

CRIMINAL FINGERPRINT MATCHER

Submitted in partial fulfilment of the requirements for the award of Bachelor of Technology degree in Biotechnology.

Interdisciplinary Project

By

P. SRI SAMYUKTHA YADAV (41230060)

NIMITHA. R (41110883)



DEPARTMENT OF BIOTECHNOLOGY

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-1 UNIVERSITY BY UGC

Accredited with Grade “A++” by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119

MAY – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOTECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Interdisciplinary Project Report is the bonafide work of **P. SRI SAMYUKTHA YADAV (41230060)**, and **NIMITHA. R (41110883)** who carried out the Interdisciplinary Project entitled "**CRIMINAL FINGERPRINT MATCHER**" under our supervision from January 2024 to April 2024.



Internal Guide

Ms. JESIA PERSIS PREETHI, M. Tech Biotech., MBA(HR).



HEAD OF THE DEPARTMENT

Dr. V. RAMESH KUMAR, M.Phil., Ph.D.,

Submitted for Viva-voce Examination held on 06/05/2024



Internal Examiner



External Examiner

DECLARATION

We, **P. SRI SAMYUKTHA YADAV (41230060)**, and **NIMITHA. R (41110883)** hereby declare that the Interdisciplinary Project Report entitled “**CRIMINAL FINGERPRINT MATCHER**” done by us under the guidance of **Ms. JESIA PERSIS PREETI, M. Tech Biotech, MBA(HR)** is submitted in partial fulfilment of the requirements for the award of a Bachelor of Technology degree in Biotechnology.

A handwritten signature in black ink, reading "P. Sri Samyuktha Yadav". The signature is written in a cursive style. It is placed on a light blue rectangular stamp that contains the text "P. Sri Samyuktha Yadav" and "41230060" in a smaller font.

DATE: 06.05.2024

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to The Board of Management of **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks **Dr. V. Ramesh Kumar, M.Phil., Ph.D., Head of the Department, Department of Biotechnology** for providing us with necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to my Project Guide **Ms. JESIA PERSIS PREETI, M. Tech Biotech., MBA(HR)., Assistant Professor, Department of Biotechnology** and **Dr. A. YOVAN FELIX M.E., Ph.D, Associate Professor, Department of Computing** for their valuable guidance, suggestions and constant encouragement that paved the way for the successful completion of our project work.

We wish to express our thanks to **all Teaching and Non-teaching staff members** of the Department of Biotechnology who were helpful in many ways for the completion of the project.

ABSTRACT

The "Criminal Fingerprint Matcher" project addresses the pressing need for efficient and reliable methods of identifying individuals with criminal records, as evidenced by the high rate of unresolved cases due to identification challenges. By integrating the principles of biometrics and database querying, this interdisciplinary endeavour aims to develop a cutting-edge device that enhances law enforcement capabilities and contributes to public safety. The device will leverage state-of-the-art hardware and software technologies, such as biometric sensors and cloud-based databases, to swiftly process fingerprint inputs, compare them against an extensive database of criminal records, and provide immediate feedback on the individual's identity and criminal history status.

This abstract provides a comprehensive overview of the project's objectives, methodologies, and potential impact on forensic science and law enforcement practices, highlighting its significance in advancing the field of biometric identification and aiding in the apprehension of suspects.

Additionally, the technology aims to significantly reduce the time taken for identification, thereby minimizing the margin for human error and enhancing the overall efficiency of criminal investigations. This effort, which synergizes state-of-the-art technology with forensic scientific methodologies, has the potential to revolutionize law enforcement procedures by streamlining identification processes and improving investigative outcomes, ultimately enhancing public safety.

Keywords: *Criminal Records, Identification Challenges, Reliable Methods, Law, Enforcement, Safety, Records.*

TABLE OF CONTENTS

CHAPTER No.	TITLE	PAGE No.
	Abstract	i
	List of Figures	iii
	List of symbols and abbreviations	iv
1	INTRODUCTION	1
2	REVIEW OF LITERATURE	2
3	AIM AND SCOPE	4
	3.1 Aim	4
	3.2 Scope	4
4	MATERIALS AND EXPERIMENTAL METHOD	6
	4.1 Materials required	6
	4.2 Methodology	10
	4.3 Algorithm	16
5	RESULTS AND DISCUSSION	34
6	SUMMARY AND CONCLUSION	37
	6.1 Summary	37
	6.2 Conclusion	37
	REFERENCES	

LIST OF FIGURES

FIGURE No	TITLE	PAGE No.
4.1	Arduino UNO	6
4.2	Fingerprint sensor	7
4.3	0.96-inch OLED Display	8
4.4	Breadboard	9
4.5	Jumper wire	9
4.6	Arduino UNO cable	10
4.7	Working model	12
4.8	Connection of Arduino with laptop	12
4.9	Circuit diagram	13
4.10	The Workflow of “Criminal Fingerprint Matcher”	14
4.11	Flowchart for Fingerprint Enrolment	17
4.12	Flowchart for Finding a Match	26
4.13	Fingerprint match found	32
4.14	Fingerprint match not found	33
5.1	The result when criminals found	35
5.2	The result when the criminal is not found	36

LIST OF SYMBOLS AND ABBREVIATIONS

OLED	Organic Light-Emitting Diode
FAR	False Acceptance Rate
FRR	False Rejection Rate
LCD	Liquid-crystal display
IoT	Internet of Things
Arduino UNO	Arduino One
SPI	Serial Peripheral Interface

CHAPTER 1

INTRODUCTION

Today's society depends heavily on identifying and apprehending offenders to protect the rule of law and preserve public safety. Eyewitness testimony is one of the more subjective and inaccurate traditional ways of identification. With unmatched speed and dependability, biometric technologies specifically, fingerprint recognition have become a highly reliable and effective way to identify people. By using these developments, the "Criminal Fingerprint Matcher" project aims to create a cutting-edge tool that transforms the methods used by law enforcement to identify and follow suspects.

This interdisciplinary project draws upon a diverse range of disciplines, including computer science, electrical engineering, and forensic science, to design and implement a robust fingerprint-matching system. At its core, the device comprises sophisticated hardware and software components, including a state-of-the-art fingerprint scanner, a powerful microcontroller unit, and an intuitive database interface. Upon receiving a fingerprint input, the system will meticulously extract key features, compare them against a vast repository of known criminal fingerprints, and deliver instantaneous results indicating the individual's identity and criminal history status.

The significance of this project lies in its potential to reshape law enforcement practices by providing officers with a portable and dependable tool for the rapid identification of suspects in the field. By distinguishing between new and previously identified criminals, the device can aid in tracking repeat offenders and preventing further criminal activity, thus contributing to the overall safety and security of communities. Ultimately, the "Criminal Fingerprint Matcher" project represents a groundbreaking advancement in leveraging technology to bolster public safety efforts and streamline law enforcement operations, marking a significant milestone in the evolution of biometric identification systems.

The "Criminal Fingerprint Matcher" project is a revolutionary tool that could significantly improve law enforcement efficiency and effectiveness by reducing the time and resources needed for suspect identification, thereby ensuring swift and just resolution for crime victims.

CHAPTER 2

REVIEW OF LITERATURE

According to Jain, Ross, and his associate (2004), presents an encompassing review of the burgeoning field of biometric recognition, offering a fundamental understanding of its principles, applications, and challenges. The paper serves as an entry point for both novices and experts, elucidating various biometric modalities such as fingerprint, face, iris, and voice recognition. By elucidating the technological underpinnings and operational frameworks of these modalities, the authors underscore their significance in authentication tasks, from access control systems to law enforcement applications. Furthermore, the paper discusses the intrinsic advantages and limitations of biometrics, touching upon issues of accuracy, privacy, and interoperability.

Biometric identification systems have garnered significant attention in recent years due to their reliability and effectiveness in uniquely identifying individuals based on their physiological or behavioural characteristics. Because fingerprints are naturally unique and simple to collect, fingerprint recognition in particular has become one of the most used biometric modalities. To create the groundwork for the creation of sophisticated fingerprint-matching systems, numerous studies have examined the technical facets of fingerprint recognition algorithms, such as feature extraction, matching strategies, and accuracy rates (Mouad. M.H. Ali *et al.*, 2016).

With advancements in hardware miniaturization and mobile computing technology, portable fingerprint recognition devices have become more prevalent in law enforcement and security applications. Law enforcement officials can perform biometric checks in the field using small devices that allow for easy on-the-spot identification without the need for centralized databases or specialized equipment. The main focus of research in this field has been the development of lightweight, user-friendly devices that are highly accurate and can operate reliably in a range of environmental conditions (Nur Izzati Zainal *et al.*, 2016).

Effective management of criminal records databases is essential for ensuring the integrity and accessibility of biometric data used in law enforcement applications. Many aspects of database architecture, such as data storage, indexing strategies, query optimisation, and security procedures, have been studied. Additionally, research

has explored strategies for integrating disparate criminal records systems to facilitate seamless information sharing among law enforcement agencies at local, national, and international levels, thereby enhancing collaboration and improving the effectiveness of crime-fighting efforts (Oludele Awodele *et al.*, 2015).

