computers, phones, and wireless hardware. The fingerprint identification process can vary depending on the matching techniques. The existing matching techniques can be classified into three types:

Correlation-based matching, Minutiae-based matching, and Non-minutiae feature-based matching [10]. The most popular technique is the minutiae feature-based matching. The fingerprint identification system using minutiae-based matching typically has four primary stages: The first stage is the Acquisition stage that involves capturing the fingerprint image. The second stage is the Pre-processing stage that involves enhancing the captured images, Binarisation and Thinning of the image. The third stage is the Feature extraction stage that involves extracting key features such as singular points, minutiae, ridges and valleys from a thinned image. The fourth stage is the Identification and Verification stage which involves matching minutiae points from two fingerprint images using a matching algorithm that measures distance and similarity [11]. A fingerprint identification's sensitivity and accuracy usually depend on the fingerprint images' quality, the pre-processing techniques, and matching algorithms.

1.4.4 Fingerprint Detection

In a crime scene context, fingerprint images are found on any solid surface, including a victim's body [12]. Fingerprints can be classified into 3D plastic prints (fingerprints found on soft surfaces such as soap, wax, wet paint, etc.). The second type is Patent prints (visible fingerprints found on hard surfaces), and the third is Latent prints (fingerprints found on hard surfaces that are not visible). Latent prints are formed when a person's finger deposits its natural oils and sweat on to a surface. These invisible or latent prints must be made visible by an investigator to be detected, and this can be done by using alternate light sources, fingerprint powders or chemical reagents. For latent prints, the smoother and less porous a surface is, the higher the chances of detecting and developing the fingerprint [12].

1.4.5 Fingerprint Capture and Collection

Scanned fingerprint images are classified into two kinds, rolled and plain fingerprints. Rolled fingerprint images are captured by rolling fingerprints from one side to another on a sensor. Plain fingerprints are obtained by pressing the fingers against the fingerprint scanner [13]. Patent fingerprints are photographed in high resolution using a forensic measurement scale in the image [12], and typically no alternate light sources or dyes are needed to improve patent print images. Latent print images are made visible using both physical and chemical means. A standard method is dusting fingerprint powder (e.g., aluminium flake) over surfaces where fingerprints can be detected. Once the fingerprint appears, the image is taken with a high-quality camera, lifted using a lifting tape and placed on a latent lift card for preservation. However, fingerprint powders can potentially contaminate the evidence and prevent any further processing on the lifted image; hence, alternate light sources are preferably used by some investigators [12]. Latent print images can also be made visible using chemical developers that react with the latent print residue, e.g., inorganic salts and amino acid. An example of a chemical developer is Ninhydrin, a chemical substance that turns fingerprints purple for more comfortable photographing [12]. Due to the nature of crime scenes and the shortcomings of the various methods of revealing a latent print, most latent print images lifted from or captured at crime scenes are noisy, smudgy, incomplete or overlapped with one another, posing challenges for automated identification.

1.4.6 Fingerprint Image Pre-processing

The fingerprint images must be pre-processed and enhanced in the fingerprint identification process to aid feature extraction and matching in the following stages. Pre-processing usually includes basic processes such as noise removal, binarisation, thinning and filtering [14]. Other methods include histogram equalisation and segmentation. Noise in an image is the unwanted information represented as grains depicting a random variation of intensity values [16]. There are four common types of noise potentially present in fingerprint image: Gaussian noise, impulse noise, Poisson noise and speckle noise. Different special filters are better suited to eliminate each kind of noise [17].

Binarisation converts a fingerprint image to binary from its gray-scale, while thinning converts it to a one-pixel finger image [18]. Histogram equalisation is a practical and straightforward image contrast enhancement technique [19]. It enhances the illumination of latent fingerprint images, improving the image's clarity and quality [18]. Segmentation is the process of portioning a digital image into smaller sets of pixels to change the image's representation for more straightforward analysis [20]. Segmentation of latent fingerprint images is separating the useful finger area from the background [18]. Latent fingerprints usually require more pre-processing techniques than the processes above.

1.4.7 Fingerprint Image Feature Extraction and Matching

The fingerprint consists of protruding lines called ridges and concave lines known as valleys that form unique patterns and distinguishing features. There are two main classifications of fingerprint features, the Global Feature and the local feature. The Global feature, commonly known as the Henry feature, describes the fingerprint grain structures globally. The local feature, commonly known as Galton features, describes the prints' minutiae details, such as the end of ridges, junctions and bifurcation points [9]. Other classifications of fingerprint features are directional field feature and singular points. The directional field (DF) is a global feature that describes the ridge-valley structure's local orientation in a fingerprint image. Singular points are the cores and deltas that refer to the fingerprint's directional field's discontinuities [21]. The core is the topmost point of the deepest curving ridge, and the delta is the point where three ridges meet, as seen in fig 1.

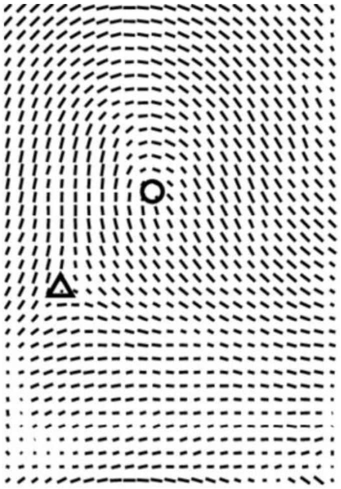


Figure 1 Directional field and Singular points. Core indicated by circle and delta indicated by cross [21]

The most common method of feature extraction is the minutiae-based feature extraction method. The minutiae refer to ridge endings and ridge bifurcations in a fingerprint image, as shown in fig 2. The ridge ending is where the ridge lines end suddenly, while the ridge bifurcation is where a ridge splits into two ridges [22].

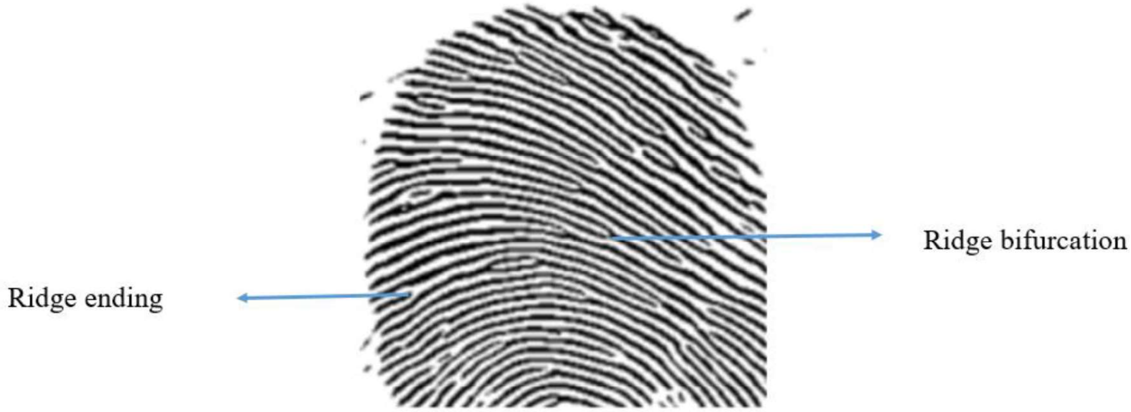


Figure 2. Ridge bifurcation and ridge ending

The feature matching stage is the final stage of identification, where an input fingerprint image is compared to a set of fingerprints in a database to find a match. Several matching techniques can be divided into three categories: Correlation, minutiae, and non-minutiae or pattern-based [10].

Correlation-based matching involves superimposing two fingerprint images and calculating the correlation between corresponding pixels at varying rotations and displacements [24]. Minutiae-based matching minutiae are extracted from two fingerprint images to find a maximum number of matching sets. In pattern matching, the fingerprint image’s global features, such as the frequency and region orientation of two fingerprints, are aligned to determine a match [24].

Chapter 2: Literature Review and Related Work

There are a lot of existing algorithms for the different stages of fingerprint identification [14], [25]- [28]. Currently, many of these algorithms perform better on scanned fingerprint images. There is ongoing research around the processing and matching of latent fingerprints captured at crime scenes. This chapter reviews 5 of the existing techniques and approaches for both scanned and latent fingerprint images.

2.1 Fingerprint Image Preprocessing

A challenge in fingerprint images is the presence of noise. Noise is introduced in an image through the sensor used to capture the image. Impulse noise is caused by sharp and sudden disturbances to the sensor used to capture the image. It appears as the maximum grey value (white point) and minimum grey value (black point) [25]. Traditional median filters use a fixed window size that causes a problem where if the window size is too small, it fails to eliminate all the impulse noise.

On the other hand, if the window size is too big, the image becomes blurry. Han, Wang and Chen propose a new method of removing impulse noise from an image known as adaptive median filtering [25]. The method involves initialising a window and determining if the pixel at the filter window's centre is impulse noise. Next, determine the window size based on the median, maximum and minimum value within the filter window. Finally, perform adaptive median filtering. Their algorithm was tested on two fingerprint images, rolled fingerprint image and a latent fingerprint image, both with impulse noise densities of 0.1, 0.2, 0.3,0.4 and 0.6. The PSNR of the images contaminated with impulse noise was compared against the PSNR of traditionally filtered images and the adaptive filtered images. The adaptive filtered images had higher PSNR values for both images at the different impulse noise densities. Their adaptive filtering method proved efficient for impulse noise applied by the researchers; however, the performance on images with impulse noise direct from the sensors was not evaluated.

Another challenge with latent fingerprint images is the lack of distinct features known as minutiae. Recent research seeks to find ways to reconstruct these missing features in a fingerprint image. Liban and Hilles suggest the combination of Edge Directional Total Variation (EDTV), image enhancement and lost minutiae reconstruction to pre-process fingerprint images [26]. They performed the algorithms on the NIST SD27 database on fingerprint images classified by “good”, “bad”, and “ugly”. The problems with similar technique, i.e. Gabor filtering, Gabor filtering using Short Time Fourier Transform (STFT) and the Adaptive Directional Total Variation (ADTV) method. The Gabor filters work better for high-quality images. The Gabor filter using the STFT method requires a manual mark-up of singular points and regions of interest for the enhancement.

On the other hand, Total variation methods automatically decompose latent images into their texture and cartoon components [26]. The texture is the oscillatory fingerprint ridge patterns, while the cartoons are the unwanted content[26]. The Gabor filters can also lead to false ridge artefacts because it uses fixed orientation. In regions of high curvature, the assumption of a single dominant ridge orientation is not valid. The Gabor filters also do not restore ridge structure destroyed by heavily structured noises. Finally, the Adaptive Directional Total Variation model does not easily estimate local parameters for low quality latent images. The proposed solution combines edge total variation and a multi-scale based sparse representation to remove noise and improve the ridge structure’s clarity. The methods used are normalisation (to reduce variations in grey levels along ridges and valleys of the image), adaptive denoise based on EDTV, reliable orientation image estimation, local frequency estimation, region masking, Gabor filtering. The Root Mean Square Error (RMSE) and PSNR of the proposed method were compared against other total variation models such as the pure total variation model and the directional total variation model for the good, bad and ugly images in the database. The method proved a 30% improvement in matching accuracy. A limitation of this method is that it does not account for overlapped fingerprint images.

Van, Vu and le propose another pre-processing technique that focuses on direct grey-scale minutiae extraction of plain and rolled fingerprint images [27]. The technique provides a means to remove noise and a form of minutiae reproduction. It combines Gabor filters which act as band-pass filters to remove noise and clarify the ridges in the fingerprint image, with an Adaptive Modified Finite Random Transform (AMFRAT) filter. The AMFRAT filter develops the MFRAT filter that adjusts window size according to coherence values. The filtering result produces the ridges as linearly symmetrical areas that are ideal for direct grey-scale minutiae extraction. This method sought to address the limitations of other existing approaches such as the Directional Fourier domain filtering, Gabor filters, Hong's algorithm using Anisotropic filters and Directional Median filter. Some of the limitations of these techniques that the method Van et al. addresses are:

- Directional Median Filter (DMF) and anisotropic filter could not efficiently join broken ridges without destroying essential singularities in the noisy fingerprint image [27].
- The methods are limited for direct grey-scale minutia extraction [27].

A closer competitor to the method proposed by Van et al. is Bigun's method that suggests creating a Laplacian-like image pyramid to detect minutiae by using symmetry filter correctly. However, this technique proved not as efficient in reproducing ridges of very dry fingerprint images. Hence, the paper proposes a hybrid approach using the Gabor filter and AMFRAT to provide a linear symmetry filter to locate minutiae to enhance images better and quicker than Bigun’s method. Using the following methods: normalisation, orientation field computing, orientation smoothing, frequency image estimation, Gabor filtering, AMFRAT filtering and direct grey-scale extraction, they performed experiments on the FVC2004 databases (set A). Comparisons were made on the Gabor filters, curved Gabor filters, Bigun’s method using a “good” index that evaluated mean, standard deviation, Equal Error Rate (EER) and average time. The proposed method outperformed the other methods overall; however, it needs to be adapted for latent fingerprint images.

2.2 Fingerprint Image Feature Extraction and Matching

Saad and Issawi propose a method for feature extraction and matching based on neural networks. Their method involves the extraction of minutiae features using a feed-forward multiple layer perception using three layers. The network consists of a hidden layer that uses nine neurons linked to the input vector and, a hidden layer containing five neurons and an output layer having one neuron. They used a code “1” for ridge bifurcation and a code “0” for ridge ending. The feature extraction and matching were done using MATLAB neural network toolbox on the FVC2002 database. To extract the features from the enhanced fingerprint images, they used the following process

- determined the region of interest (ROI) to determine the core point of the fingerprint image.
- Specified a 3 × 3 window to scan over the image, extract the features and determine if it is a minutia or not

To match, they determine the core point of the fingerprint image using the Poincare algorithm. Next, they calculated the Euclidean distance between two vectors of two fingerprints. The Euclidean distance of the fingerprint image is the ordinary distance between two sets of ridge bifurcations or ridge endings or a ridge bifurcation or ridge ending and the fingerprint's core point.