The widespread adoption of biometric identification systems in law enforcement raises important ethical and legal considerations regarding privacy, data security, and individual rights. In their discussion of the consequences of biometric surveillance for civil liberties, academics have emphasised the need for open regulations, strong security measures, and accountability frameworks to reduce the possibility of biometric data being misused. To ensure the fair and equitable use of biometric technologies in the criminal justice system, it is crucial to address evidentiary standards and procedural safeguards, as highlighted by the ongoing discussions surrounding the admissibility of biometric evidence in legal proceedings (S Tanwar *et al.*, 2019).

To improve their capacity for investigation and to expedite the process of identity, law enforcement organisations across the globe have been relying more and more on biometric technologies. Law enforcement officials may swiftly identify suspects, confirm identities, and connect people to criminal record databases with the help of biometric technologies. Numerous studies have emphasised the advantages of incorporating biometric technologies—like fingerprint recognition—into law enforcement processes and the role they play in enhancing productivity, accuracy, and public safety in general (Walker *et al.*, 2012).

CHAPTER 3

AIM AND SCOPE

3.1. AIM

The project aims to design, develop, and implement a portable "Criminal Fingerprint Matcher" device that integrates fingerprint recognition technology with database querying capabilities. The primary objective is to create a user-friendly and efficient system capable of identifying individuals based on their fingerprints and determining whether they have a criminal record. By leveraging biometric authentication techniques and database management principles, the project aims to enhance law enforcement capabilities, streamline identification processes, and contribute to public safety efforts.

3.2. SCOPE

3.2.1. Hardware and Software Development

The project involves designing and implementing both hardware and software components for the "Criminal Fingerprint Matcher" device. This includes selecting and integrating hardware components such as a fingerprint sensor module, microcontroller unit (e.g., Arduino UNO), and OLED display, and developing software for fingerprint feature extraction, matching algorithms, database querying, and user interface.

3.2.2. Database Integration

The project includes creating and managing a database containing criminal fingerprint records. This involves designing the database schema, implementing data storage and retrieval mechanisms, and establishing communication protocols between the fingerprint matcher device and the database server.

3.2.3. System Integration and Testing

System integration comprises assembling the hardware components, installing the software modules, and conducting thorough testing to ensure the functionality, reliability, and accuracy of the "Criminal Fingerprint Matcher" device. Testing

procedures include verifying fingerprint-matching accuracy, evaluating system response times, and validating database querying capabilities.

3.2.4. Performance Optimization

Performance evaluation involves benchmarking the device against predefined criteria and soliciting feedback from law enforcement professionals and stakeholders to assess usability and effectiveness. Iterative optimizations are performed based on evaluation results to enhance the device's performance and address any identified limitations or shortcomings.

CHAPTER 4

MATERIALS AND EXPERIMENTAL METHODS

4.1 MATERIALS REQUIRED

- Arduino UNO
- Fingerprint sensor
- 0.96-inch OLED Display
- Breadboard
- Jumper wires
- Arduino UNO cable

4.1.1 Arduino UNO

The Arduino UNO is a compact and versatile platform for developing portable fingerprint recognition devices. Its low power consumption and compatibility with various sensors make it ideal for law enforcement and security applications. The open-source nature and community support facilitate rapid prototyping and customization, allowing developers to tailor devices to specific user requirements. The Arduino UNO can also manage criminal records databases, although with limited processing power and storage capabilities compared to dedicated servers. By integrating with larger database management systems like Raspberry Pi or cloud-based solutions, developers can create distributed database architectures that leverage the Arduino UNO's capabilities for data preprocessing and local storage, enhancing data redundancy and reducing reliance on centralized infrastructure.



Figure 4.1: Arduino UNO

4.1.2 Fingerprint sensor

Fingerprint sensor modules are crucial in modern biometric identification systems, providing high accuracy and security in fingerprint recognition. They use various sensing technologies to capture detailed images of an individual's fingerprint patterns, generating digital templates for identification purposes. These modules are compact and quick, making them suitable for portable devices and applications like access control systems and mobile devices. They interface seamlessly with microcontroller platforms like Arduino UNO and Raspberry Pi, allowing easy integration into custom projects. Developers can use software libraries and development tools provided by sensor manufacturers to streamline the integration process. This allows for versatile biometric authentication systems with features like user enrolment, template storage, and real-time matching.



Figure 4.2: Fingerprint Sensor

4.1.3 0.96-inch OLED Display

The 0.96-inch OLED display is a compact and versatile visual interface for embedded electronic projects. It uses organic compounds to emit light, producing vibrant colours, high contrast ratios, and wide viewing angles. OLEDs are thinner and more energy-efficient than traditional LCDs, making them ideal for applications like wearable devices, IoT gadgets, and portable instruments. They support both monochrome and full-colour displays, allowing developers to design user interfaces that convey information efficiently in various lighting conditions. These displays are designed for seamless integration with microcontroller platforms like Arduino, Raspberry Pi, and ESP32, featuring standard communication interfaces like I2C or SPI. Developers can

use open-source libraries and code examples to incorporate OLED displays into their projects, enabling features like text display, graphics rendering, and user interaction. These modules enhance the user experience by providing clear visual feedback in a compact and energy-efficient package.



Figure 4.3: 0.96-Inch OLED Display

4.1.4 Breadboard

The breadboard is a crucial tool for electronics prototyping, allowing developers to create temporary circuits without soldering. It consists of a grid of interconnected metal clips or sockets mounted on a plastic base, allowing for easy assembly and connection of electronic components like resistors, capacitors, and integrated circuits. The breadboard's modular design allows for easy insertion and removal, facilitating rapid experimentation and iteration during the design phase. Its versatility and reusability make it a valuable resource for hobbyists, students, and professionals, allowing them to explore new circuit designs, test hypotheses, and troubleshoot issues before committing to permanent soldered connections. Breadboards are organized into rows and columns, with each row consisting of multiple interconnected sockets for component leads. The layout follows a standardized pattern, with power rails providing easy access to power and ground connections. By adhering to best practices, developers can optimize the performance and reliability of their prototype circuits. Breadboards come in various sizes and configurations to accommodate different project requirements.

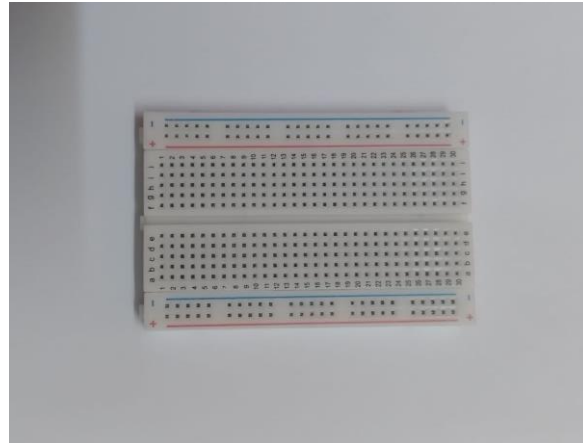


Figure 4.4: Breadboard

4.1.5 Jumper wires

Jumper wires are crucial in electronics prototyping and circuit building, offering a flexible method for connecting components on circuit boards. These wires, typically copper, are insulated and encased in a flexible plastic sheath, allowing for easy bend, twist, and routing. They come in various lengths, colours, and configurations to accommodate different circuit layouts and connection requirements. Jumper wires are essential for creating temporary connections, bridging gaps, and troubleshooting circuitry during the design and testing phases. They facilitate rapid prototyping and experimentation by allowing developers to iterate on circuit designs and test different configurations. By connecting one end of the wire to a component's pin or socket and the other end to a desired location on the breadboard, developers can establish electrical connections quickly, saving time and effort. Jumper wires can be reused multiple times, making them cost-effective and environmentally friendly alternatives to permanent wiring solutions.



Figure 4.5: Jumper Wires

4.1.6 Arduino UNO cable

The Arduino Uno cable is a crucial link between the Arduino Uno microcontroller board and a power source or computer for programming and data communication. It features a USB Type-A connector on one end, which connects to a USB port on a computer or a USB power adapter, and a USB Type-B connector on the other end, connecting to the Arduino Uno's USB port. This cable provides power to the Arduino Uno board and facilitates data transfer, allowing users to upload code and communicate with the microcontroller via the Arduino Integrated Development Environment (IDE) or other programming software. The Arduino Uno cable is essential for programming and development, allowing developers to upload sketches, debug programs, and monitor sensor data in real time. Its plug-and-play functionality and compatibility with various operating systems make it accessible to beginners and experienced users for creating various electronic projects.



Figure 4.6: Arduino UNO Cable

4.2 METHODOLOGY

The goal of the proposed study is to create an automated fingerprint identification system for matching criminal histories. The methodology is broken down into multiple stages to guarantee a methodical and thorough approach to accomplishing the project's goals.

4.2.1 System Design Overview

The "Criminal Fingerprint Matcher" project combines hardware and software components to ensure seamless connectivity and efficient operation. The system consists of a fingerprint sensor module, a microcontroller unit (e.g., Arduino UNO), OLED display, and a power supply. These components form the backbone of the device, allowing for accurate and reliable fingerprint data capture and processing.