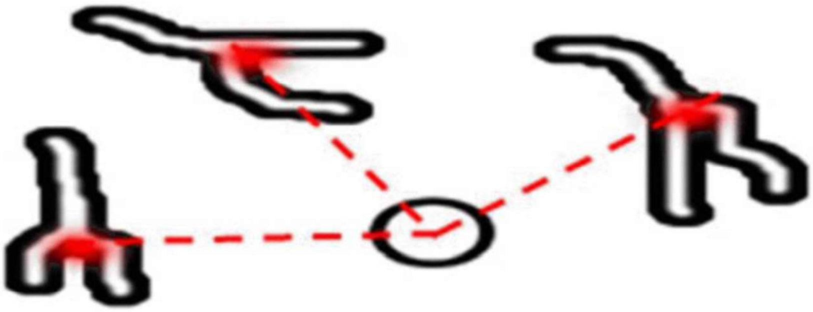


Figure 3. Euclidean distance between the bifurcation and core point of a fingerprint image [14].

The equation to calculate the Euclidean distance is given as

$$D = ||x - y|| = \sqrt[2]{\sum_{i=1}^{n}(x_i - y_i)^2}$$

To test their Feed-Forward Feature Neural Network (FFNN) feature extraction algorithm, they use a stratified 10-fold cross-validation scheme to train the FFNN. They divided their dataset into ten subsets, 9 for training and 1 for testing and repeated the process ten times. Their strategy extracted all ridge bifurcations and ridge endings; however, it is computationally expensive due to the several scanning turns. To evaluate the whole system, they performed experiments on five copies of actual fingerprints on ten persons. The total average matching percentage accuracy was 91.6 %. More informative features about the fingerprint other than the minutiae points could be extracted to improve their algorithm [14].

As previously highlighted in chapter 1, there are two common ways to extract features from a fingerprint image, image-based (global) and pattern-based (local). The most popular feature extraction method is checking patterns using minutiae points representing the location of distinguishing features in a fingerprint image. Typical minutiae-based pattern matching methods rely on the minutiae extraction point's accuracy and detection of the fingerprint's cores and delta for pre-alignment.

Sindhu and Arunadevi introduce a pattern-matching technique where features like the core are not fully defined or where the fingerprint only has some partial overlapping [28]. The method uses a hybrid shape and orientation descriptor that helps address detection problems. The hybrid filter filters false and unnatural minutia pairings while using the fingerprint image's ridge orientation to improve matching accuracy [28]. The technologists used 150 standard database images and 40 real-time images to compare with the database images to test their algorithms. They compared the accuracy, entropy, and the false acceptance rate of their matching algorithm to Zhe Jin's kernel method and Meng-Hui's genuine interval concealment for binary biometric representation. Their method outperformed the kernel method and the genuine interval concealment for binary biometric representation for scanned fingerprint images.

Chapter 3: System Requirements

This chapter highlights the design requirements and specifications for the proposed system. These requirements were gathered through observation and interviews with police officers at Force Criminal Investigation and Intelligence Department, Nigeria and secondary research on Automated Fingerprint Identification Systems (AFIS). The user requirements, system requirements and non-functional requirements of the system are highlighted below.

3.1.1 User Requirements

The user requirements are the user expectations for the operation and user interaction of the on-site fingerprint identification system. This project's users are the police who investigate crime scenes (crime scene investigator), fingerprint analysts, and the police who arrest criminals. The user requires the system to

1. Be portable
2. Have a low cost of implementation
3. Have a low cost of maintenance
4. Present a well-organised collection of data
5. Be secure
6. Be easy to use

3.1.2 System Requirements

The system must:

1. Have a sensor device to capture images from crime scenes and a sensor device to scan and enrol fingerprint at a police office. The devices should have high effective resolutions and pixel densities to obtain good quality images.

2. Have a web application with front end applications and a server-side application to send captured data, interact with the database, view results of the fingerprint processing.
3. Have a web application with a front-end application to secure registration of the criminal department officials into the system, register criminals, and view criminal information.
4. Have geolocation automatically tag the location of a crime scene at the point of sending fingerprint image and other metadata.
5. Have a server-side application with accurate and quick algorithms appropriate for real-time identification
6. Produce accurate results for fairly low-quality images with slightly noisy backgrounds
7. Facilitate communication over a long-range
8. Ensure the protection of folders where the fingerprint images are stored
9. Deny access to the system to non-criminal department officials
10. Provide user-specific features and functionalities on the web application

3.1.3 Non-functional Requirements

1. The system should be accurate
2. The system should be consistent and reliable
3. The system should be secure with administrative user login credentials.

Chapter 4: Design & Implementation

Considering the requirements specified in chapter 3, the first part of this chapter discusses the proposed system's design. The system is divided into two components, a hardware component and a software component serving as the core of this system. The details of the blocks in the hardware and software component will be discussed later in this chapter.

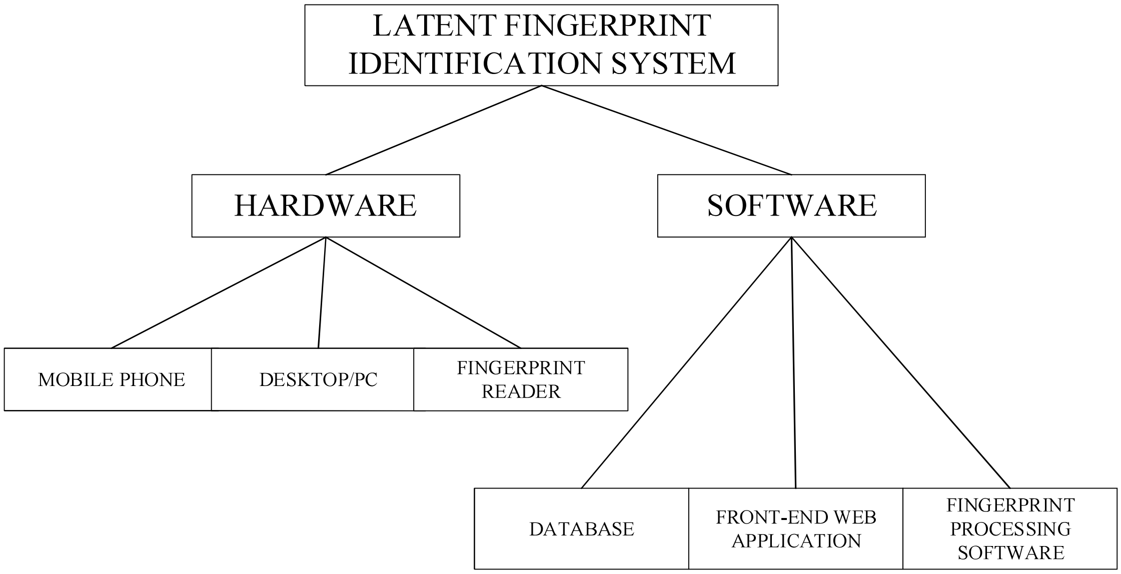

LATENT FINGERPRINT IDENTIFICATION SYSTEM
HARDWARE
SOFTWARE
MOBILE PHONE
DESKTOP/PC
FINGERPRINT READER
DATABASE
FRONT-END WEB APPLICATION
FINGERPRINT PROCESSING SOFTWARE

Figure 4. On-site Latent Fingerprint Identification System Design Architecture

4.1 Hardware Component

The project's hardware component consists of a mobile phone, a desktop or PC, and a fingerprint reader. The mobile phone consisting of a camera serves as a capturing device to capture fingerprint images from a crime scene. It will temporarily store the images and send the images over the internet to a remote server on a desktop or PC. The desktop/PC will be situated in a police office where criminals' registration and criminal intelligence are performed. The fingerprint reader will be used to enrol criminal fingerprint into the system.

Mobile phone

The rear camera in iPhone 11 has an effective resolution of 12 Megapixels (MP) which is 2 MPs above the minimum resolution (10 MP) for capturing a fingerprint image, making it suitable to take a good fingerprint image. The iPhone 11 weighs 194 grams, about a quarter the weight of professional cameras used in crime scene investigation.

Desktop/pc

An HP Pavilion x360 convertible laptop is used as the testing environment and the situation of the server. It has an i3-7100U CPU with a 2.40 GHz clock speed and 4GB RAM.

Fingerprint Reader

The *Digital Persona U Are U* fingerprint reader is an optical fingerprint reader used to scan individual fingerprint images. It has a USB 2.0 cable suitable for a connection to a desktop/PC. The scanner has a resolution of 512 dpi, which satisfies the recommended scanner resolution by the FBI.

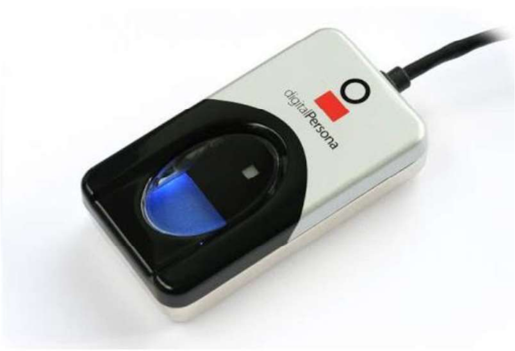


Figure 5. Digital Persona U Are U Fingerprint Reader

4.2 Software Component

The first software component consists of a database that stores the collection of fingerprint images, evidence information, user login information, criminal registration information and criminal activity information. To build the database for the system, certain factors were considered:

- Structure: the way the data will be stored and accessed
- Size: the amount of data being stored
- Type: the type of data being stored
- Speed and scale: the time taken to read from and write to the database.

The data collected from the crime scene consists of the fingerprint image and its metadata. The metadata is information describing the fingerprint image as evidence, the investigator and the location. These are pre-defined fields set by the head of the criminal investigation department. All other data being stored such as user information and processing results, consists of text and images. Typically, government information has pre-defined categories, and over time, only minimal modifications are made to these categories. The information collected is extensive as it consists of information about each individual in a country's population. Similarly, crime data have a structured nature, with a clear relationship between the users (police officers, criminals) and criminal activities. Considering this, the prototype database was built using a Structured Query Language (SQL), SQLAlchemy.

The second component is the front-end application for the users, the crime scene investigators, fingerprint analysts and police officers. This component serves as an interface for the users to interact with the database and processing software. A web application interface was created using Python Flask for the crime scene investigator to upload the fingerprint images and information about the image evidence to a remote server. The application consists of 3 pages: a page for logins and authentication, a page for registering users onto the platform and a form page to tag the evidence, i.e., fill image evidence information and upload the image to a folder. On the other end, there are two pages to register criminal information, view match results. The crime scene investigator will send fingerprint information over the internet using HTTP web protocols. The criminal investigator and other police officers would be able to view the matching results.

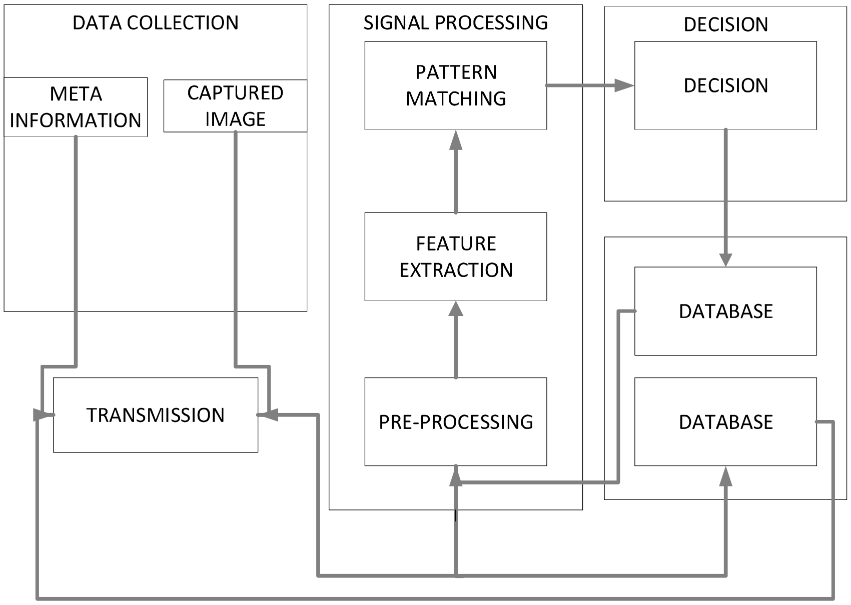


Figure 6. Block Diagram for Proposed System

The block diagram for the proposed latent fingerprint identification system can be seen in fig. 6. The diagram highlights the different stages of the crime scene fingerprint identification process. The details of the stages are discussed below.

4.3 Data Collection

Data collection is the first stage where the crime scene investigator takes a photo of the fingerprint impression made on a crime scene's surface. The proposed system's assumed method is the use of a fingerprint brush and an aluminium-based fingerprint powder. The investigator uses a phone camera with an acceptable resolution to capture the images. The images are temporarily stored in the phone's memory. The investigator logs in to the application and fills in information about the fingerprint evidence specifying the crime, selecting the location and describing the item. The fingerprint images are uploaded through the application to the server to be processed.

To enroll criminals on the other end of the system, the assumed method was an optical scanner collecting individual prints from each criminal's ten fingers. This method though time-consuming simplifies the comparison against the single fingerprint images in a crime scene.

4.4 Transmission

The fingerprint images and the evidence information will be sent through the Web application using Wi-Fi 802.11ax or 4G LTE cellular network and HTTP protocol. The image and form information is sent as HTTP post requests to the Python Flask server.

The application can ensure connectivity over a long range of the cellular network cell tower architecture.

4.5 Preprocessing

The pre-processing techniques used for the latent fingerprint images were: conversion to grey-scale, normalisation, ridge segmentation, ridge orientation estimation, ridge frequency estimation, ridge filtering, region mask generation, binarisation and thinning [11] [29] [31].

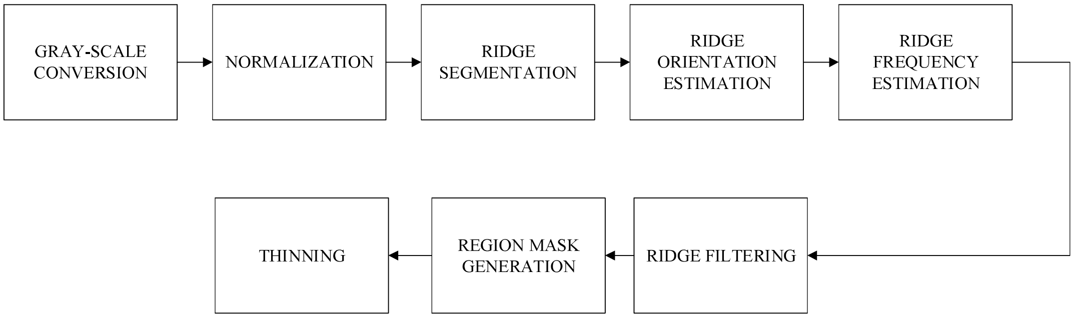


Figure 7. Pre-processing steps

4.5.1 Gray-Scale conversion

Initial fingerprint images contain three channels of Red, Green and Blue in the RGB colour scheme. The fingerprint image is converted to Gray-scale, which contains only one image channel, essentially a 2D matrix. Let the grey-level image, I be defined as an $N \times N$ matrix where $I(i, j)$ is the intensity value of the pixel at the ith and jth column [29].