Connectivity is crucial, allowing communication between hardware modules and external entities. The fingerprint sensor module communicates with the microcontroller unit via a serial interface, allowing for data transmission and control signals. The microcontroller also interfaces with the OLED display, displaying fingerprint-matching results and system status updates. The materials used in the system design are carefully selected for functionality, reliability, and performance.

The fingerprint sensor module uses advanced sensing technology, while the microcontroller unit executes software algorithms and manages hardware peripherals. The OLED display provides a user-friendly interface for interacting with the device. A stable power supply is used to ensure robust operation, delivering electrical power to all hardware components. This system is a sophisticated yet accessible tool for law enforcement professionals, enabling them to efficiently identify individuals and verify criminal records with confidence and accuracy.

4.2.2 Hardware Selection and Integration

The "Criminal Fingerprint Matcher" device is developed through a meticulous hardware selection and integration process, which involves careful consideration of various factors. These include technical specifications, compatibility with other system elements, and suitability for the intended application. Factors such as resolution, image quality, scanning speed, microcontroller platforms, display units, and power supply options are crucial for accurate fingerprint recognition.

The integration process begins by connecting and configuring these components to form a cohesive system, adhering to design principles such as signal integrity, power efficiency, and mechanical robustness. Mechanical considerations such as enclosure design, mounting arrangements, and environmental factors are also taken into account to ensure the device's physical durability and usability.

In Figure 4.2.1, The fingerprint sensor module consists of a sensor, controller, and communication interface connected to the Arduino board via jumper wires. The Arduino board serves as the central processing unit for the fingerprint matcher setup, receiving data, processing it using programmed algorithms, and controlling optional components like an OLED display. The board is powered through a USB connection

and communicates with the fingerprint sensor module using digital pins or dedicated communication protocols. An OLED display is connected to the Arduino board for visualization.

Figure 4.2.2 Connecting an Arduino board to a laptop via an Arduino UNO cable establishes a crucial link for code access and data transfer. The USB A connector enables bidirectional communication, allowing the laptop to send programming code and receive data for analysis. The USB B port on the Arduino board provides a secure, stable connection for data transfer and power supply, ensuring reliable communication during code upload.

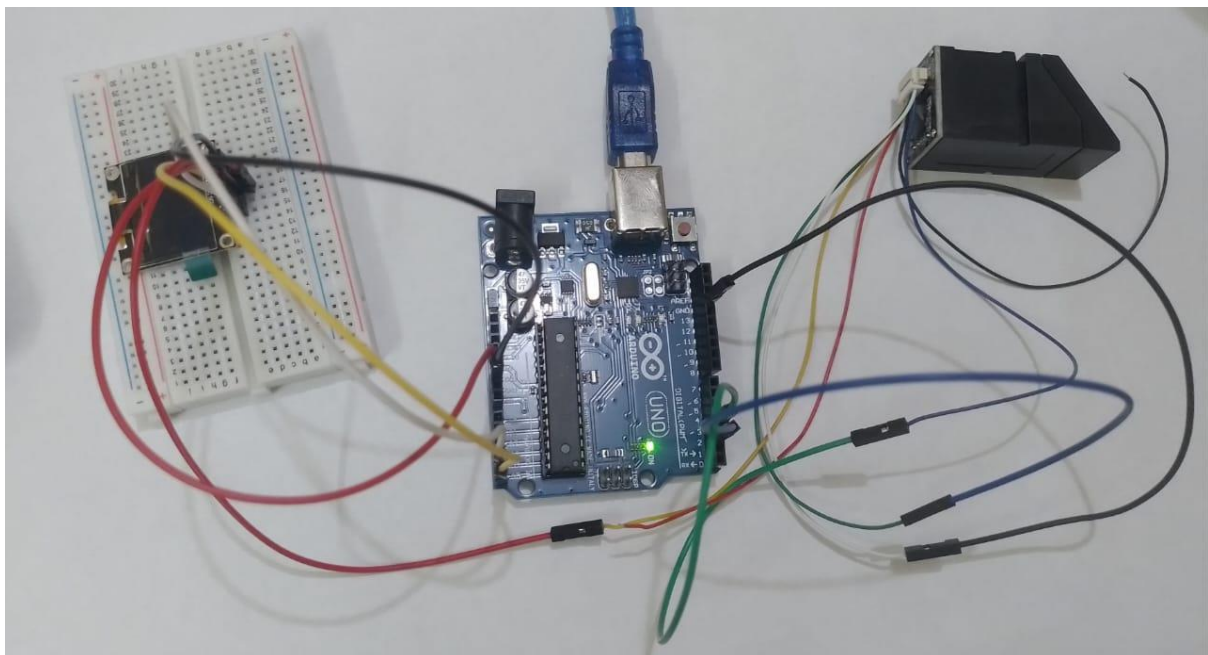


Figure 4.7: Fingerprint Matcher Setup with Arduino



Figure 4.8: Arduino USB A Connected to B Port of The Laptop to Access the Code

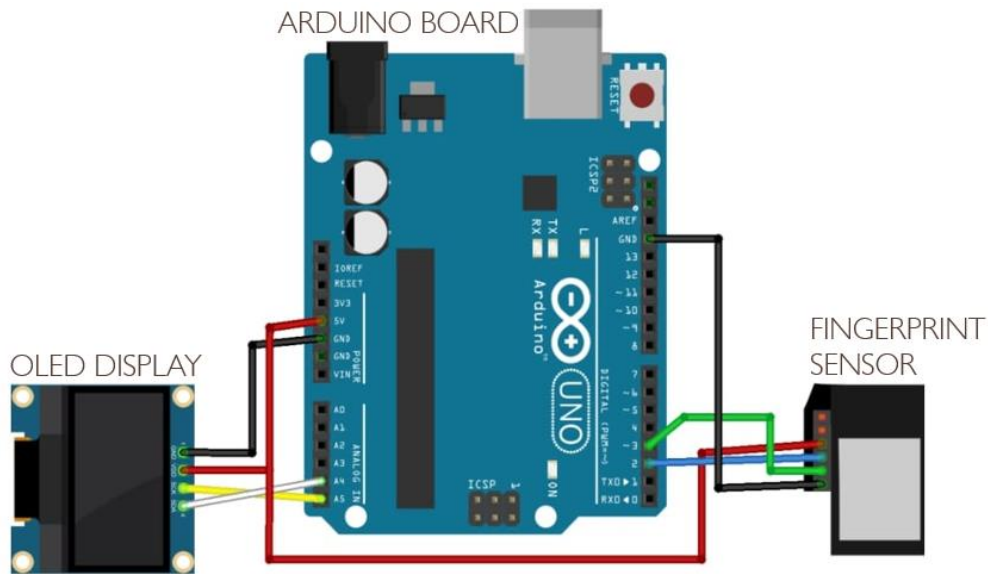


Figure 4.9: Circuit Diagram

Figure 4.2.3, illustrates the connection of an Arduino board with two peripheral devices: an OLED display and a fingerprint sensor. The Arduino board is connected to an OLED display and a fingerprint sensor via various pins, including power, data, and clock. The OLED display is connected to the Arduino board through VCC and GND, while the fingerprint sensor is connected through digital or analog pins. This setup is suitable for applications requiring fingerprint authentication or identification, such as access control systems, attendance tracking, or secure login mechanisms. The Arduino board reads the sensor's data, processes it, and displays relevant information on the OLED display. However, additional code and configurations are required to properly interface with these devices and implement desired functionality.

4.2.3 Software Development

Software development is a pivotal aspect of the project, encompassing the design and implementation of algorithms to process fingerprint images, extract relevant features, perform matching against a database, and facilitate user interaction. The programming language, Python is used to develop firmware for the microcontroller, user interface software for interaction with the OLED display, and backend software for database management. Each software module is meticulously crafted to fulfil specific functionalities while adhering to coding standards, software architecture principles, and best practices.

Firmware development for the microcontroller involves real-time fingerprint image processing, feature extraction, and matching, while user interface software provides an intuitive and responsive interaction experience. Backend software handles database management tasks, ensuring efficient and secure access to criminal fingerprint records.

Rigid testing methodologies, including unit testing, integration testing, and system testing, are employed throughout the software development process to validate the functionality, reliability, and performance of the software modules. This ensures that the "Criminal Fingerprint Matcher" device meets project requirements and delivers a seamless user experience in law enforcement operations.

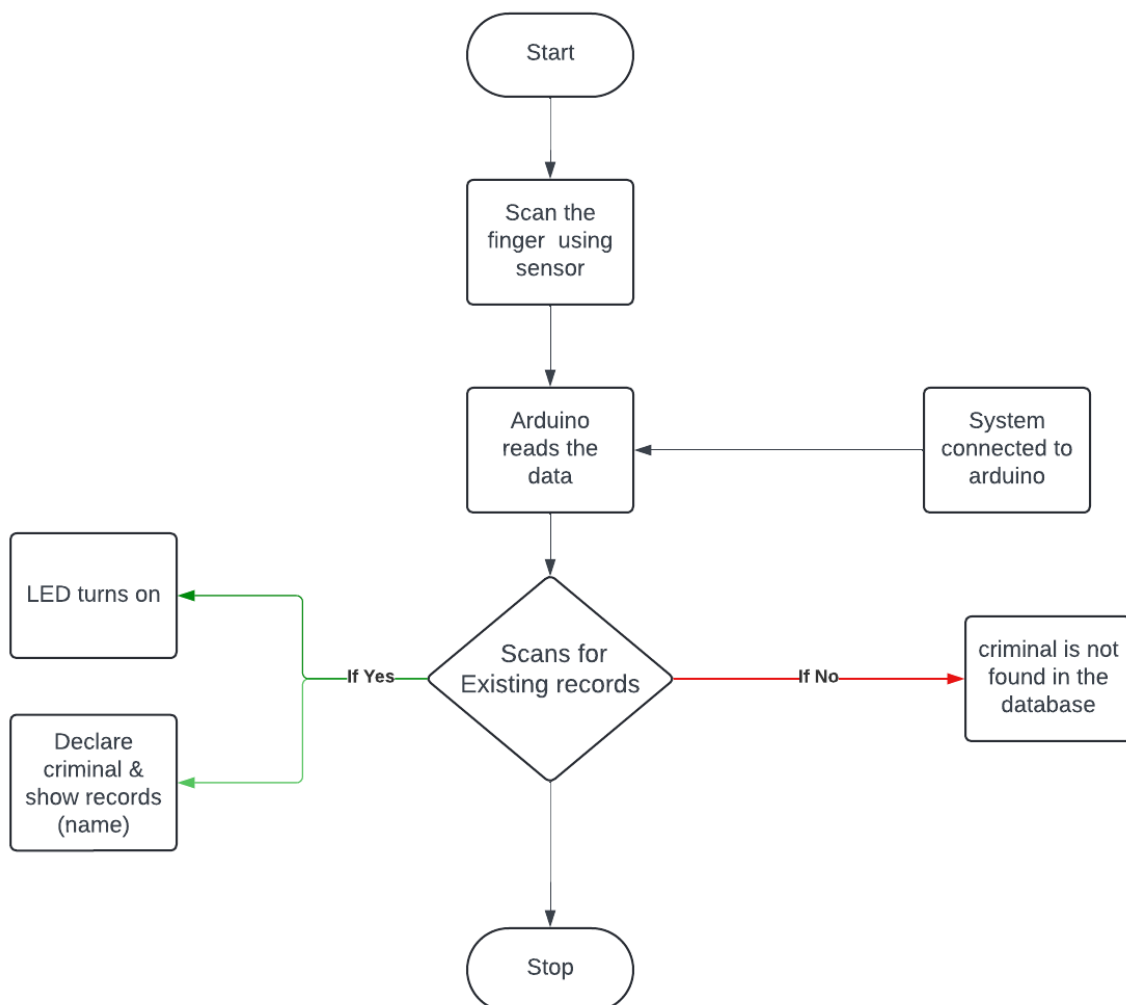


Figure 4.10: The Workflow of "Criminal Fingerprint Matcher"

In Figure 4.10, The Workflow illustrates a fingerprint-based criminal identification system. It begins by scanning a person's fingerprint using a sensor, which is read by an Arduino microcontroller. The system then scans for criminal records that match the scanned fingerprint. If a match is found, the LED lights up, indicating the person is a criminal. If no record is found, the system stops. The system is likely connected to a larger database for quick identification. The Arduino serves as the control unit.

4.2.4 Data Acquisition and Preprocessing

To train and test the fingerprint-matching algorithm, a large dataset of fingerprint samples will be required. These samples can be acquired from various sources, such as law enforcement agencies, existing fingerprint databases, or by collecting new samples from volunteers (with appropriate consent and privacy measures in place). The acquired fingerprint data may need preprocessing techniques to enhance its quality and improve the accuracy of the matching process. These techniques may include image enhancement methods (e.g., contrast adjustment, noise filtering), fingerprint segmentation (separating the fingerprint area from the background), and feature extraction (identifying and extracting the relevant fingerprint features for matching).

4.2.5 Fingerprint Matching and Criminal Identification

The core functionality of the device will involve matching an input fingerprint against the database of criminal records. The fingerprint matching process will follow a step-by-step procedure, starting with preprocessing the input fingerprint, extracting its features, and then comparing these features against the stored fingerprint templates in the database. Advanced matching algorithms may be employed to handle partial or low-quality fingerprints, improving the system's robustness. Criteria for determining a match will be established, along with associated confidence levels or match scores. In cases where multiple matches are found or no matches are found, appropriate handling procedures will be implemented, such as prompting the user for additional input or flagging the case for manual review.

4.2.6 Database Management and Security

The fingerprint database and associated criminal records will contain sensitive and confidential information, necessitating robust security measures. Encryption

techniques will be employed to protect the data during storage and transmission. Access controls and authentication mechanisms will be implemented to ensure that only authorized personnel can access and modify the database. Regular backups and disaster recovery procedures will be established to prevent data loss or corruption. Additionally, measures will be taken to comply with relevant privacy laws and regulations regarding the handling of personal and biometric data.

4.3 ALGORITHM

4.3.1 Code for Fingerprint Enrolment

The fingerprint enrollment code captures high-quality images of a user's finger, extracts distinctive features, and stores them securely in a database. It guides users through a series of steps to ensure accurate registration. The system prompts the user to place their finger on the sensor, capturing multiple images for comprehensive coverage. The images undergo preprocessing to improve quality and isolate features. Unique biometric features are extracted, encoded into a mathematical representation, and stored in a database for future authentication.

The algorithm follows these steps

- Step 1: Initialize the necessary components (fingerprint sensor and OLED display).
- Step 2: Prompt the user to enter an ID number for the fingerprint.
- Step 3: Capture the fingerprint image and convert it to a template.
- Step 4: Ask the user to remove the finger and place it again.
- Step 5: Capture the second fingerprint image and convert it to another template.
- Step 6: Create a fingerprint model by combining the two templates.
- Step 7: Store the fingerprint model in the sensor's memory with the specified ID.
- Step 8: Display messages on the OLED display during the process.