4.5.2 Normalisation

This step involves changing the pixel intensity values of a fingerprint image to have a pre-specified mean and variance. It helps maintain a normal range or data distribution and reduce the variations across each pixel value of the fingerprint image. Normalisation is also known as histogram or contrast stretching.

For a grey-valued pixel, $I(i,j)$ with M and VAR denoting an estimated mean and variance of the value I, the equation [29]

$$G(i,j) = \begin{cases} M_0 + \sqrt{\dfrac{VAR_0(I(i,j) - M)^2}{VAR}}, & if\ I(i,j) > M \\ M_0 - \sqrt{\dfrac{VAR_0(I(i,j) - M)^2}{VAR}}, & otherwise \end{cases}$$

4.5.3 Ridge Segmentation

Segmentation is dividing an image into many parts called segments to simplify the image's representation for better analysis. A necessary step in segmentation is locating a Region of Interest (ROI). In image processing applications it involves separating the foreground from a background or selecting specific components from the foreground from the rest of the image. In fingerprint images, the ridge regions are the region of interest. In the proposed system, the ridge regions are identified by:

- Normalising the pixel intensity values, so the values have zero mean and unit standard deviation.
- Specifying a threshold value of the unit standard deviation of the image pixel values
- Breaking the image into blocks of the size of a specified window
- Evaluating the standard deviation in each block
- If the standard deviation of the block is above the specified threshold, then it is

considered part of the fingerprint ridges

- A mask containing the information about which regions are ridges is obtained and used for the ridge frequency estimation stage [29].

4.5.4 Ridge Orientation Estimation

The orientation of the fingerprint image unvarying coordinates of the ridges and valleys in a local neighbourhood. A local neighbourhood refers to the location of pixels relative to a center pixel. Ridge orientation is a block-wise operation, and the steps for the estimation algorithm are:

- Divide the normalised input image, G, into small blocks of size, $w \ \times w$
- Compute the x and y gradients of the pixel at the ith and jth column. The gradient is represented as $\partial_x(i,j)$ and $\partial_y(i,j)$. The x-gradient is computed using a horizontal Sobel operator, while the y- gradient is computed using a vertical Sobel operator [30].

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Horizontal Sobel operator Vertical Sobel Operator

- Estimate the local orientation of each block at pixel (i,j) using the equations [29]:

$$V_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v)$$

$$V_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} \partial^2{}_x(u,v)\partial^2{}_y(u,v)$$

$$\theta(i,j) = \frac{1}{2}tan^{-1}(\frac{V_y(i,j)}{V_x(i,j)})$$

$\theta(i,j)$ represents the least square estimate of the local ridge orientation of the block with pixel (i,j) at the center.

- Performing low-pass filtering on the oriented image by first converting it to a continuous vectors field and smoothing the image with a low-pass filter. This is done to correct any errors of the local frequency estimation of corrupted ridge blocks. The equation to convert the oriented image to a continuous vector field is [29]:

$$\varphi_x(i,j) = \cos(2\theta(i,j))$$

$$\varphi_y(i,j) = \sin(2\theta(i,j))$$

  The low pass filter equation is [29]:

$$\varphi'_x(i,j) = \sum_{u=-{}^{w_\varphi}/_2}^{{}^{w_\varphi}/_2} \sum_{v=-{}^{w_\varphi}/_2}^{{}^{w_\varphi}/_2} W(u,v)\varphi_x(i-uvw, j-vw)$$

$$\varphi'_y(i,j) = \sum_{u=-{}^{w_\varphi}/_2}^{{}^{w_\varphi}/_2} \sum_{v=-{}^{w_\varphi}/_2}^{{}^{w_\varphi}/_2} W(u,v)\varphi_y(i-uvw, j-vw)$$

- Computing and smoothing the local ridge orientation at (i,j). The equation is [30]:

$$0(i,j) = \frac{1}{2}tan^{-1}(\frac{\varphi'_y(i,j)}{\varphi'_x(i,j)})$$

4.5.5 Ridge Frequency Estimation

The ridge frequency of the fingerprint image is obtained by extracting the ridge map from the image. The steps to extracting the ridge map are:

- Dividing the normalised image, G into equal-sized blocks, $w \times w$
- Computing the oriented window, $l \times w$ of each lock

- Computing the X-signature of the ridges and valleys within the oriented window for each block [29]:

$$X[k] = \frac{1}{w}\sum_{d=0}^{w-1} G(u, v) \qquad\qquad where\ k = 0,1, \dots\dots.l-1$$

$$u = i + \left(d - \frac{w}{2}\right) cosO(i, j) + \left(k - \frac{l}{2}\right) sinO(i, j)$$

$$v = j + \left(d - \frac{w}{2}\right) sinO(i, j) + \left(\frac{l}{2} - k\right) cosO(i, j)$$

When no minutiae and singular points exist in the oriented window, the x-signature forms a discrete sinusoidal wave of the same frequency of the ridges and valleys in the oriented window. Hence, the ridge and valley frequencies can be estimated using the X-signature. Let $T(i, j)$ be the average number of pixels between the two consecutive peaks in the x-signature. The frequency is represented as:

$$\Omega(i, j) = 1/T(i, j)$$

When no consecutive peaks are detected from the x-signature, the frequency differentiated from other valid frequency values by assigning it a value -1.

- For scanned fingerprint mages with a fixed resolution, there is an expected range of frequency values. If an estimated ridge or valley frequency value is outside of this range, the frequency value is assigned a -1.

- Corrupted minutiae, singular points, ridges and valleys do not form a well-defined sinusoidal wave; hence, the frequency values for corrupted blocks must be interpolated. The interpolation is done using the equations:

$$\begin{cases} \Omega(i,j), & if\ \Omega(i,j) \neq -1 \\ \sum_{u=-{w_\Omega}/{2}}^{{w_\Omega}/{2}} \sum_{v=-{w_\Omega}/{2}}^{{w_\Omega}/{2}} W_g(u,v)\mu(\Omega(i-uw,j-vw)), & otherwise \\ \sum_{u=-{w_\Omega}/{2}}^{{w_\Omega}/{2}} \sum_{v=-{w_\Omega}/{2}}^{{w_\Omega}/{2}} W_g(u,v)\delta(\Omega(i-uw,j-vw)+1), & otherwise \end{cases}$$

Where,

$$\mu(x) = \begin{cases} 0, & x \leq 0 \\ x, & otherwise \end{cases}$$

$$\delta(x) = \begin{cases} 0, & x \leq 0 \\ 1, & otherwise \end{cases}$$

And W_g is a Gaussian kernel with zero mean, variance of 9 and the size of the kernel, w_Ω as 7 [29]

- Low pass filter to remove outliers in f'

$$F(i,j) = \sum_{u=-{w_\Omega}/{2}}^{{w_l}/{2}} \sum_{v=-{w_\Omega}/{2}}^{{w_l}/{2}} W_l(u,v)\Omega'(i-uw,j-vw)$$

4.5.6 Ridge Filtering

Gabor filters are used as a band-pass filter to remove noise using the frequency and orientation of the ridges and valleys in the fingerprint image. The form of an even-symmetric Gabor filter is represented as [29]:

$$h(x,y\colon \emptyset,f) = \exp\left\{-\frac{1}{2}\left[\frac{x_\emptyset^2}{\delta_x^2} + \frac{y_\emptyset^2}{\delta_y^2}\right]\right\}\cos(2\pi f x_\emptyset)$$

$$x_\emptyset = x\cos\emptyset + y\sin\emptyset$$

$$y_\emptyset = -x\sin\emptyset + y\cos\emptyset$$

Where $\emptyset\ and\ f$ represent the orientation and the frequency of the filter and δ_x and δ_y are the constants in the space domain of the filters along the x and y axes.

The Modulation transfer function is given as [29]:

$$H(u, v\colon \emptyset, f) = 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(u_\emptyset - u_0)^2}{\delta_u^2} + \frac{(v_\emptyset - v_0)^2}{\delta_v^2}\right]\right\}$$

$$+\ 2\pi\delta_x\delta_y \exp\left\{-\frac{1}{2}\left[\frac{(u_\emptyset + u_0)^2}{\delta_u^2} + \frac{(v_\emptyset + v_0)^2}{\delta_v^2}\right]\right\}$$

$$u_\emptyset = u\cos\emptyset + v\sin\emptyset$$

$$v_\emptyset = -u\sin\emptyset + v\cos\emptyset$$

$$u_0 = \frac{2\pi\cos\emptyset}{f}$$

$$u_0 = \frac{2\pi\sin\emptyset}{f}$$

$$\delta_u = {}^{1}\!/_{2\pi\delta_x}$$

$$\delta_v = {}^{1}\!/_{2\pi\delta_y}$$

The selection of δ_x and δ_y parameters for the filters is a trade-off as the larger the values, the more effective the filter is in removing noise; however, larger values are more likely to generate spurious ridges and valleys [29].

4.5.7 Binarisation

Binarisation is the process of transforming the fingerprint image from 256 levels to two levels, 0 and 1, i.e. black and white. This will be achieved using an adaptive local binarisation method [11] summarised in the following steps:

1. Divide the image into blocks of specified window size
2. Calculate the average intensity value of the pixels in the blocks
3. If the pixel's intensity value is greater than the mean intensity value in the block, the pixel value will be set to 1. If the intensity value is less than the mean intensity value in the block, the pixel value will be 0 [11].

4.5.8 Thinning

Thinning is performing morphological operations on a binary fingerprint image to reduce the thickness of the lines in the fingerprint. It removes redundant pixels and transforms a set of parallel pixels into a single vector.

4.6 Feature Extraction

A minutiae-based feature extraction algorithm will be used on the pre-processed fingerprint image. The algorithm will identify the key features such as ridge endings, bifurcation, lakes, dots, spurs and crossovers, as seen in fig 8.

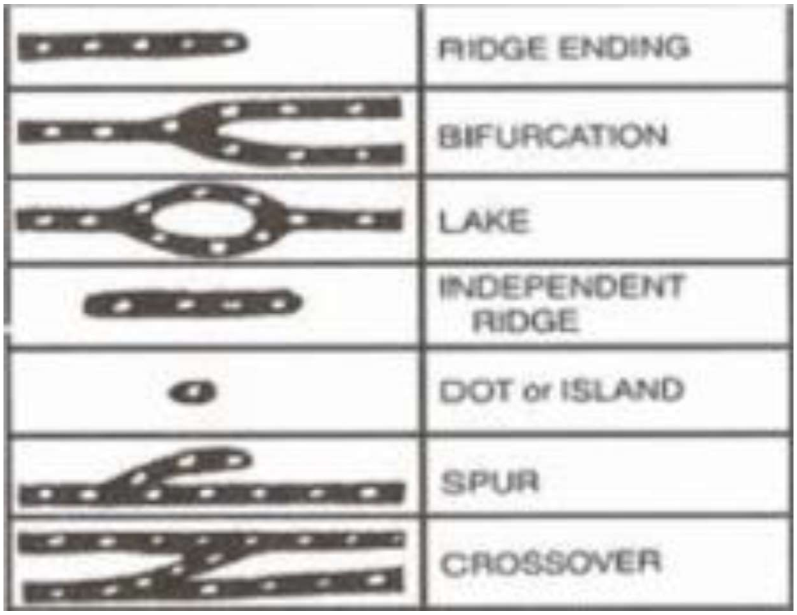


Figure 8 Types of Minutiae [14]

4.7 Matching

Two Open CV image processing algorithms, Flann-based matcher with SIFT extractor and Brute force matcher with SIFT extractor, were explored for a match. A common points matching algorithm that calculates the distance between the different key points (minutiae) extracted from each fingerprint image was tested. A convolutional network approach for image classification was tested for matching performance.

4.7.1 Common points Matching algorithm

This algorithm attempts to find the common points in fingerprint images with the set of extracted minutiae points, N1 and N2. The algorithm finds a tuple, M, representing the intersections of the points N1, N2 i.e. $M = N1 \cap N2$. For ith minutiae point, M(i) represents a tuple representing unique information about minutiae point, i. Two sets of Minutiae tuples, M(i) – tuples can be compared for common points to find a match in an input and base image. The steps to finding the tuples of minutiae in an image are [34]:

- Find the five nearest minutiae points to the minutiae from i = 1 to N1 using their Euclidean distances.
- For the set of nearest points to minutiae point i: {i1, i2, i3, i4, i5}, calculate 10 ratios of the distance between two points: (i-i1): (i1-i2), (i-i1): (i1-i3), (i-i4): (i1-i5), (i-i2): (i1-i3), (i-i3): (i1-i4), (i2-i5): (i-i5), (i-i3): (i-i4), (i-i3): (i-i4), (i-i3): (i-i5), (i-i4): (i-i5). The ratio of the distances is calculated using the equation [34]:

$$(a-b):(a-c) = \frac{Max\,\{(a-b),(a-c)\}}{Min\,\{(a-b),(a-c)\}}$$

- Calculate the angle formed between ‘bac’ or ‘cab’ when finding the ratio. The angle between the ratio (i-i1): (i1-i2), for example, is found by extending any edge, e.g. (i-i1) extended to (i1-i-extended line) forming 180° as seen in fig. 9. Angle 2 represents the angle from (extended line –i-i2), and the angle need is the angle from i1-i-i2 or i2-i-i1 [34].

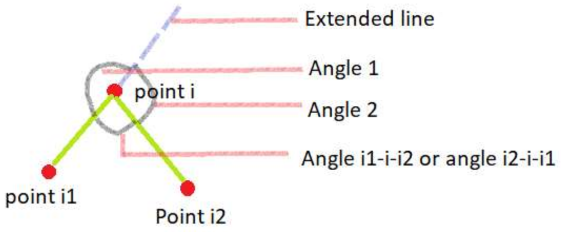


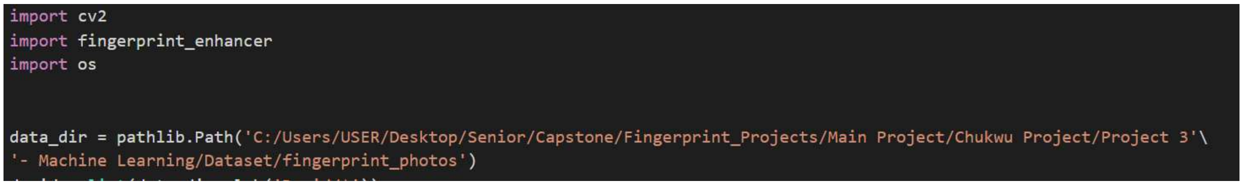
Figure 9. angle between edges of minutiae points

- Store the angles and the ratio values in a tuple.

Two sets of tuples (ratios and angles) from an input image and a base image are considered to be the same if 2 or more ratios and their angles match [34].

4.7.2 Convolutional Neural Network Classification

A convolutional neural network approach for image classification was explored to identify the captured latent prints. It involved four steps: building a pipeline for the input fingerprint images, building the model, training the model and testing the model.

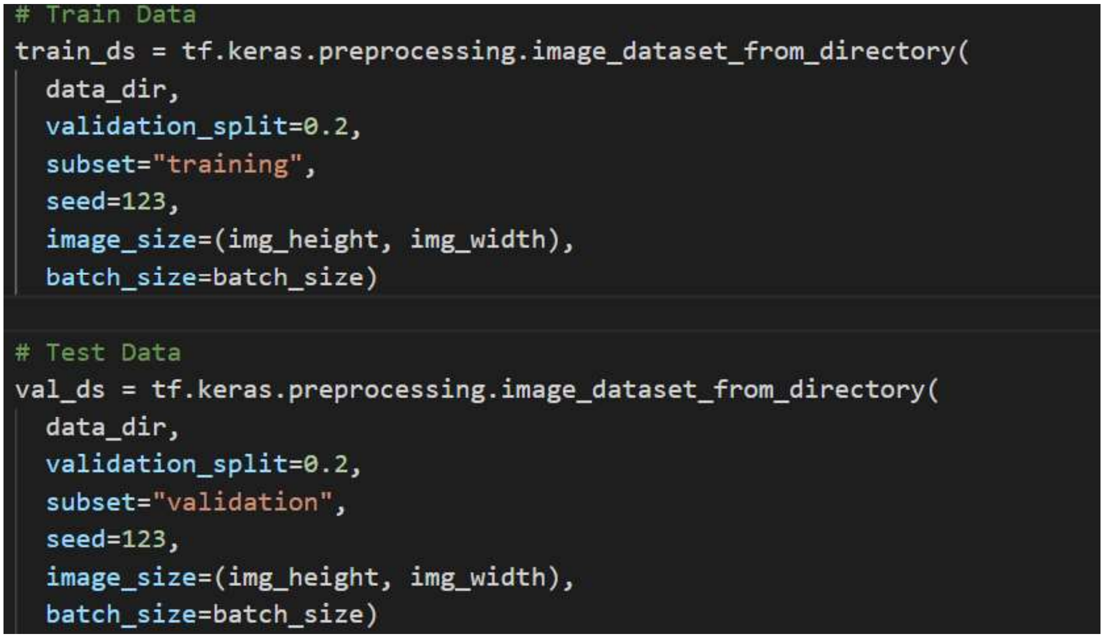


```
import cv2
import fingerprint_enhancer
import os


data_dir = pathlib.Path('C:/Users/USER/Desktop/Senior/Capstone/Fingerprint_Projects/Main Project/Chukwu Project/Project 3'\
'- Machine Learning/Dataset/fingerprint_photos')
```

Figure 10. Loading dataset

The dataset was split by 20% into a test and validation set.

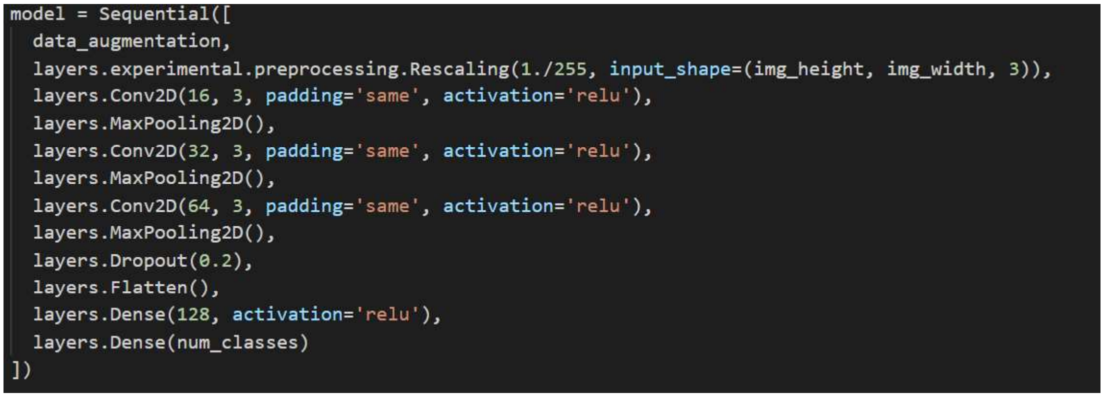


```
# Train Data
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="training",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)

# Test Data
val_ds = tf.keras.preprocessing.image_dataset_from_directory(
  data_dir,
  validation_split=0.2,
  subset="validation",
  seed=123,
  image_size=(img_height, img_width),
  batch_size=batch_size)
```

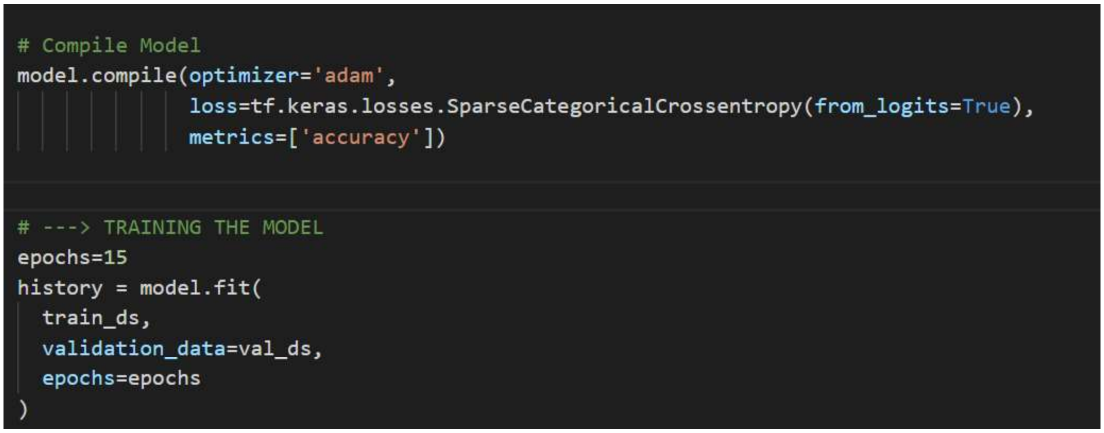
Figure 11. Splitting the data into train and validation

The dataset was resized to a specific width and height of 180 pixels each. The model was built with 12 layers and three convolution blocks, each having a max pool layer. Special layers, augmentation and dropout layers were added to address overfitting, i.e. the large difference between the training and validation accuracy. The augmentation layer was included to generate more training data from the initial dataset by randomly transforming the images. The dropout layer handled the regularisation of the data [33].



```
model = Sequential([
  data_augmentation,
  layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
  layers.Conv2D(16, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(32, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Conv2D(64, 3, padding='same', activation='relu'),
  layers.MaxPooling2D(),
  layers.Dropout(0.2),
  layers.Flatten(),
  layers.Dense(128, activation='relu'),
  layers.Dense(num_classes)
])
```

Figure 12. Defining the model



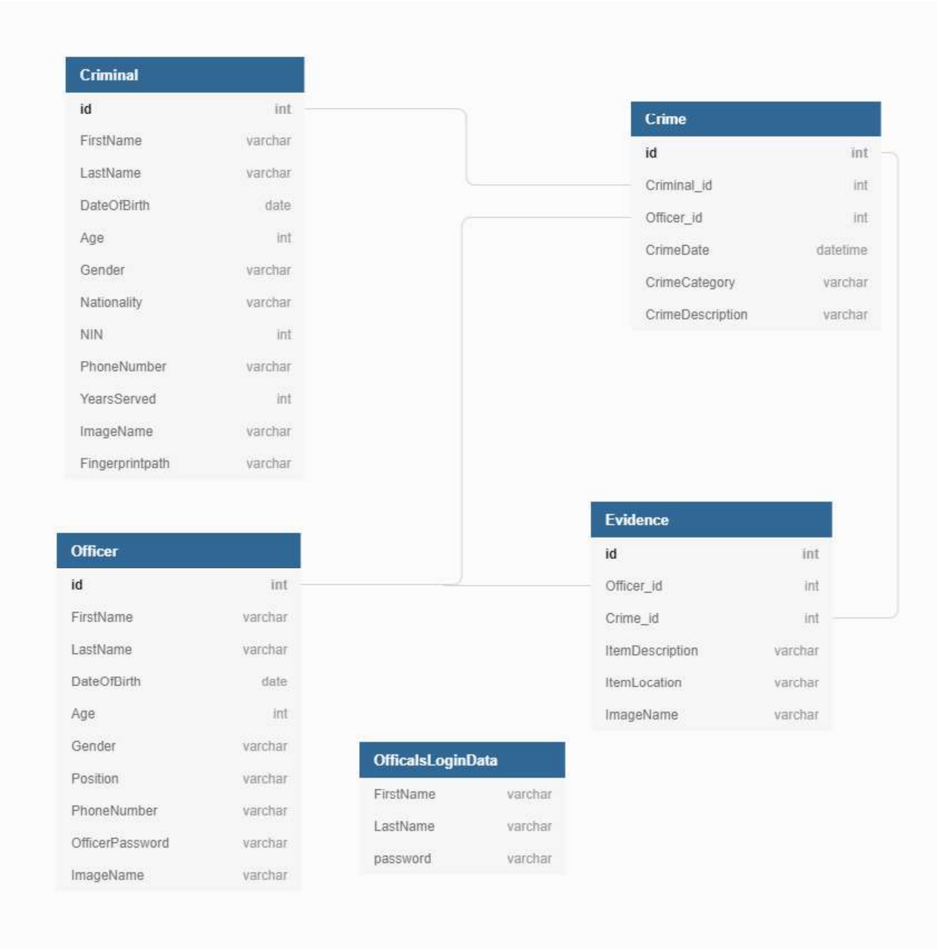
```
# Compile Model
model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])


# ---> TRAINING THE MODEL
epochs=15
history = model.fit(
  train_ds,
  validation_data=val_ds,
  epochs=epochs
)
```

Figure 13. Compile and Training the model

4.8 Database

The database is created using structured query language, SQLAlchemy. SQLAlchemy is python’s toolkit for SQL and an Object Relational Mapper (ORM). It was selected for the application because it provides smooth communication with the python backend application by providing an efficient way to map a database schema to the Python application. It also provides more efficient database access through simple python queries [32]. The database consists of 5 tables, a criminal table, an officer table, a crime table, an evidence table and an officials Login table. The crime table has foreign keys associated with the criminal and officer table with a many to one relationship for both tables. The evidence table has a foreign key associated with the officer table with a many to one relationship. The fields of each table and the relationship between the five tables are shown in fig 14.



Criminal	
id	int
FirstName	varchar
LastName	varchar
DateOfBirth	date
Age	int
Gender	varchar
Nationality	varchar
NIN	int
PhoneNumber	varchar
YearsServed	int
ImageName	varchar
Fingerprintpath	varchar

Crime	
id	int
Criminal_id	int
Officer_id	int
CrimeDate	datetime
CrimeCategory	varchar
CrimeDescription	varchar

Officer	
id	int
FirstName	varchar
LastName	varchar
DateOfBirth	date
Age	int
Gender	varchar
Position	varchar
PhoneNumber	varchar
OfficerPassword	varchar
ImageName	varchar

Evidence	
id	int
Officer_id	int
Crime_id	int
ItemDescription	varchar
ItemLocation	varchar
ImageName	varchar

OfficalsLoginData	
FirstName	varchar
LastName	varchar
password	varchar

Figure 14. Database design showing tables and relationships between the tables

,

Chapter 5: Testing and Results

The implemented system testing was done in 3 stages, unit testing, component testing, and system testing.

5.1 Unit testing

Tests were performed on a fingerprint enhancer algorithm and two feature extraction algorithm, and matching algorithms. The feature extraction algorithms were compared to determine which algorithm accurately records all possible minutiae points in a fingerprint image. A total number of 3 persons (2 females, named Gloria and Tito. and a male named David) made ten fingerprints impressions on a white mug. This item was selected as an everyday household item that can be found at a crime scene. A fingerprint brush was dipped into a fingerprint powder and dusted on the items' surface to make the print impressions visible. Once the latent prints were visible, a range of 2-10 captures of the impressions were taken incrementally; a number in the range of 60-300 latent images was available at each stage of testing. The images were captured using an iPhone 11, transferred to a PC, and stored in a JPEG format.

The same 3 participants were selected to obtain scanned fingerprint images for comparison and to be enrolled in the criminal database. A range of 3-10 images of each of the three participant's ten fingers was scanned incrementally using the Digital Persona U Are U 4500 optical fingerprint reader and stored in a PNG format. Hence, a number in the range of 90-300 scanned images was available at each testing stage.

5.1.1 Fingerprint Enhancement Algorithm

A good fingerprint image has its ridges clearly defined with no background noise, smudges, falsely connected ridges or any other distortion form. A bad fingerprint image has its ridges less defined, has some background noise and other distortions. An ugly fingerprint image has high amounts of distortions, noisy backgrounds and unrecoverable regions difficult to process. The collected latent and scanned fingerprint images were divided into good, bad, and ugly categories. Samples of the image categorisation are shown in fig 15.