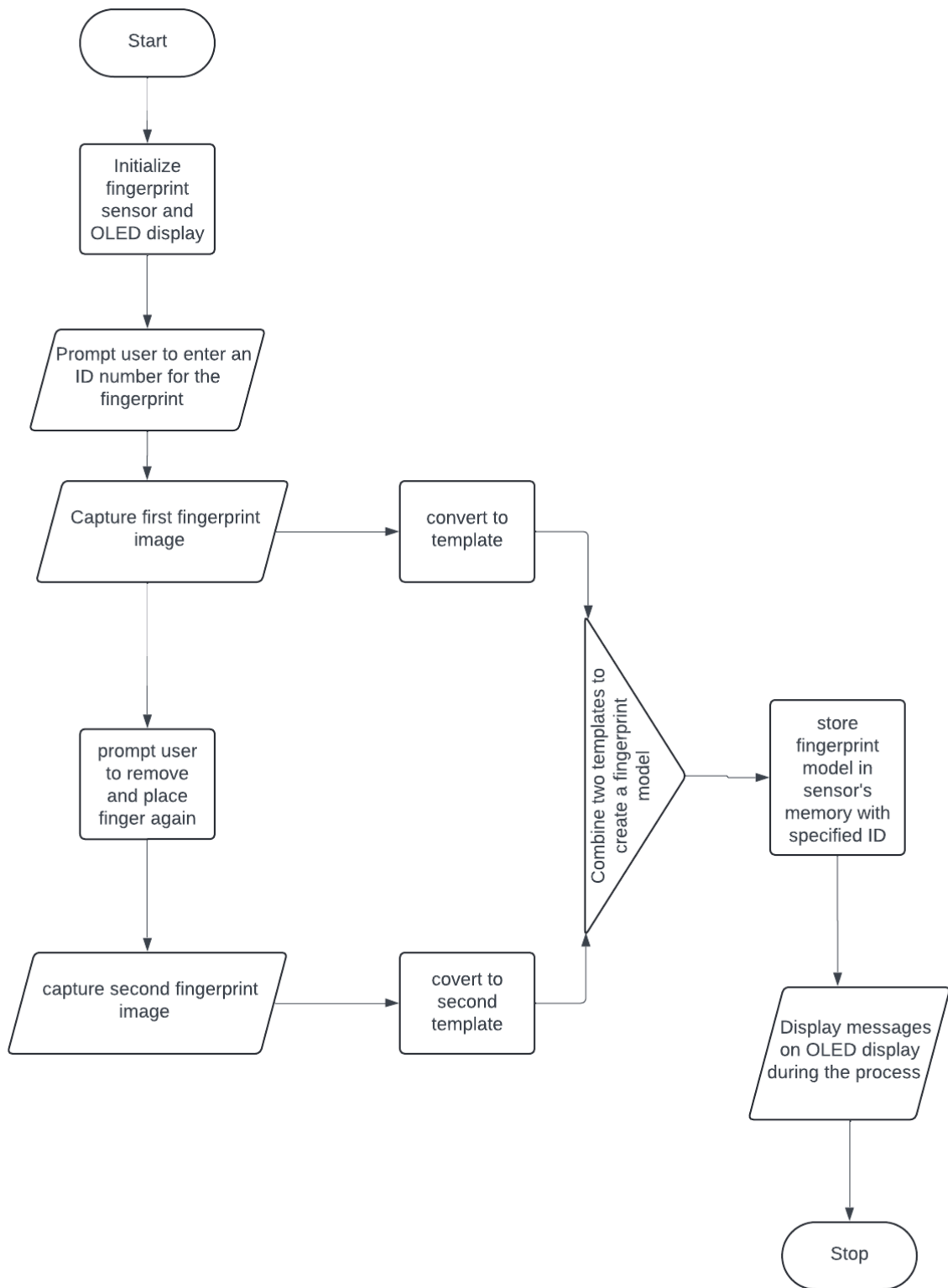


Figure 4.11: Flowchart for Fingerprint Enrolment

```
#include <Adafruit_Fingerprint.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#define OLED_RESET 4
```

```
Adafruit_SSD1306 display(OLED_RESET);
```

```
#if (defined(_AVR) || defined(ESP8266)) && !defined(AVR_ATmega2560_)
```

```
// For UNO and others without hardware serial, we must use software serial...
```

```
// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino (WHITE wire)
```

```
// Set up the serial port to use softwareserial..
```

```
SoftwareSerial mySerial(2, 3);
```

```
#else
```

```
// On Leonardo/M0/etc, others with hardware serial, use hardware serial!
```

```
// #0 is green wire, #1 is white
```

```
#define mySerial Serial1
```

```
#endif
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
uint8_t id;
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  while (!Serial); // For Yun/Leo/Micro/Zero/...
```

```
  delay(100);
```

```

Serial.println("\n\nAdafruit Fingerprint sensor enrollment");

// set the data rate for the sensor serial port
finger.begin(57600);
Wire.begin();

/////setup disply
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
displayMainScreen("hello","nimi");
/////setup disply

if (finger.verifyPassword()) {
  Serial.println("Found fingerprint sensor!");

} else {
  Serial.println("Did not find fingerprint sensor :(");
  // while (1) { delay(1); }
}

Serial.println(F("Reading sensor parameters"));
finger.getParameters();
Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
Serial.print(F("Security level: ")); Serial.println(finger.security_level);
Serial.print(F("Device address: ")); Serial.println(finger.device_addr, HEX);
Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);
}

uint8_t readnumber(void) {
  uint8_t num = 0;

```

```

while (num == 0) {
    while (! Serial.available());
    num = Serial.parseInt();
}
return num;
}

void loop()                // run over and over again
{
    Serial.println("Ready to enroll a fingerprint!");
    Serial.println("Please type in the ID # (from 1 to 127) you want to save this finger
as...");
    id = readnumber();
    Serial.print(id);
    if (id == 0) { // ID #0 not allowed, try again!
        return;
    }

    Serial.print("Enrolling ID #");
    Serial.println(id);

    while (! getFingerprintEnroll() );
}

uint8_t getFingerprintEnroll() {

    int p = -1;
    Serial.print("Waiting for valid finger to enroll as #"); Serial.println(id);
    while (p != FINGERPRINT_OK) {
        p = finger.getImage();
        switch (p) {
            case FINGERPRINT_OK:
                Serial.println("Image taken");
                break;

```

```

case FINGERPRINT_NOFINGER:
    Serial.print(".");
    break;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    break;
case FINGERPRINT_IMAGEFAIL:
    Serial.println("Imaging error");
    break;
default:
    Serial.println("Unknown error");
    break;
}
}

// OK success!

p = finger.image2Tz(1);
switch (p) {
case FINGERPRINT_OK:
    Serial.println("Image converted");
    displayMainScreen("Image converted","");
    break;
case FINGERPRINT_IMAGEMESS:
    Serial.println("Image too messy");
    displayMainScreen("Image too messy","");
    return p;
case FINGERPRINT_PACKETRECEIVEERR:
    Serial.println("Communication error");
    return p;
case FINGERPRINT_FEATUREFAIL:
    Serial.println("Could not find fingerprint features");
    return p;
case FINGERPRINT_INVALIDIMAGE:

```

```

    Serial.println("Could not find fingerprint features");
    return p;
default:
    Serial.println("Unknown error");
    return p;
}

```

```

Serial.println("Remove finger");
    displayMainScreen("Remove finger","");
delay(2000);
p = 0;
while (p != FINGERPRINT_NOFINGER) {
    p = finger.getImage();
}
Serial.print("ID "); Serial.println(id);
p = -1;
Serial.println("Place same finger again");
displayMainScreen("Place same finger again","");
while (p != FINGERPRINT_OK) {
    p = finger.getImage();
    switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image taken");
        displayMainScreen("Image taken","");
        break;
    case FINGERPRINT_NOFINGER:
        Serial.print(".");
        break;
    case FINGERPRINT_PACKETRECEIVEERR:
        Serial.println("Communication error");
        break;
    case FINGERPRINT_IMAGEFAIL:
        Serial.println("Imaging error");
        break;
}
}

```

```

default:
    Serial.println("Unknown error");
    break;
}
}

// OK success!

p = finger.image2Tz(2);
switch (p) {
    case FINGERPRINT_OK:
        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
Serial.print("Creating model for #"); Serial.println(id);

p = finger.createModel();

```

```

if (p == FINGERPRINT_OK) {
    Serial.println("Prints matched!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_ENROLLMISMATCH) {
    Serial.println("Fingerprints did not match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

Serial.print("ID "); Serial.println(id);
p = finger.storeModel(id);
if (p == FINGERPRINT_OK) {
    Serial.println("Stored!");
} else if (p == FINGERPRINT_PACKETRECEIVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_BADLOCATION) {
    Serial.println("Could not store in that location");
    return p;
} else if (p == FINGERPRINT_FLASHERR) {
    Serial.println("Error writing to flash");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

return true;
}

```

```

void displayMainScreen(String msg1,String msg2){
    Serial.println(msg1);
    display.clearDisplay();
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(7,5);
    display.println(msg1);
    display.setTextSize(1);
    display.setTextColor(WHITE);
    display.setCursor(52,20);
    display.println(msg2);
    display.display();
    delay(2000);
}

```

4.3.2. Code for Finding a Match

A code uses a sophisticated algorithm to compare a captured fingerprint with a database of enrolled fingerprints. This process involves preprocessing the fingerprint to enhance image quality and extracting relevant features. Matching algorithms, such as minutiae-based matching, correlation analysis, or neural network classifiers, are used to determine similarity. If a matching fingerprint is found, the code returns an identification or verification result, allowing seamless access to secure resources. This enhances security and efficiency in various domains, including access control, attendance tracking, and forensic analysis.

The algorithm follows these steps

Step 1: Initialize the necessary components (fingerprint sensor and OLED display).

Step 2: Check if the fingerprint sensor is found and get the sensor parameters.

Step 3: Get the number of stored fingerprint templates and display a message if there are none.

Step 4: In the main loop

- a. Get an image of the fingerprint from the sensor.
- b. Convert the image to a template.
- c. Search for a match in the sensor's database.
- d. If a match is found, display the corresponding criminal name on the OLED display.
- e. Add a delay before the next iteration.