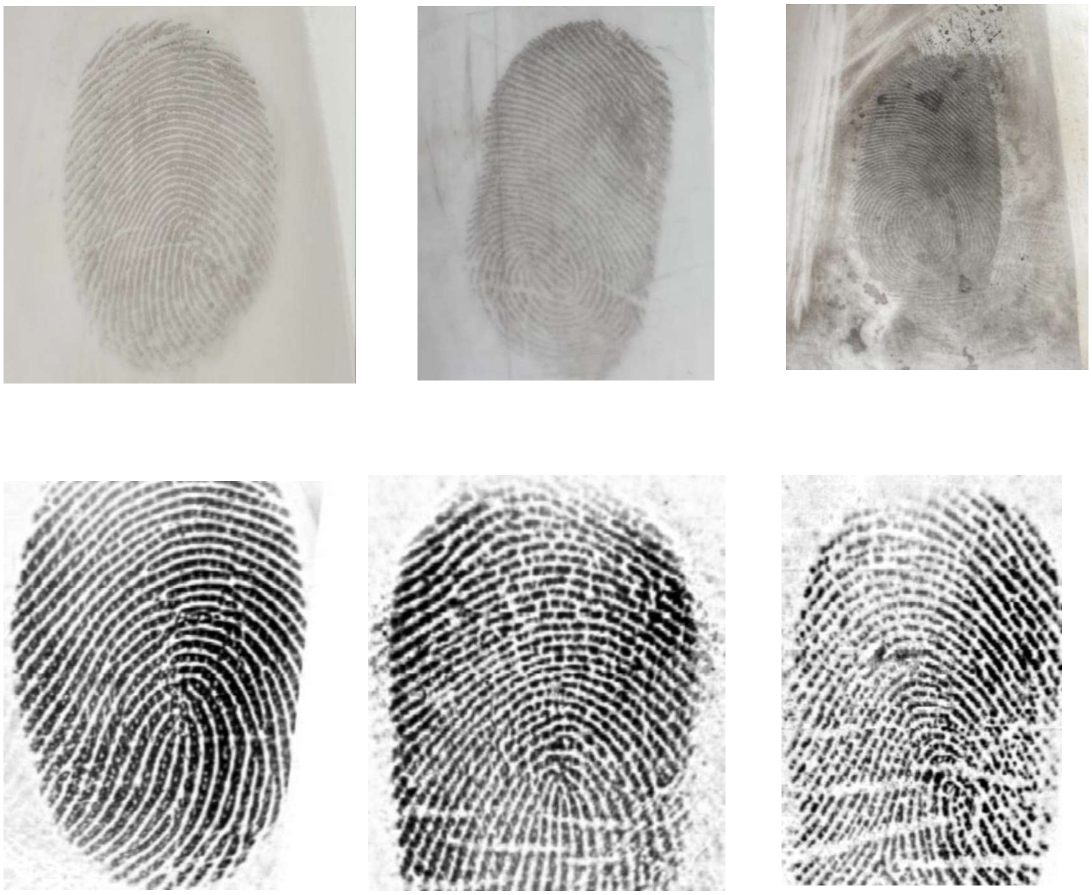


Figure 15. good, bad, ugly latent fingerprints from left to right (top row). Good, bad and ugly scanned prints from left to right (bottom row)

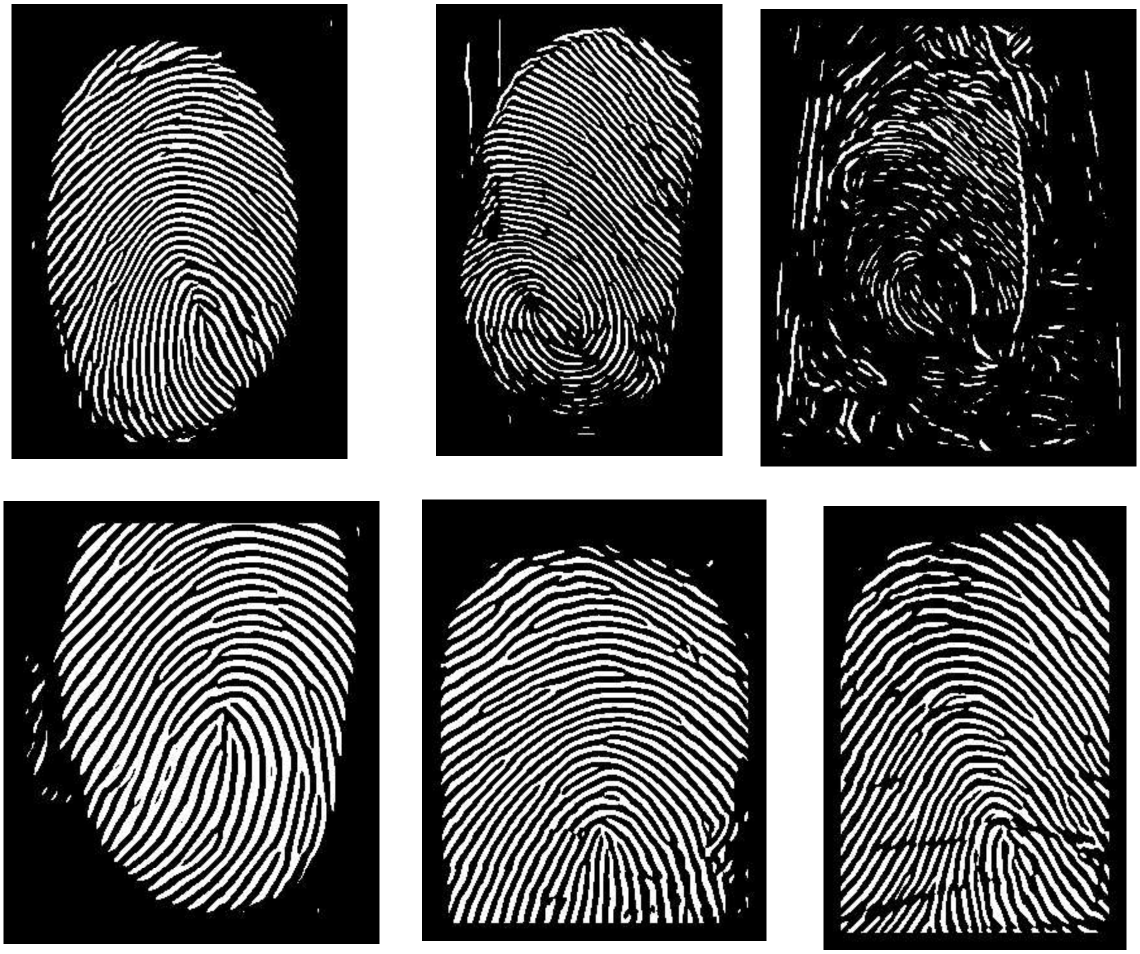


Figure 16. Enhanced latent images good, bad and ugly from left to right (top row). Enhanced scanned images good, bad and ugly, from left to right (bottom row).

As seen in fig 16, the fingerprint enhancer algorithm performed better on good, bad and ugly scanned fingerprint images. For the latent print images, the ridges in the good fingerprint images are more defined than the ridges in the bad latent fingerprint images. The smudges in the bad latent fingerprint images produced holes in the enhanced image. The enhancer performed poorly on the ugly image, creating an enhanced image with no defined ridges or features to extract.

For the scanned images, the enhancer performed extremely well on the good scanned image. On the bad and ugly scanned images, the enhancer connected broken ridges in the fingerprint; however, it could not reconstruct ridges corrupted by smudges, dirt or lighting distortions.

The enhancer algorithm is powerful enough to reconstruct broken ridges using the fingerprint orientation and frequency; however, it fails for certain amounts of distortion. The accuracy of the feature extraction stage of the images heavily depends on the image quality and the enhancement of the fingerprint image. Certain factors such as sensor noise, sensor quality, cleanliness of the scanner plate, and the person's attitude can affect a scanned fingerprint image. For latent fingerprint images, the feature extraction algorithm can exclude regions that fall outside of the fingerprint image; however, the algorithm works best for good latent fingerprint images.

5.1.2 Feature Extraction Algorithm A

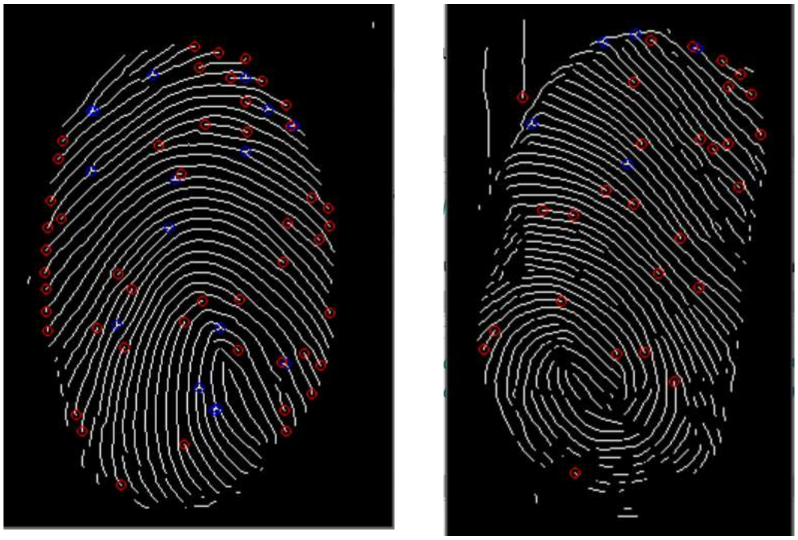


Figure 17. Features extracted from good and bad latent fingerprints (feature extraction A)

As seen in fig. 17 the feature extraction algorithm could identify the ridge endings (represented as red circles) and the ridge bifurcations (represented as blue circles). The algorithm ignored the distorted areas in the bad latent fingerprint image and extracted some features from the more defined areas of the image. The algorithm also ignores lines outside of the fingerprint regions. However, the algorithm records false minutiae seen at the beginning (borders) of the fingerprint images.

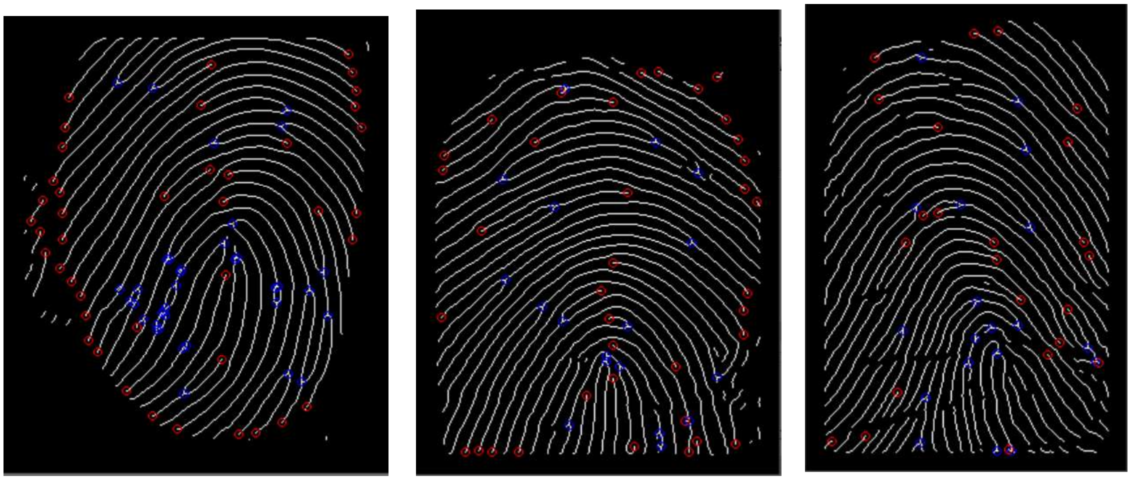


Figure 18. Features extracted from good, bad and ugly scanned fingerprints (feature extraction A)

The feature extraction algorithm can extract the ridge endings and the ridge bifurcations of the scanned images. It extracts ridge endings from line outside of the fingerprint image for the good and bad scanned fingerprint image (possibly fingerprint residue left on the scanner before a new scan). It records broken ridges as ridge endings for the bad and ugly fingerprint images; hence, it extracts some false minutiae points.

5.1.3 Feature Extraction Algorithm B

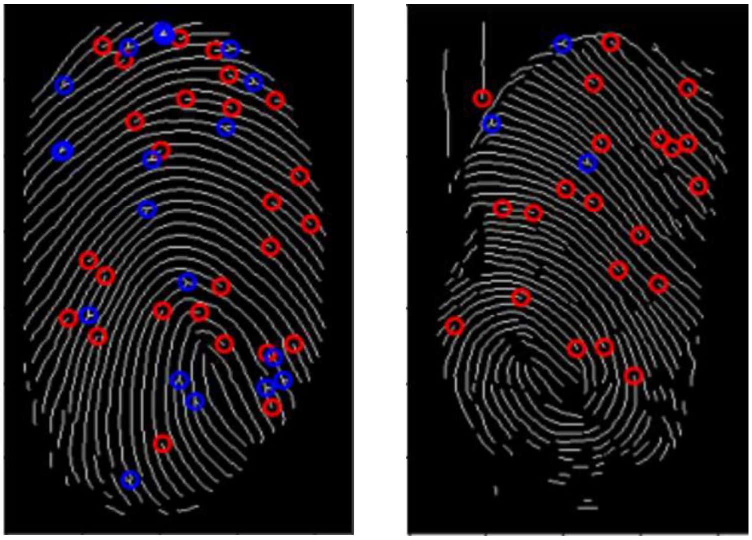


Figure 19 Features extracted from good and bad latent fingerprints (feature extraction B)

Feature extraction algorithm B performs post-processing on extracted minutiae points from feature extraction algorithm A. As seen in fig. 19., the termination (red circles) and bifurcation (blue circles) points are retained. The false-minutiae points, i.e. beginning of ridges recorded as ridge endings, seen in fig 17., are removed. However, the minutiae points from a few falsely connected ridges are recorded.