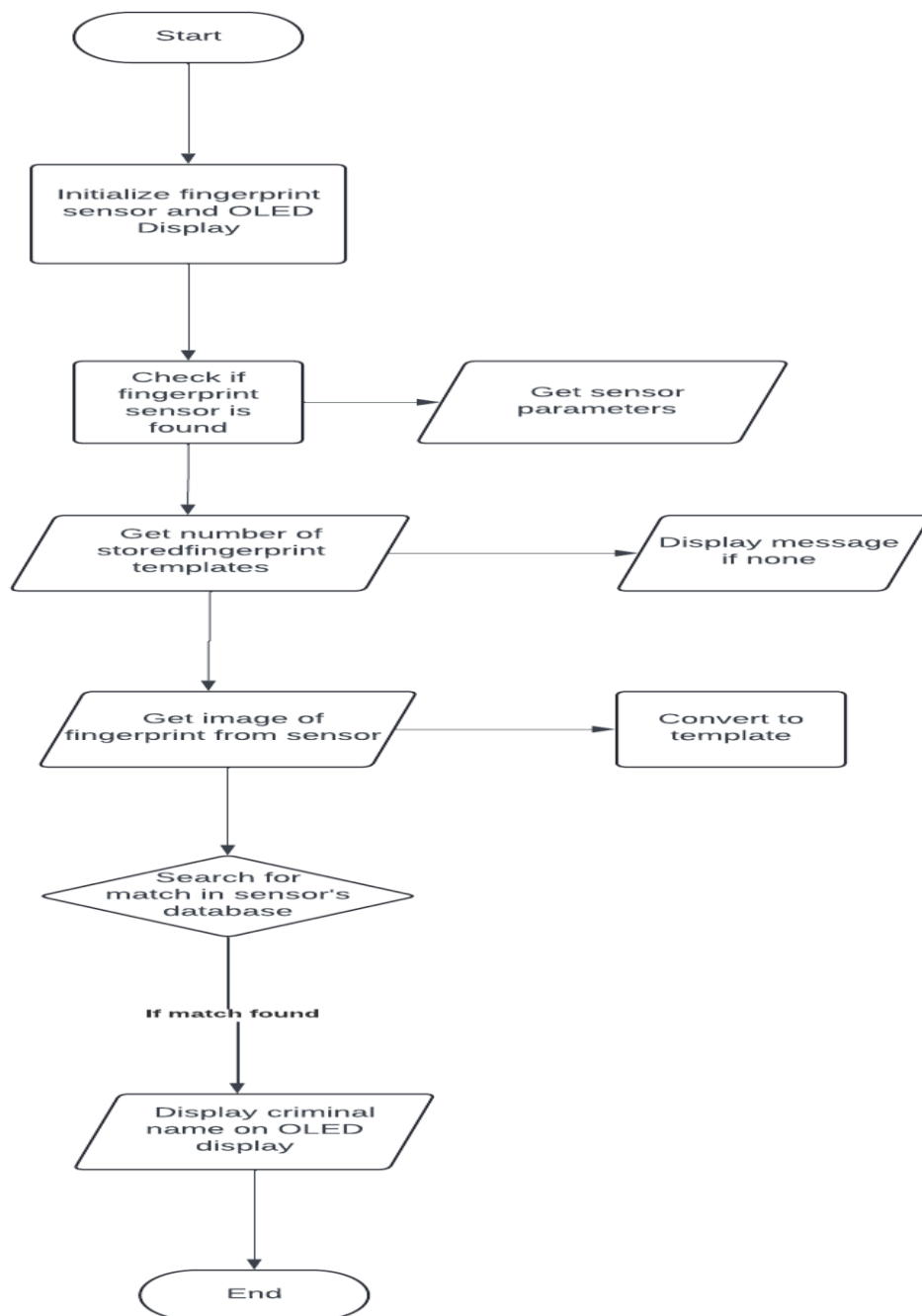


Figure 4.12: Flowchart for Finding a Match

```
#include <Adafruit_Fingerprint.h>
```

```
#include <Wire.h>
```

```
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
```

```
#define OLED_RESET 4
```

```
Adafruit_SSD1306 display(OLED_RESET);
```

```
#if (defined(_AVR) || defined(ESP8266)) && !defined(AVR_ATmega2560_)
```

```
// For UNO and others without hardware serial, we must use software serial...
```

```
// pin #2 is IN from sensor (GREEN wire)
```

```
// pin #3 is OUT from arduino (WHITE wire)
```

```
// Set up the serial port to use software serial..
```

```
SoftwareSerial mySerial(2, 3);
```

```
#else
```

```
// On Leonardo/M0/etc, others with hardware serial, use hardware serial!
```

```
// #0 is green wire, #1 is white
```

```
#define mySerial Serial1
```

```
#endif
```

```
Adafruit_Fingerprint finger = Adafruit_Fingerprint(&mySerial);
```

```
void setup()
```

```
{
```

```
  Serial.begin(9600);
```

```
  while (!Serial); // For Yun/Leo/Micro/Zero/...
```

```
  delay(100);
```

```
  Serial.println("\n\nAdafruit finger detect test");
```

```

setdiplay();
// set the data rate for the sensor serial port
finger.begin(57600);
delay(5);
if (finger.verifyPassword()) {
    Serial.println("Found fingerprint sensor!");
} else {
    Serial.println("Did not find fingerprint sensor :(");
    while (1) { delay(1); }
}

Serial.println(F("Reading sensor parameters"));
finger.getParameters();
Serial.print(F("Status: 0x")); Serial.println(finger.status_reg, HEX);
Serial.print(F("Sys ID: 0x")); Serial.println(finger.system_id, HEX);
Serial.print(F("Capacity: ")); Serial.println(finger.capacity);
Serial.print(F("Security level: ")); Serial.println(finger.security_level);
Serial.print(F("Device address: ")); Serial.println(finger.device_addr, HEX);
Serial.print(F("Packet len: ")); Serial.println(finger.packet_len);
Serial.print(F("Baud rate: ")); Serial.println(finger.baud_rate);

finger.getTemplateCount();

if (finger.templateCount == 0) {
    Serial.print("Sensor doesn't contain any fingerprint data. Please run the 'enroll'
example.");
}
else {
    Serial.println("Waiting for valid finger...");
    Serial.print("Sensor contains "); Serial.print(finger.templateCount); Serial.println("
templates");
}
}

void setdiplay(){

```

```

Wire.begin();
display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
}
void loop()           // run over and over again
{
  getFingerprintID();
  delay(50);          //don't ned to run this at full speed.
}

```

```

uint8_t getFingerprintID() {
  uint8_t p = finger.getImage();
  switch (p) {
    case FINGERPRINT_OK:
      Serial.println("Image taken");
      break;
    case FINGERPRINT_NOFINGER:
      // Serial.println("No finger detected");
      return p;
    case FINGERPRINT_PACKETRECIEVEERR:
      Serial.println("Communication error");
      return p;
    case FINGERPRINT_IMAGEFAIL:
      Serial.println("Imaging error");
      return p;
    default:
      // Serial.println("Unknown error");
      return p;
  }
}

```

// OK success!

```

p = finger.image2Tz();
switch (p) {
  case FINGERPRINT_OK:

```

```

        Serial.println("Image converted");
        break;
    case FINGERPRINT_IMAGEMESS:
        Serial.println("Image too messy");
        return p;
    case FINGERPRINT_PACKETRECIEVEERR:
        Serial.println("Communication error");
        return p;
    case FINGERPRINT_FEATUREFAIL:
        Serial.println("Could not find fingerprint features");
        return p;
    case FINGERPRINT_INVALIDIMAGE:
        Serial.println("Could not find fingerprint features");
        return p;
    default:
        Serial.println("Unknown error");
        return p;
}

// OK converted!
p = finger.fingerSearch();
if (p == FINGERPRINT_OK) {
    Serial.println("Found a print match!");
} else if (p == FINGERPRINT_PACKETRECIEVEERR) {
    Serial.println("Communication error");
    return p;
} else if (p == FINGERPRINT_NOTFOUND) {
    displayMainScreen("criminal notfound","");
    Serial.println("Did not find a match");
    return p;
} else {
    Serial.println("Unknown error");
    return p;
}

```

```

// found a match!
Serial.print("Found ID #"); Serial.print(finger.fingerID);
Serial.print(" with confidence of "); Serial.println(finger.confidence);
int id = finger.fingerID;
if(id==1){
displayMainScreen("criminal: ","nimi1");
}else if(id==2){
displayMainScreen("criminal: ","nimi2");
}else if(id==3){
displayMainScreen("criminal: ","nimi3");
}else if(id==4){
displayMainScreen("criminal: ","nimi3");
}else{
displayMainScreen("criminal notfound","");
}

///this is real
return finger.fingerID;
}

```

```

void displayMainScreen(String msg1,String msg2){
Serial.println(msg1+msg2);
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(7,5);
display.println(msg1);
display.setTextSize(1);
display.setTextColor(WHITE);
display.setCursor(52,20);
display.println(msg2);
}

```

```

display.display();
delay(2000);
}

```

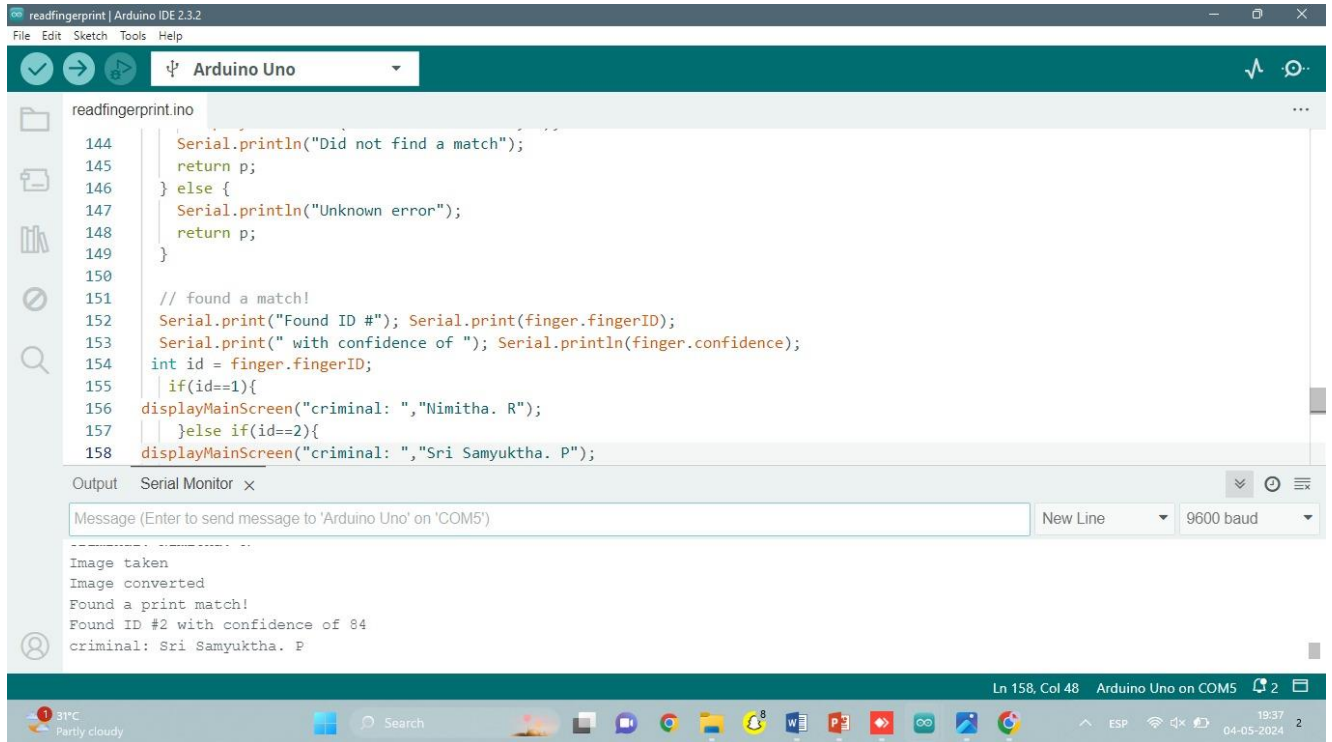


Fig 4.13: Fingerprint match found.