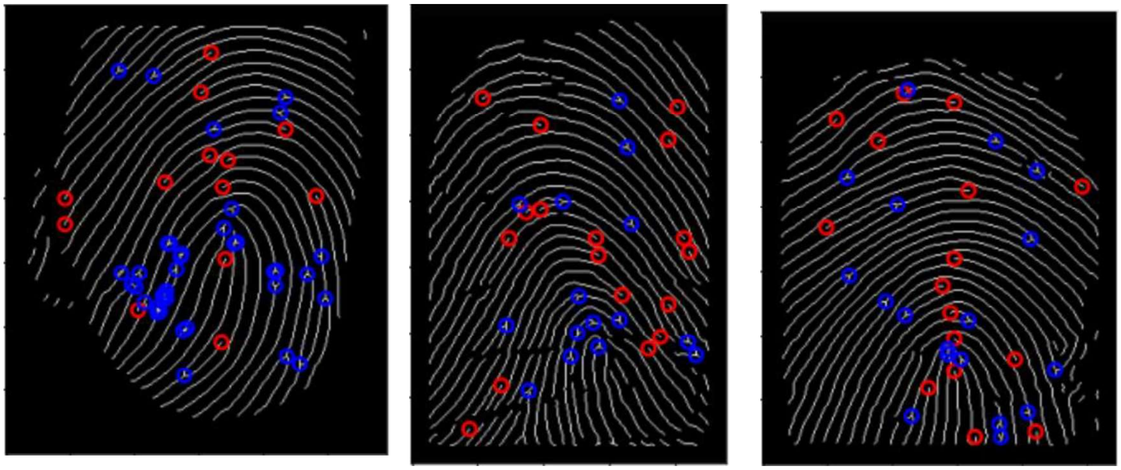


Figure 20 Features extracted from good and bad latent fingerprints (feature extraction B)

For the scanned images seen in fig 20. Noise from the background image as well as ridge beginnings is recorded as terminations (red circles). However, feature extraction algorithm B removes these false minutiae points and retains the actual minutiae points mirroring the true minutiae points from fig. 18 exactly. Feature extraction B performed better than feature extraction A and was used in the next stages of matching.

5.2 Component testing

The common points matching algorithm and two matching algorithm sets from Open CV Python library, Flann-Based Matcher, and the Brute Force Matcher, are tested for matching percentages. Two participants, Gloria and David were selected from the dataset and 3 copies of both latent and scanned images of each of their 10 fingers were selected. The images of each of the ten fingers of the 2 participants were compared for a match using the following categories:

Table 1. Testing levels for the fingerprint identification component

Input	comparison	label
Same print scanned image	Same print scanned image	SSA
Print scanned image	Different image but same print scan	SSB
Print scanned image	Different prints scan	SSD
Same print latent image	Same print latent image	LLA
Print latent image	Different image but same print latent	LLB
Print latent image	Different prints latent	LLD
Same print scanned image	Same print latent image	LS
Print scanned image	Different print latent image	LD

5.2.1 Flann-Based Matching algorithm with SIFT

Table 2. Test Case 1: Gloria (raw) Flann-Based Matcher with SIFT

	SSA (%)	SSB (%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	100	13.56	1.49	100	21.67	1.39	1.09	0.65
Left index	100	13.95	1.11	100	34.78	1.11	1.76	0.78
Left middle	100	13.18	1.34	100	35.98	1.09	1.95	1.39
Left ring	100	13	1.56	100	43.67	1.67	1.84	1.21
Left pinkie	100	13.65	1.32	100	25.88	1.32	1.77	1.34
Right thumb	100	13.54	1.22	100	27.67	1.45	1.75	1.45
Right index	100	13.69	1.24	100	30.56	1.23	1.45	0.88
Right middle	100	13.78	1.45	100	39.55	1.55	1.12	0.96
Right ring	100	13.75	1.34	100	29.46	1.67	1.35	1.05
Right pinkie	100	13.23	1.55	100	28.89	1.36	1.56	1.18
Average	100.00	13.53	1.36	100.00	31.81	1.38	1.56	1.09

As seen in table 1., the Flann-Based Matcher performs well for matching a scanned print image to itself, as expected, with matches of 100%. The Flann-based matcher records match percentages 13% to 14% for matching the same print but different scanned image. While matching the same print but different captured latent image, the matcher records match percentages of 21% to 44%. For matching different scanned prints, the matcher records percentages of 1.1% to 1.6%. While for different captured latent prints, it records percentages 0.6% to 1.5%. The matcher records percentages from 1% to 2% for matching the same print and a captured latent print to a scanned print.

Though the SIFT feature extraction algorithm extracts local features from the image, it does not extract the fingerprint image's biometric features. The Flann-Based Matcher records a match using a nearest-neighbour classification of the SIFT extracted key points. It performs great for the same images as expected for any image feature extraction algorithm. The matching percentages are too low to record a true match for matching the same prints, but different scan takes. This can be attributed to the nature of the features extracted. Different scans would have different local points that are specific to the image. Also, the scanned images cover different parts of a finger. Due to the scanning surface area, some scans cover the bottom part of the fingerprint more than the top and vice versa. Hence, comparing similar scans will result in low percentage similarities for varied features detected by the SIFT algorithm. However, the percentages for matching similar latent prints are significantly higher. This is because the latent prints are more similar as the images' difference is due to fingerprint to background ratio, lighting and size.

Expectedly, the algorithm set fails to record matches for both different scanned and latent prints. However, it cannot match a captured latent print to a scanned print of the same finger from the participant Gloria.

Table 3. Test Case 2: David (raw) Flann-Based Matcher

	SSA (%)	SSB (%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	100.00	12.07	1.52	100.00	98.17	1.45	1.56	1.77
Left index	100.00	12.56	1.78	100.00	98.19	1.56	1.59	0.85
Left middle	100.00	13.05	1.45	100.00	98.73	1.39	1.67	1.90
Left ring	100.00	12.45	1.55	100.00	96.77	1.36	1.85	2.31

Left pinkie	100.00	12.44	1.67	100.00	97.55	1.66	1.80	0.89
Right thumb	100.00	12.39	1.79	100.00	95.67	1.45	1.67	1.16
Right index	100.00	12.05	1.73	100.00	96.66	1.36	1.59	2.89
Right middle	100.00	13.55	1.69	100.00	99.55	1.67	1.64	2.35
Right ring	100.00	12.91	1.90	100.00	100	1.44	1.85	1.70
Right pinkie	100.00	12.32	1.53	100.00	99.45	1.37	1.62	1.09
Average	100.00	12.58	1.66	100.00	98.07	1.47	1.68	1.69

In table 3., the Flann Based Matcher performs excellently well for matching for same scanned images and latent images for the participant, David. The latent prints collected from David were initially of better quality (less noisy, sharper picture) than Gloria, and so the algorithm set performed better for David's captured latent prints overall. However, it still fails to record a match for a captured latent print to a scanned print of David's same finger.

5.2.2 Brute-Force Matching algorithm

Table 4. Test Case 3: Gloria (raw) Brute-Force Matcher

	SSA (%)	SSB(%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	98.10	13.29	1.46	96.45	21.02	1.36	1.07	0.63
Left index	98.32	13.77	1.08	98.90	34.41	1.09	1.73	0.76
Left middle	98.43	12.92	1.31	98.08	35.18	1.07	1.90	1.36
Left ring	98.22	12.83	1.54	97.72	42.70	1.64	1.80	1.17
Left pinkie	98.17	13.38	1.29	98.97	25.36	1.29	1.73	1.31
Right thumb	98.22	13.30	1.20	97.99	26.84	1.42	1.70	1.42
Right index	98.11	13.57	1.22	97.78	29.58	1.21	1.42	0.86

Right middle	98.12	13.54	1.42	98.44	38.67	1.52	1.09	0.93
Right ring	98.38	13.48	1.31	94.32	28.77	1.64	1.31	1.02
Right pinkie	98.29	12.97	1.51	98.34	20.99	1.33	1.51	1.16
Average	98.24	13.30	1.33	97.70	31.39	1.36	1.53	1.06

Table 5. Test Case 4: David (raw) Brute Force Matcher

	SSA (%)	SSB (%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	97.19	11.83	1.48	96.99	93.37	1.41	1.53	1.73
Left index	98.67	12.18	1.74	97.77	92.09	1.53	1.57	0.83
Left middle	99.15	12.66	1.43	98.00	94.33	1.35	1.62	1.86
Left ring	98.55	12.17	1.52	97.13	90.87	1.35	1.79	2.29
Left pinkie	97.78	12.04	1.63	97.55	97.55	1.63	1.76	0.86
Right thumb	97.06	12.26	1.75	98.54	91.94	1.40	1.62	1.13
Right index	98.77	11.92	1.69	99.45	91.82	1.32	1.55	2.81
Right middle	97.45	13.12	1.66	98.32	95.95	1.62	1.62	2.30
Right ring	95.48	12.75	1.84	98.34	98.05	1.40	1.79	1.67
Right pinkie	96.65	12.24	1.79	97.27	95.42	1.12	1.67	2.09
Average	97.68	12.32	1.64	97.94	94.14	1.44	1.65	1.72

For both participants, David and Gloria, the Brute Force Matcher records about 2% lower for all comparison case. Both Algorithms, however, are pure image processing algorithms. They are not optimised to match raw captured latent prints to their equivalent scanned prints. This is due to the high variance of orientation, alignment, size, finger area, fingerprint to background ratio and noise. After enhancing the raw images, the algorithms fail by recording non-matches as matches for the different prints. The enhancement makes the images look as similar as possible, and so key points from different images will be seen as a match.

5.2.3 Common points Matching algorithm

Table 6 Common points matching algorithm (Gloria)

Gloria								
	SSA (%)	SSB (%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	100	7.12	0	100	24.59	0	0.12	0
Left index	100	10.55	0	100	3.67	0	0	0
Left middle	100	6.84	0	100	0.23	0	0	0
Left ring	100	9.45	0	100	7.77	0	0.67	0
Left pinkie	100	6.38	0	100	0	0	0	0
Right thumb	100	3.31	0	100	13.89	0	0	0
Right index	100	0	0	100	6.75	0	0	0
Right middle	100	7.65	0	100	0.55	0	0	0
Right ring	100	2.19	0	100	1.22	0	0	0
Right pinkie	100	9.9	0	100	0.13	0	0	0
Average	100	6.34	0	100	5.88	0	0.08	0

As seen in table 6., the common points matching algorithm can match the same scanned and same latent images for Gloria with 100%. The average match percentage for the same print different scan for Gloria was 6.34%. The match percentages were between 2.2% to 10.6%, and one print returned 0%. This algorithm records no matches for different scanned and latent print images for Gloria. The same print different captured latent prints match with an average of 5.88% for Gloria. The values fall in a wide range from 0% to

24.6%. The algorithm performs poorly when matching the same captured latent print to its scanned print with an average of 0.08%.

Table 7. Common points matching on David

David								
	SSA (%)	SSB (%)	SD(%)	LLA (%)	LLB (%)	LLD(%)	LS (%)	LD(%)
Left thumb	100	7.12	0	100	18.18	0	0	0
Left index	100	5.55	0	100	1.3	0	0	0
Left middle	100	1.34	0	100	0	0	1.23	0
Left ring	100	2.45	0	100	7.85	0	0	0
Left pinkie	100	6.38	0	100	2.55	0	0	0
Right thumb	100	1.67	0	100	0	0	0	0
Right index	100	5.38	0	100	11.35	0	0	0
Right middle	100	18.65	0	100	2.58	0	1.45	0
Right ring	100	10.33	0	100	12.78	0	1.67	0
Right pinkie	100	5.89	0	100	0.29	0	1.55	0
Average	100	6.48	0	100	5.69	0	0.59	0

The algorithm records a 100% match for the same scanned and captured latent images for David. It records 0% matches for different scanned and captured latent images. For the same print but different scanned images, the average match percentage was 6.48%. For different captured latent images but same prints, the average match percentage was 5.69%. Though the algorithm performed slightly better for the same print latent to scanned images for David, it recorded a very low match percentage of 0.59%.

Though the common points algorithm uses biometric features, i.e. minutiae points to match, it does not account for variables such as:

- **The surface area of the scanner:**

  The scanner has a relatively small surface area, and for bigger fingers such as the thumb and ring, it does not capture full images. Hence, a comparison of full to partial images will have fewer common points and will record low match percentages.

- **Different parts of a fingerprint being compared:**

  In a scan to scan comparison, different takes of a fingerprint show different parts of the fingerprint. Attempting to match minutiae points with significantly different parts of a scanned image will result in a low match percentage.

- **Alignment issues**

  The scanned images differ in alignment; hence the ratios and their corresponding angles will differ across the same print. Attempting a similar scanned match or the same print latent to a scanned match would result in low match percentages.

- **The ratio of the fingerprint image to the background**

  The fingerprint area in the picture affects the x and y coordinates of the extracted minutiae points. When attempting to find common points for images with varying fingerprint to background ratio, the tuples (ratio and angles) would differ. They can result in a low or no match percentage.

5.2.4 Convolutional Neural Network matching algorithm

A dataset of scanned and latent fingerprint images of participants Gloria and David were collected for classification. The dataset consists of 76 scanned images and 67 latent images for David making 143 fingerprint images for David. There were 68 scanned images and 103 latent images for Gloria making 171 fingerprint images for Gloria. 16 latent images for Tito (not included in the model) and 16 scanned images for Tito (not included in the model) were used for testing. The total dataset of 314 photos was split by 20%, where 252 files were used for training and 62 files were used for testing. For the dataset of 252 files, match percentages ≥ 92% are recorded as a good match.