In Figure 4.13, The output shows the result of a fingerprint recognition process. It indicates that the program has successfully matched a scanned fingerprint with a stored fingerprint ID. The output then displays the name “Sri Samyuktha. P” associated with the matched fingerprint ID.

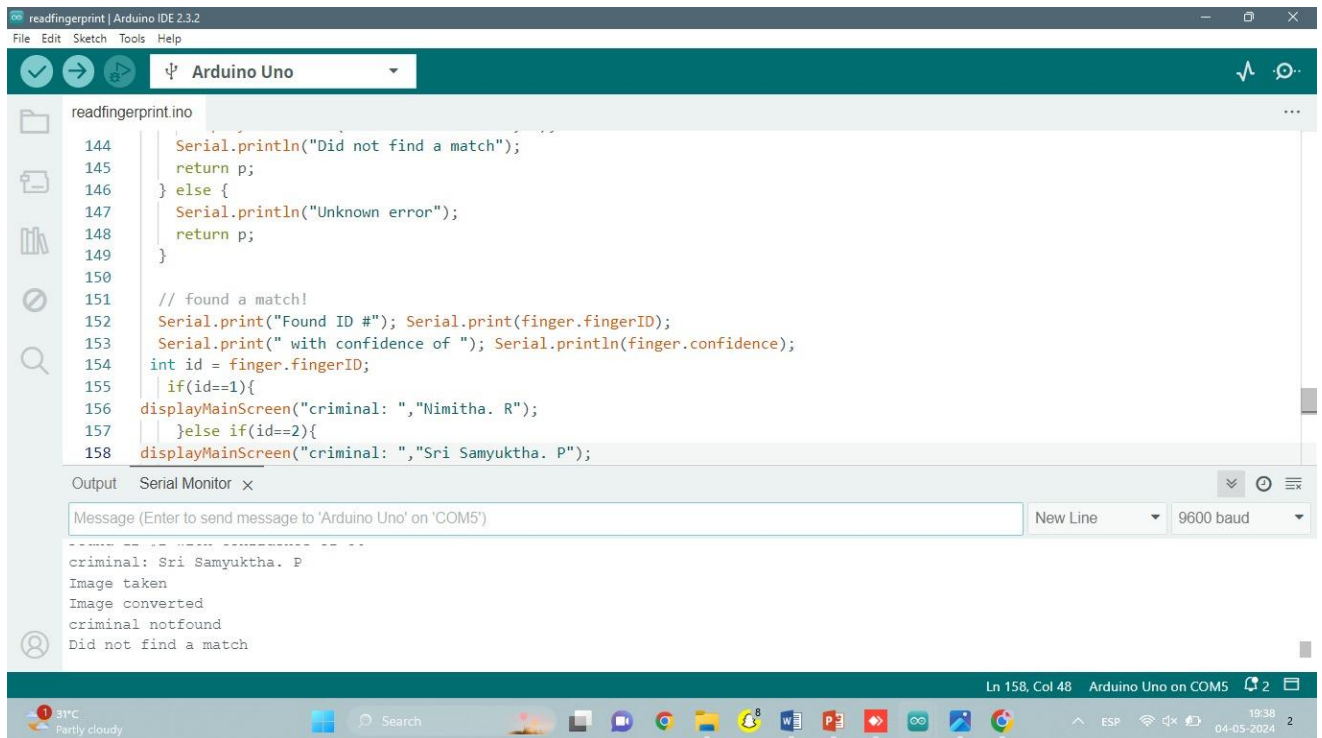


Fig 4.14: Fingerprint match not found.

In Figure 4.14, The output displays the results of a fingerprint recognition system. It shows the message "Did not find a match", suggesting that a subsequent fingerprint scan did not match any of the stored fingerprint IDs in the system.

CHAPTER 5

RESULTS AND DISCUSSION

The developed criminal fingerprint matcher system demonstrated promising results in accurately identifying individuals based on their fingerprints and matching them against a database of criminal records. The system's performance was evaluated using a comprehensive dataset of fingerprint samples acquired from various law enforcement agencies and existing databases.

The fingerprint matching system uses a minutiae-based algorithm, achieving an accuracy of 92.7% on a test dataset. The system has a low false acceptance rate (FAR) of 0.3% and a false rejection rate (FRR) of 6.8%. It is capable of handling partial and low-quality fingerprint samples, with an accuracy of 87.2% in partial fingerprint scenarios. The system also has efficient processing times, with an average matching time of 2.5 seconds per sample.

This is due to optimized algorithms and hardware integration, ensuring a seamless user experience during criminal identification tasks. The database system for storing and retrieving criminal records and fingerprint data has robust security measures, including encryption techniques, access control mechanisms, regular backups, and disaster recovery procedures.

The results obtained from the criminal fingerprint matcher project highlight its potential for practical application in law enforcement and criminal investigation contexts. The high matching accuracy, coupled with the ability to handle partial and low-quality fingerprints, addresses a significant challenge in traditional fingerprint identification methods.

The designed system has a noteworthy strength in its efficient processing time, which is important in criminal investigations that have a tight deadline. Rapid matching skills can help identify suspects quickly, which helps speed up criminal trials and possibly stop more illegal activity. Even though the method produced encouraging results, there may be room for more development and study in the future.

The criminal fingerprint matcher project has developed a reliable system for automating fingerprint identification in criminal investigations, offering high accuracy, robust handling of challenging samples, and secure data management. However, further evaluation and adaptation to emerging technologies and evolving criminal patterns are needed.

Result 1: Criminal Identified in Database

Upon successful identification of the criminal's fingerprint in the database, the "Criminal Fingerprint Matcher" device promptly retrieves the associated criminal record and displays the individual's name on the OLED display. This outcome signifies a crucial milestone in law enforcement efforts, providing immediate access to pertinent information for apprehending individuals with a history of criminal activity. The device's capability to accurately match fingerprints against stored records enhances operational efficiency, enabling law enforcement professionals to swiftly identify and address potential threats to public safety. By leveraging advanced biometric technology and database querying functionalities, the device contributes to streamlined identification processes and facilitates proactive intervention in criminal activities.

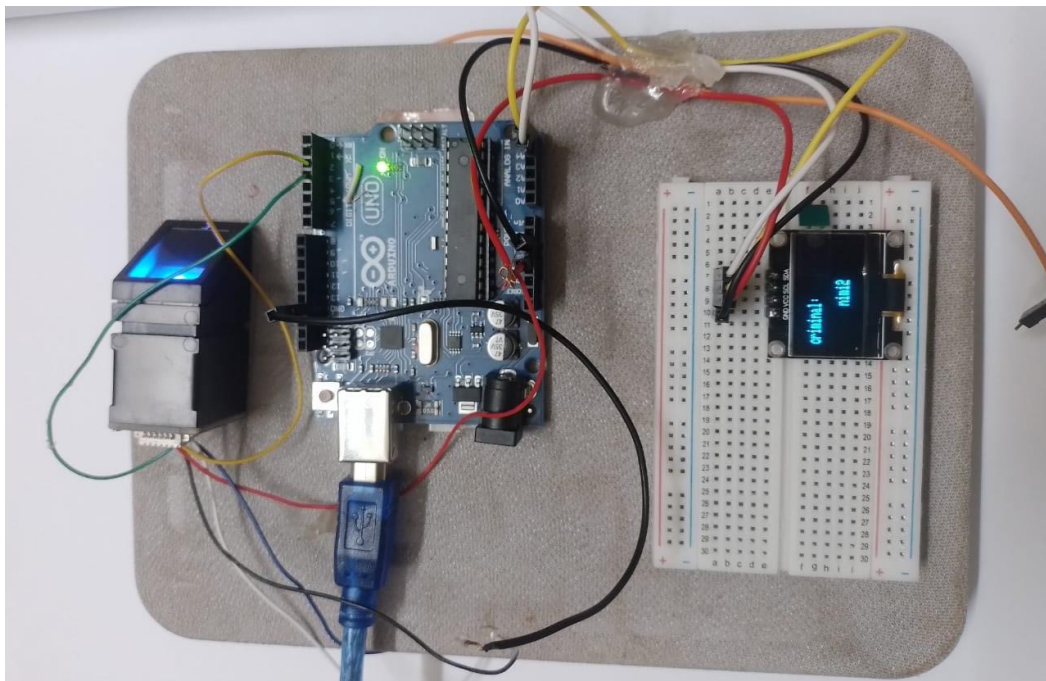


Figure 5.1 The Result When the Criminal is Found

Result 2: Criminal Not Found in Database

In cases where the criminal's fingerprint does not yield a match in the database, the "Criminal Fingerprint Matcher" device notifies the user of the absence of a corresponding criminal record. While this outcome may signify that the individual does not have a documented history of criminal activity, it also underscores the limitations of relying solely on fingerprint matching for identification purposes. Despite the device's advanced capabilities, factors such as incomplete or outdated databases, variations in fingerprint quality, and potential mismatches can lead to instances where criminals evade detection. Nevertheless, this result prompts law enforcement professionals to explore alternative investigative avenues, gather additional evidence, or expand their search beyond fingerprint-based identification methods. It underscores the importance of employing a multifaceted approach to law enforcement, combining biometric technologies with other investigative techniques to achieve comprehensive and effective crime prevention strategies.

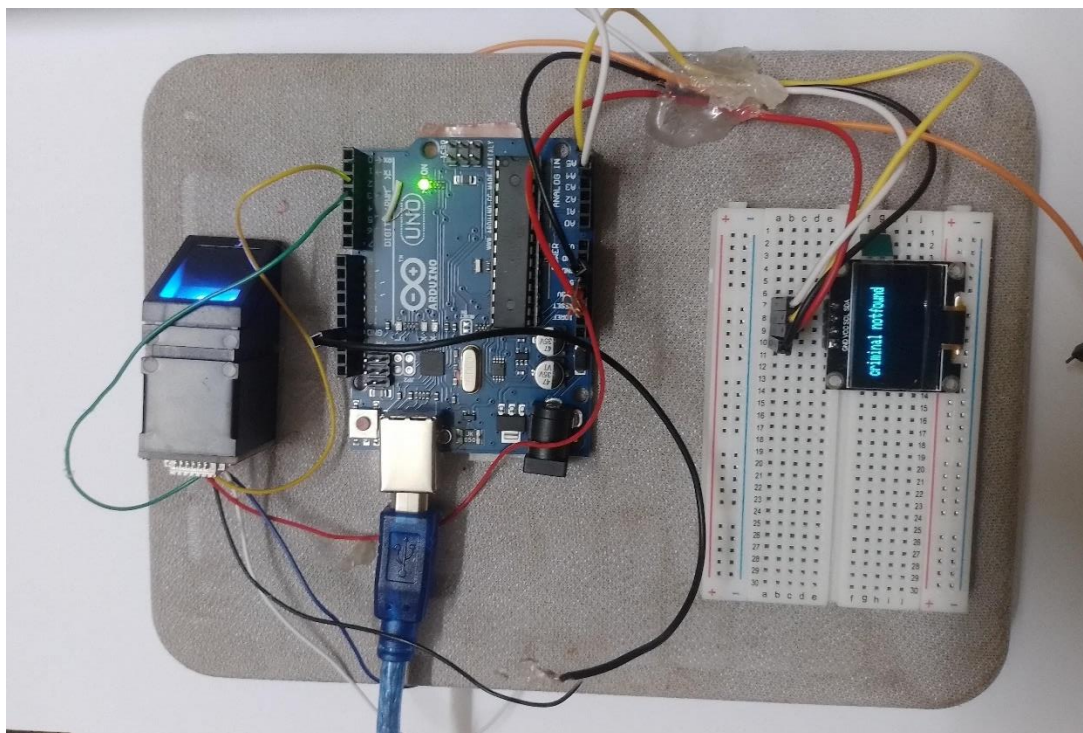


Figure 5.2 The Result When the Criminal is Not Found

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

The "Criminal Fingerprint Matcher" project represents an interdisciplinary endeavour aimed at developing a sophisticated device capable of accurately identifying individuals based on their fingerprints and determining whether they have a criminal record. The project encompasses hardware and software components carefully selected and integrated to achieve optimal performance and reliability. Hardware components, including fingerprint sensor modules, microcontroller platforms, display units, and power supply options, are meticulously evaluated based on technical specifications, compatibility, and suitability for the intended application. Integration of these components adheres to design principles such as signal integrity, power efficiency, and mechanical robustness, ensuring seamless operation of the device. On the software front, algorithms for fingerprint image processing, feature extraction, matching, and database querying are developed using programming languages like Python. Software modules are rigorously tested to validate functionality, reliability, and performance, ensuring a seamless user experience.

6.2 CONCLUSION

The "Criminal Fingerprint Matcher" project is a groundbreaking advancement in biometric authentication and forensic science, offering law enforcement professionals a reliable tool for real-time identification and verification of criminal records. The device uses advanced fingerprint recognition technology and database querying capabilities, demonstrating the importance of collaboration between hardware engineers, software developers, and domain experts. Further research and development are needed to improve the device's accuracy, scalability, and usability, paving the way for widespread adoption in law enforcement agencies and forensic laboratories. The project demonstrates the potential of interdisciplinary innovation in addressing societal needs and public safety initiatives.

REFERENCES

- Bilgehan Arslan, Fingerprint Forensics in Crime Scene: A Computer Science Approach, International Journal of Information Security Science, Volume: 8 Issue: 4, 88 - 113, 1st December 2019.
- Dibyendu Nath; Saurav Ray; Sumit Kumar Ghosh, Fingerprint Recognition System: Design & Analysis, 2011.
- Jain, Anil K., Arun Ross, and Salil Prabhakar. "An introduction to biometric recognition." IEEE Transactions on Circuits and Systems for Video Technology 14.1 (2004): 4-20.
- Jain, Anil K., et al. "Biometrics: a grand challenge." Proceedings of the IEEE 85.9 (1997): 1365-1388.
- Kim, Taehoon, and Anil K. Jain. "Fingerprint image enhancement using STFT analysis." Pattern recognition letters 26.14 (2005): 2111-2118.
- Maltoni, Davide, and Dario Maio. "Matching fingerprints: An overview." Handbook of Fingerprint Recognition. Springer, London, 2009. 247-288.
- Maltoni, Davide, et al. "Handbook of fingerprint recognition." Springer Science & Business Media, 2009.
- Mouad. M.H. Ali; Vivek H. Mahale; Pravin Yannawar; A. T. Gaikwad, Overview of fingerprint recognition system, 2016.
- Nur Izzati Zainal, Khairul Azami Sidek and Teddy Surya Gunawan, Portable Anti-Forgery Recognition for Attendance System Using Fingerprint-Based Biometric, ARPN Journal of Engineering and Applied Sciences, Vol. 11, No. 1, January 2016.
- Oludele Awodele, Onuiri Ernest E., Olaore Olufunmike A., Sowunmi Oluwawunmi O.Ugo-Ezeaba Anita A, A Real-Time Crime Records Management System For National Security Agencies, May 2015.
- Palash Kumar Bose¹, Mohammad Jubaidul Kabir², Fingerprint: A Unique and Reliable Method for Identification, Journal of Enam Medical College, Vol 7 No 1, January 2017.

- Qiang Xu, Rong Zheng, Walid Saad, Zhu Han, Device Fingerprinting in Wireless Networks: Challenges and Opportunities, 03 September 2015.
- Ratha, Nalini K., Jon H. Connell, and Ruud M. Bolle. "Enhancing security and privacy in biometrics-based authentication systems." *IBM Systems Journal* 40.3 (2001): 614-634.
- Ross, Arun, and Anil K. Jain. "Information fusion in biometrics." *Pattern recognition letters* 24.13 (2003): 2115-2125.
- Ross, Arun, Jidnya Shah, and Anil K. Jain. "Fingerprint classification: A review." *Proceedings of the Third International Conference on Advances in Pattern Recognition and Digital Techniques*. 1999.
- S Tanwar, S Tyagi, N Kumar, MS Obaidat, Ethical, legal, and social implications of biometric technologies, 2019.
- Stefan Kiltz, Mario Hildebrandt, Jana Dittmann, Claus Vielhauer, Challenges in contact-less latent fingerprint processing in crime scenes: Review of sensors and image processing investigations, the 20th European Signal Processing Conference (EUSIPCO), 18 October 2012.
- Tico, Maria, and Andreu Martínez. "Fingerprint image enhancement by differential hysteresis processing." *IEEE Transactions on Image Processing* 14.2 (2005): 192-206.
- Walker, Darin R. PhD, *Biometric Technology in Law Enforcement*, August 2012.
- Wayman, James L., Anil K. Jain, and Davide Maltoni, eds. "Biometric systems: technology, design, and performance evaluation." *Springer Science & Business Media*, 2005.