Table 8. Neural network test on raw images

Raw images					
Latent of David match	Scanned of David match	Latent of Gloria match	Scanned of Gloria match	Latent of Tito match	Scanned of Tito match
92.84 % David	99.77% David	93.63% Gloria	98.66% Gloria	75.23% Gloria	93.89% David
93.01% David	99.85% David	95.23 % Gloria	99.96% Gloria	83.30% Gloria	99.75% David
93.07% David	99.70% David	91.50 % Gloria	99.44% Gloria	89.86% Gloria	91.20% David
92.89% David	99.69% David	91.15 % Gloria	98.79% Gloria	80.53% Gloria	92.22% David
90.68% Gloria	99.78% David	69.27% Gloria	77.35% David	86.17% Gloria	62.30% Gloria
98.11% David	100% David	89.28% Gloria	88.66% Gloria	97.66% Gloria	99.81% David
54.02% Gloria	99.90% David	95.36% Gloria	93.10% David	53.40% Gloria	99.75% David
86.09% David	99.98% David	95.43% Gloria	99.48% Gloria	64.79% Gloria	99.99% David
63.61% Gloria	100% David	94.20% Gloria	99.81% Gloria	83.55% Gloria	92.29% David

53.50% David	99.88% David	72.07% David	88.66% Gloria	80.80% Gloria	93.73% David
98.41% David	100% David	50.35% Gloria	93.24% Gloria	64.79% Gloria	89.07% David
98.40% David	99.90% David	95.95% Gloria	84.86% Gloria	98.41% Gloria	83.47% David
96.67% David	100% David	91.51% Gloria	56.95% David	96.41% Gloria	85.57% David
68.75% Gloria	99.98% David	96.43% Gloria	55.44% Gloria	87.76% Gloria	70.83% Gloria
94.51% David	99.78% David	92.49% Gloria	100% Gloria	82.09% Gloria	95.64% David

As seen in table 8., the neural network performs fairly well in classifying latent images for David. Occasionally, it records lower percentages of about 60% to 70% for Gloria. This is attributed to the higher number of images for Gloria than David creating a bias. The scanned images of David were all classified as David with percentages in the range 99% to 100%. The algorithm performs reasonably well in classifying latent images for Gloria as well. Occasionally it classifies Gloria's prints as David but with lower percentages between 50% to 75%. It performs fairly well for scanned images for Gloria however it classifies some scanned images for Gloria as David with a wider range of 55% to 94%. The algorithm classifies all of Tito's latent prints as Gloria with percentages between 53% to 98%. It fails for scanned images of Tito by classifying many scanned images for Tito as David with high percentages in the range of 80% to 100%.

The false classification of the scanned images for Tito can be attributed to the following reasons:

- The scanned images for Tito are noisy. The images taken for Tito at the initial testing stages have higher amount of distortion than David and Gloria's scanned images. The local features of the image may not be well defined enough to be accurately distinguished by the model.
- The cross similarities in the global features of fingerprint images would make it difficult for the model to accurately exclude images not in the model

For this system, a true positive match is a match for David on a scanned or latent David fingerprint image and a match for Gloria on a scanned or latent Gloria fingerprint image. A true negative match is a no match for either David or Gloria for a random scanned or latent fingerprint image. A false negative is a no match for David on a scanned or latent David fingerprint image and a no match for Gloria on a scanned or latent Gloria fingerprint image. A false positive match is a match for Gloria on any fingerprint image that does not belong to Gloria and a match for David on any fingerprint image that does not belong to David.

Table 9. Summary of true positive, true negative, false positive and false negative matches for the raw image classification

Raw images					
Latent of David match	Scanned of David match	Latent of Gloria match	Scanned of Gloria match	Latent of Tito match	Scanned of Tito match
True positive	True positive	True positive	True positive	True negative	False positive
True positive	True positive	True positive	True positive	True negative	False positive
True positive	True positive	False negative	True positive	True negative	True negative
True positive	True positive	False negative	True positive	True negative	False positive
False negative	True positive	False negative	False negative	True negative	True negative
True positive	True positive	False negative	False negative	False positive	False positive
False negative	True positive	True positive	False positive	True negative	False positive
False negative	True positive	True positive	True positive	True negative	False positive
False negative	True positive	True positive	True positive	True negative	False positive
False negative	True positive	False negative	False negative	True negative	False positive

True positive	True positive	False negative	True positive	True negative	True negative
True positive	True positive	True positive	False negative	False positve	True negative
True positive	True positive	False negative	False negative	False positive	True negative
False negative	True positive	True positive	False negative	True negative	True negative
True positive	True positive	True positive	True positive	True negative	False positive

Accuracy test

The accuracy for the system is defined by the measure of the degree of closeness of the true positive matches and true negative matches recorded by the system and the actual true positive matches and true negative matches available. The equation for calculating the accuracy of the system is:

$$accuracy(\%) \ = \frac{True\ positive + True\ negative}{Total\ number\ of\ elements} \times 100$$

$$accuracy(\%) \ = \frac{40 + 18}{90} \times 100 = \frac{16}{24} \times 100 = 64.44\%$$

The raw image classification has an accuracy of 64.44% for the dataset of 252 images and a test against 60 images from the database and 30 images outside of the dataset. The accuracy could be improved by collecting more fingerprint images of different parts of a finger to improve the model and reduce the cross similarities recorded.

Run time Test

Table 10. Run time values for Neural network matching algorithm on raw images

Raw images	
Run	Time(s)
Run 1	91.34
Run 2	91.52
Run 3	94.64
Run 4	93.84
Run 5	93.06
Run 6	92.24
Run 7	93.39
Run 8	92.53
Run 9	94.72
Run 10	92.86
Run 11	92.99
Run 12	92.09
Run 13	93.20
Run 14	93.36
Run 15	94.54
Run 16	93.67
Run 17	91.94
Run 18	94.07
Run 19	92.25
Run 20	91.44
Average	92.98

The average run time for the neural network classification for the dataset of 252 files is 92.28 seconds – just over a minute and 30 seconds.

The Convolutional Neural Network classification algorithm performed better than the image processing algorithm and was used in the final system.

5.3 System testing

The system test was performed to evaluate how the whole system functions with all the various components and whether it meets the requirements specified in chapter 3. The registration of users and criminals, logging in by officers, sending and receiving match results and criminal information was tested on different days by five users on different multiple times on the HP Pavilion laptop. The different activities were tested on a pass-fail metric where 1 represents a pass and 0 represents a fail.

Table 11. End-to-End Test of the latent fingerprint identification system

Activity	User 1	User 2	User 3	User 4	User 5	Total
Officer registration	1	1	1	1	1	5
Criminal registration	1	1	1	1	1	5
Officer log-in	1	1	1	1	1	5
Evidence send	1	1	1	1	1	5
View Match results	1	0	0	1	1	3
View Criminal (suspect information) information	1	0	0	1	0	2

The results from table 12 show how the components of the system performed on different days with different users. The logging in, registration, and sending functionalities proved to work consistently for the five users. The functionality to view matches failed on two occasions due to the inconsistent behaviour of the fingerprint processing script. Consequentially, the functionality to view suspect/criminal information failed twice due to the view matches failure for two users and once due to a server crash on the testing environment.

The project was able to meet the following requirements and specifications:

- Presents a well-organised collection of data
- Easy to use
- Has a sensor device to capture images from crime scenes and a sensor device to scan and enrol fingerprint at a police office. The devices should have high effective resolutions and pixel densities to obtain good quality images.
- Has a web application with front end applications and a server-side application to send captured data, interact with the database, view results of the fingerprint processing.
- Has a web application with a front-end application to secure registration of the criminal department officials into the system, register criminals, and view criminal information.
- The system should be secure with officer user login credentials.

Chapter 6: Conclusions and Future Work

6.1 Conclusion

The project aimed at providing an on-site latent fingerprint identification system to improve the criminal investigation process in Nigeria. It aimed at implementing an accurate identification algorithm to identify suspects of a crime and tackle issues of loss of evidence and lack of criminal records. The implemented system provides a latent fingerprint identification system with a Convolutional Neural Network classification model. The model proposes for the criminal department to collect both scanned and captured latent prints similar to those at crime scenes from the known criminals. The system can be used at a crime scene to send captured fingerprint evidence to be processed and stored with other relevant information about the evidence. The processed fingerprint is compared against a database of existing fingerprint images to identify a crime suspect with 64.44% accuracy. In the event of a no-match in the system, the fingerprint evidence is stored for future identification. It tackles the loss of evidence and lack of criminal information by electronically storing fingerprint evidence and providing an interface to register criminals.

The fingerprint identification process includes many variables such as image source, image orientation, image size, print alignment, brightness, background noise, ridge definition etc. The matching accuracy of the identification process is dependent on how each of these variables is accounted for. The neural network classification approach was implemented to account for some of these variables by building a model of images of different alignment, amount of background noise and different parts of the fingerprint. The implemented latent fingerprint identification algorithm with an accuracy of 64.44% for the dataset of 252 files would need to be improved to be commercially ready.

6.2 Future Work

This part of the chapter highlights some recommendations on areas where the project could be improved to develop a better system.

Optimising pure image processing algorithms

The image processing algorithms SIFT, Flann and Brute-Force work well to detect high-quality images of the same kind. A combination of biometric features, i.e. minutiae points and the SIFT features, could be collected and evaluated to generate a more robust comparison for the dataset collected.

Optimising Machine Learning algorithms

The machine learning algorithm implemented depends on the availability and the classification of both latent and scanned images. The algorithm’s accuracy could be improved by classifying minutiae features from the fingerprint images and matching an input fingerprint image’s minutiae points to the classes in the model. This approach could reduce the number of false-positive matches recorded by the system. Additionally, a step to classify the input fingerprint image as male or female can be implemented before the minutiae classification to reduce the number of recorded false positives further.

Accounting for more crime scene scenarios

While the project’s scope was limited to full fingerprints found on less noisy backgrounds, many fingerprints found at crime scenes are partial, overlapped with other prints and have noisier backgrounds. More work could be done in the pre-processing of partial, overlapped and noisy prints to extract a good number of features from them. The minutiae-based feature extraction algorithms can be developed to extract more features from a print other than the ridge endings and the ridge bifurcations. More work should be done on testing real latent prints similar to the ones obtained on crime scenes to improve the performance of the existing algorithms.

Construction of a device

To fully meet portability, security, and speed requirements, the project can be developed by designing a compact device. The device should be able to capture fingerprint images from a crime scene at acceptable quality. It will be able to do the processing at the edge, eliminating the latency associated with sending the fingerprint image and its metadata to the server for processing. It will also limit the number of security measures that need to be placed on the network and ensure better protection of sensitive government information.

References

[1] Ladapo, O. A., "Effective Investigations, A Pivot to Efficient Criminal Justice Administration: Challenges "African Journal of Criminology and Justice Studies, vol. 5, no 1,2, pp 79-94, 2011.

[2] J. Anthony, "NAF Commences Advance Forensic Crime Scene Investigation Course for Personnel in Lagos," *Herald Nigeria*, para. 1,2, Oct 14, 2020. [Online]. Available: https://www.heraldnigeria.com/2020/10/naf-commences-advance-forensic-crime.html. [Accessed Feb. 22, 2021].

[3] National Institute of Justice, *Forensic Sciences*, n.d. Accessed on Oct 5, 2020. [Online]. Available: https://nij.ojp.gov/topics/forensics

[4] "What is Forensics?" n.d. Accessed on Oct 2, 2020. [online]. Available: https://www.crimesceneinvestigatoredu.org/what-is-forensic-science/

[5] B. Fakiha, "Technology in Forensics Science," *Open Access Journal of Science and Technology*, vol. 7, no 1, pp 1-10, Jan. 2019. Accessed on Oct. 27, 2020. [Online]. Available doi:11131/2017/101258

[6] J. Van der Kleut, *Biometrics and biometric data: What is it, and is it secure?* n.d. Accessed on: Oct 5, 2020. [Online]. Available: https://us.norton.com/internetsecurity-iot-biometrics-how-do-they-work-are-they-safe.html

[7] "Common Biometrics Techniques Compared," n.d. Accessed on Oct 5, 2020. [online]. Available:https://www.recogtech.com/en/knowledge-base/5-common-biometric-techniques-compared'

[8] Webopedia Staff, *False Rejection and False Acceptance Rate*, n.d. Accessed on Oct 5, 2020. [Online]. Available: https://www.webopedia.com/definitions/false-rejection/

[9] S. Mayhew, *What is Fingerprint Identification?* n.d. Accessed on Oct 7, 2020. [Online]. Available: https://www.biometricupdate.com/201205/what-is-fingerprint-identification

[10] A.Saleh, A. Bahaa, A. Wahdan, "Fingerprint Recognition, Advanced Biometric Technologies," Girija Chetty and Jucheng Yang, IntechOpen, Aug 9, 2011. Available doi: 10.5772/23476.

[11] M. M. H. Ali, V. H. Mahale, P. Yannawar and A. T. Gaikwad, "Fingerprint Recognition for Person Identification and Verification Based on Minutiae Matching," 2016 IEEE 6th International Conference on Advanced Computing (IACC), Bhimavaram, 2016, pp. 332-339, doi: 10.1109/IACC.2016.69.

[12] National Forensic Science Technology Center, A Simple Guide to Fingerprint Analysis, 2013. Accessed on Oct 5, 2020. [Online]. Available: http://www.forensicsciencesimplified.org/prints/how.html#top

[13] U. U. Desphande, V. S. Malemath, M. S. Patil, V. S. Chaugule, "End-to-End Automated Latent Fingerprint Identification With Improved DCNN-FFT Enhancement," Frontiers in

Robotics and AI, vol.7, Nov 30, 2020. Accessed on Feb 23, 2021. [Online]. Available doi:10.3389/frobt.2020.594412

[14] B. O. Alijla, M. Saad and S. F. Issawi, "Neural network-based minutiae extraction for fingerprint verification system," 2017 8th International Conference on Information Technology (ICIT), Amman, 2017, pp. 435-441, doi: 10.1109/ICITECH.2017.8080039.

[15] D. Valdes-Ramirez et al., "A Review of Fingerprint Feature Representations and Their Applications for Latent Fingerprint Identification: Trends and Evaluation," in IEEE Access, vol. 7, pp. 48484-48499, 2019, doi: 10.1109/ACCESS.2019.2909497.

[16] S. Kaur, "Noise types and various removal techniques, "International Journal of Advanced Research in Electronics and Communication Engineering, vol. 4, no. 2, pp 226-230, Feb. 2015. Available doi: 10.1.1.683.6783

[17] Devnath, Liton & I. Raful, "Fingerprint Image Denoising by Various Filters for different Noise using Wavelet Transform," American International Journal of Research in Science, Technology, Engineering & Mathematics, vol. 13, no 1, pp 39-44, Feb. 2016. Accesed on Oct. 2020 [online]. Available doi: 10.13140/RG.2.1.2448.3448.

[18] R. Mathew, B. Thomas and J. J. Kizhakkethottam, "Review on latent fingerprint matching techniques," 2015 International Conference on Soft-Computing and Networks Security (ICSNS), Coimbatore, 2015, pp. 1-4, doi: 10.1109/ICSNS.2015.7292415.

[19] M. Kaur, J. Kaur and J. Kaur, "Survey of Contrast Enhancement Techniques based on Histogram Equalization", *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 7, p. 137, 2011. Available: 10.14569/ijacsa.2011.020721.

[20] R. Dass, P. S. Devi, "Image Segmentation Techniques," International Journal of Electronics & Communications Technology, vol. 3, no. 1, pp 66-70, Mar. 2012. Available: doi: 10.1.1.227.6638

[21] A. M. Bazen, "Fingerprint Identification - Feature Extraction, Matching, and Database Search," Formal Methods in System Design, Aug 19, 2002. Accessed on: Feb 23, 2021.[Online].Available:https://www.researchgate.net/publication/239851652_Fingerprint_Identification_-_Feature_Extraction_Matching_and_Database_Search

[22] Y. Li-Qiang and G. Ling, "Feature Extraction of Fingerprint Image Based on Minutiae Feature Points," 2012 International Conference on Computer Science and Service System, Nanjing, China, 2012, pp. 1737-1740, doi: 10.1109/CSSS.2012.434

[23] S. P. Sandip and P. H. Zope, "Selective review of fingerprint enhancement, classification and matching techniques," *2015 IEEE Bombay Section Symposium (IBSS)*, Mumbai, 2015, pp. 1-6, doi: 10.1109/IBSS.2015.7456656.

[24] U. U. Desphande, V. S. Malemath, M. S. Patil, V. S. Chaugule, "End-to-End Automated Latent Fingerprint Identification With Improved DCNN-FFT Enhancement," Frontiers in Robotics and AI, vol.7, Nov 30, 2020. Accessed on Feb 23, 2021. [Online]. Available doi:10.3389/frobt.2020.594412

[25] K. Han, Z. Wang and Z. Chen, “Fingerprint Image Enhancement Method based on Adaptive Median Filter,” *2018 24th Asia-Pacific Conference on Communications (APCC)*, Ningbo, China, 2018, pp. 40-44, doi: 10.1109/APCC.2018.8633498.

[26] A. Liban and S. M. S. Hilles, “Latent Fingerprint Enhancement Based On Directional Total Variation Model with Lost Minutiae Reconstruction,” *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, Shah Alam, 2018, pp. 1-5, doi: 10.1109/ICSCEE.2018.8538417.

[27] H. T Van, G. V. Vu, and T. H. Le. “Fingerprint enhancement for direct grey-scale minutiae extraction by combining MFRAT and Gabor filters.” *In Proceedings of the Seventh Symposium on Information and Communication Technology (SoICT ‘16). Association for Computing Machinery*, New York, NY, USA, 2016, pp 360–367. DOI: https://doi.org/10.1145/3011077.3011127

[28] S. Sindhu and B. Arunadevi, “Fingerprint Authentication Based on Adaptive Greedy Registration of Minutiae Pairs,” 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, 2018, pp. 1360-1364, doi: 10.1109/ICOEI.2018.8553975.

[29] Lin Hong, Yifei Wan and A. Jain, “Fingerprint image enhancement: algorithm and performance evaluation,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 777-789, Aug. 1998, doi: 10.1109/34.709565.

[30] A. Rao, A Taxonomy for Texture Description and Identification. NewYork, NY: Springer-Verlag, 1990

[31] Babatunde, I. G., Kayode, A. B., Charles, A. O., & Olatubosun, O. (2012). Fingerprint image enhancement: Segmentation to thinning.

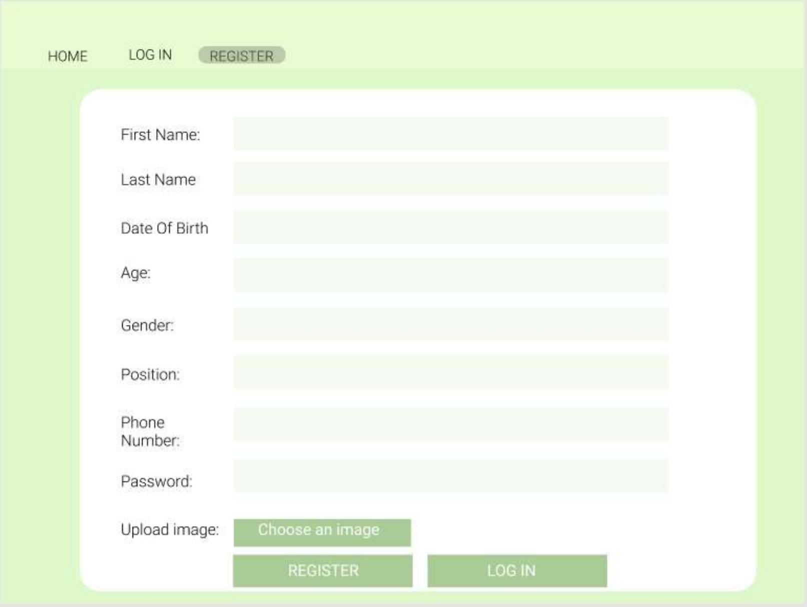
[32] “The Python SQL Toolkit and Object Relational Mapper.” SQLAlchemy. Retrieved from https://www.sqlalchemy.org/

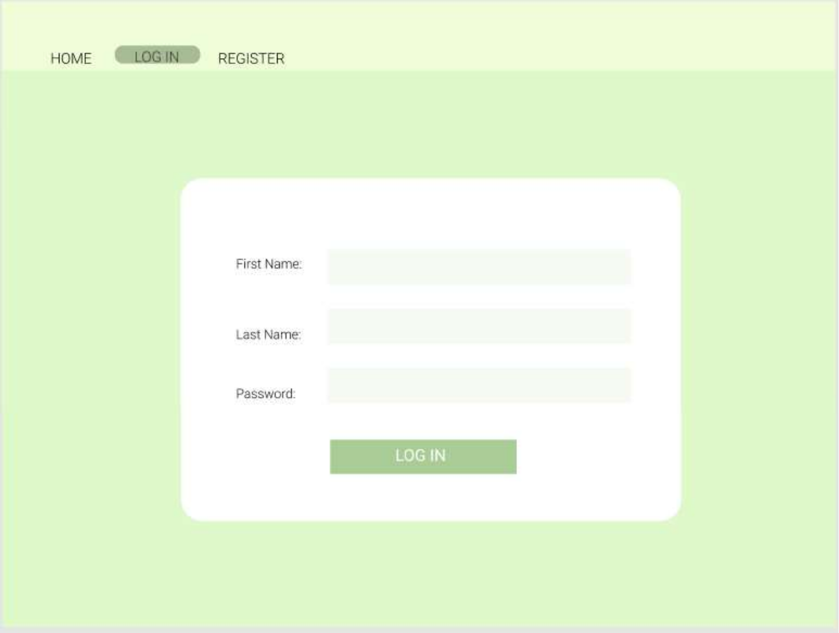
[33] Tensor Flow, Image Classification. Available https://www.tensorflow.org/tutorials/images/classification

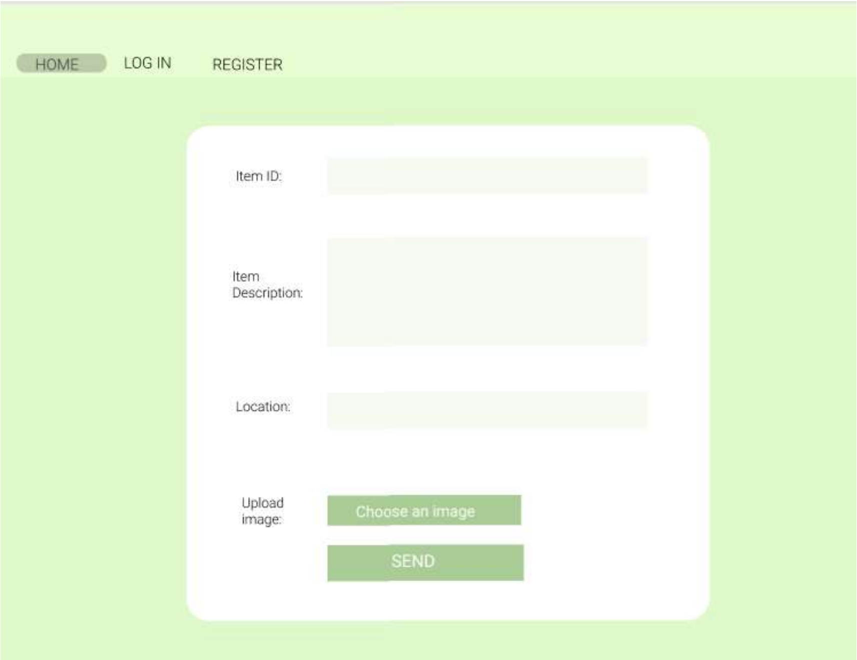
[34] A. Chandrasekaran and B. Thuraisingham, “Fingerprint Matching Algorithm Based on Tree Comparison using Ratios of Relational Distances,” The Second International Conference on Availability, Reliability and Security (ARES’07), 2007, pp. 273-280, doi: 10.1109/ARES.2007.90.

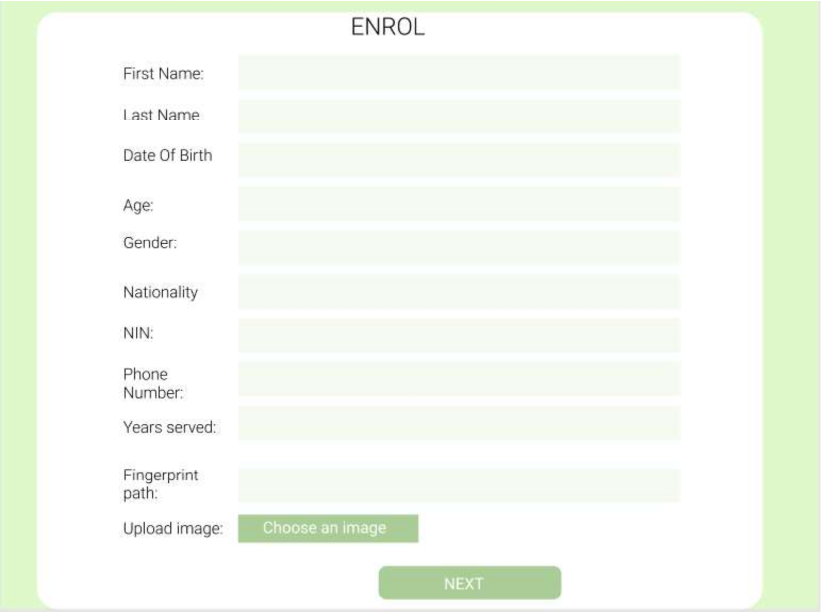
Appendices

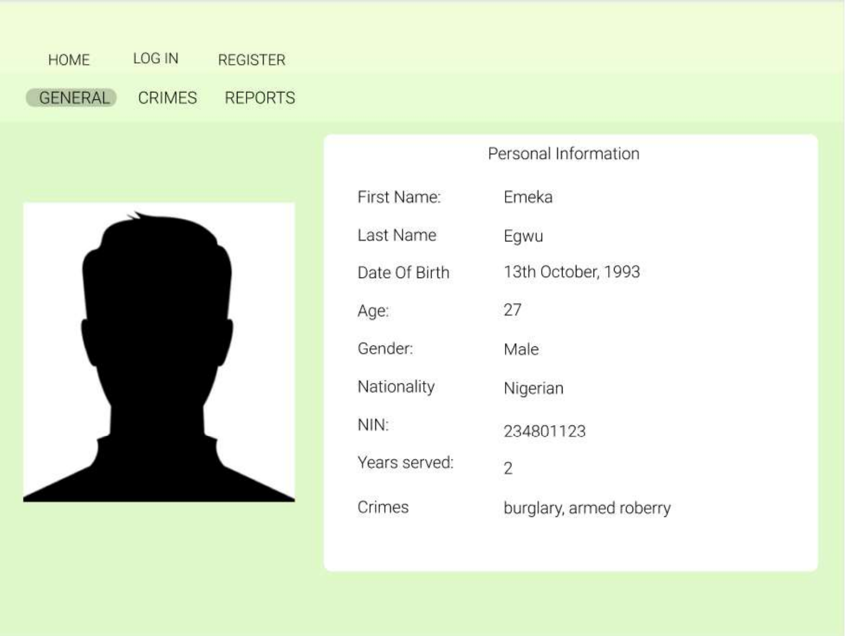
Appendix A-Web pages





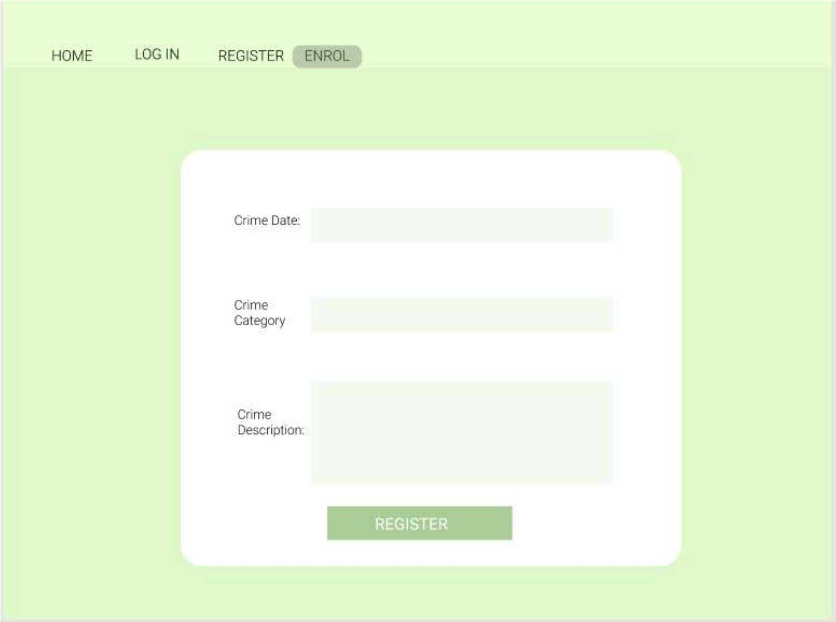

HOME
LOG IN
REGISTER
Item ID:
Item Description:
Location:
Upload image:
Choose an image
SEND


ENROL
First Name:
Last Name
Date Of Birth
Age:
Gender:
Nationality
NIN:
Phone Number:
Years served:
Fingerprint path:
Upload image:
Choose an image
NEXT



Personal Information

First Name:	Emeka
Last Name	Egwu
Date Of Birth	13th October, 1993
Age:	27
Gender:	Male
Nationality	Nigerian
NIN:	234801123
Years served:	2
Crimes	burglary, armed roberry



Crime Date:

Crime Category

Crime Description: