Indian Institute of Technology Kanpur

PIL-logo.png

SURGE INTERNSHIP REPORT

System And Method for Fingerprint Anti-Spoofing

Student Name	Student ID
Naina	2430527

Guide:

Dr. Tushar Sandhan

Submission Date: 13/07/2024

1 Abstract

This report details the deployment of a machine learning model on a Raspberry Pi to create a fingerprint recognition system with anti-spoofing capabilities, specifically to detect the liveness of fingerprints. The system differentiates between live and fake fingerprints using the R307 fingerprint sensor and libraries such as Adafruit, PySerial, pyfingerprint, TensorFlow, and PyTorch. Throughout the project, we faced several challenges, which are discussed in this report, along with how we overcame them to achieve a reliable and robust system. The report provides a comprehensive overview of the methods and techniques used to ensure the system's robustness and accuracy.

2 Introduction

Biometric authentication systems, including fingerprints, facial recognition, and iris scanning, are increasingly adopted across diverse applications for their convenience and security benefits. Despite their advantages, these systems are vulnerable to spoofing attacks where attackers use fake biometric data, such as replicated fingerprints or high-resolution photos, to gain unauthorized access. Successful spoofing can lead to significant risks such as financial fraud and identity theft, highlighting the critical need for enhanced security measures in biometric authentication technologies.

This project aimed to make fingerprint security better. We created a new system that uses hardware and advanced computer programs to tell if a fingerprint is from a real person or a fake one. This "liveness detection" is key to stopping fake fingerprints from fooling the system. By improving this detection, our goal is to make it much harder for attackers to trick fingerprint-based security systems.

This report covers our project in detail, including its goals, how we approached it, and what we achieved. We focused on designing hardware and using machine learning to detect if fingerprints are real or fake. We faced challenges along the way. The report also describes our prototype, what it can do, and how it could be used in real life. In conclusion, it discusses why our work is important and how it could make biometric security systems safer.

3 International and National status

3.1 Global Biometric Technology Trends

- Biometric authentication systems, including fingerprints, facial recognition, and iris scanning, are experiencing rapid global adoption due to their precision, accessibility, and decreasing costs.
- Advancements in sensor technologies and software algorithms continue to improve biometric accuracy and security worldwide.
- The international status on anti-spoofing fingerprint technology showcases significant advancements, driven by a growing need for secure biometric authentication systems. Here's an more detailed overview:
  - The LivDet (Liveness Detection) competitions, started in 2009, have been crucial in pushing the boundaries of fingerprint liveness detection.

These events have progressively challenged participants to develop systems that can detect increasingly realistic fake fingerprints made from materials like gelatin, latex, ecoflex, and silicone. Each year, the competition has introduced new hurdles, such as using unknown materials and involving operators with different skill levels to better mimic real-world conditions. This has ensured that the algorithms developed are not only effective but also adaptable to a wide range of spoofing techniques and scenarios(ar5iv)(SpringerLink).

3.2 Biometric Deployment in India

- India's digital transformation, notably through initiatives like Aadhaar, has led to extensive biometric data integration covering over 1.3 billion citizens.
- Biometrics are integral to government services, financial transactions, and public infrastructure, necessitating robust security measures against emerging threats like spoofing attacks.

Recent incidents highlight the severity of these vulnerabilities. For example, reports have surfaced of stolen fingerprints being used to empty bank accounts linked to the Aadhaar system in India . In another case, fraudsters have cloned fingerprints to create fake shell entities, raising concerns about the safety of Aadhaar-linked biometric data . These cases underscore the critical need for effective liveness detection to prevent such fraudulent activities.

Figure 1: some examples

In response to these challenges, ongoing research aims to integrate advanced materials and deep learning techniques to enhance biometric security. This includes the development of anti-spoofing technologies tailored to diverse operational environments, reflecting a global effort to safeguard biometric data integrity and privacy across public and private sectors.

4 Literature Survey

In the past, detecting fake fingerprints mainly relied on expensive and rigid hardware. This included special sensors and materials designed to spot counterfeit

prints. Nowadays, the field has moved towards using software with machine learning, which is more flexible and cost-effective. Techniques used now include analyzing the texture, ridge patterns, and sweat patterns on fingerprints to tell if they are real or fake.

Convolutional neural networks (CNNs) are popular in these methods because they are very good at recognizing patterns. Despite these advances, there are still challenges, such as creating diverse and reliable datasets and making sure the models work well with different types of fake fingerprints. Researchers are focusing on improving these models to make them more accurate and adaptable for various spoofing scenarios(Research gate)

We used what we learned from research to create a fingerprint recognition model. It's currently being set up on the Raspberry Pi 4B and uses advanced techniques for processing fingerprints inspired by research methods and sensor strategies. We aimed to make a reliable system that works well in real-life situations. Using the Raspberry Pi 4B's computing power, we ensured it runs smoothly in various applications needing secure biometric authentication.

5 Equipment Used

- Fingerprint Sensor R307

Figure 2: Fingerprint sensor R307

- Raspberry Pi 4B

Figure 3: Rpi

- ST 7735 LCD display
- 64GB SD Card
- RTC Module

Figure 4: RTC module

- USB to UART Serial Converter
- ST7735 LCD Display
- Breadboard and Jumper Wires
- Push Buttons

6 Methodology

We followed a structured approach to achieve our project's goals, beginning with clear objectives: ensuring secure and reliable fingerprint recognition that can detect spoofing attempts.

6.1 setting up the system and configure Hardware

We chose several essential hardware components to set up the system: the Raspberry Pi 4B as the main unit, the R307 Fingerprint Sensor for capturing fingerprint images, and the ST7735 LCD display for displaying output. We also included an RTC module for time-stamping authentication, a 64GB SD card for storage, and a USB to UART Serial Converter for communication between the sensor and Raspberry Pi. Breadboards, jumper wires, and push buttons were used for prototyping and making connections. After verifying connections and configuring GPIO pins, we ensured all project parts were ready for development.

Next, we integrated the R307 fingerprint sensor with the Raspberry Pi, validating basic functions like enrollment and detection. We reviewed essential libraries such as

- **Adafruit:** For communication with fingerprint sensors.
- **FPM:** To manage and interface with the R307 fingerprint sensor.
- **PySerial:** For serial communication between Raspberry Pi and fingerprint sensors.
- **pyfingerprint:** Specifically for handling fingerprint sensor operations (enrollment, search, deletion).
- **TensorFlow:** For developing and integrating machine learning models for fingerprint recognition and spoof detection.
- **PyTorch:** For training and deploying machine learning models used in the system.

6.2 Spoof Fingerprint Creation and Image Acquisation

For our project, we utilized silicone putty to shape molds and Fevicol to generate artificial fingerprints, resulting in the authentic spoofs for testing.

Then, we Ensured clear and detailed fingerprint images were captured from diverse individuals, essential for effective machine learning model training.

We also Optimized image quality by adjusting sensor settings and enhancing captured images through normalization and resizing (Standardizing the dimensions of images to a uniform size suitable for processing and analysis). Additionally, we applied the ten-crop process, which involves cropping the input into ten patches, helping the model learn to recognize objects from multiple perspectives for optimal image quality.

6.3 Model Training and Optimization

- Integrated the ResNet-18 machine learning model to enhance spoof detection capabilities.
- Utilized Google Colab's GPU for efficient model training, enhancing pattern recognition capabilities.
- Deployed the trained digit recognition code on Raspberry Pi, refining performance and addressing initial challenges.

- Implemented spoof detection features, enhancing overall system security and reliability.

6.4 Software Development

- Developed algorithms for fingerprint enrollment, deletion, search functionalities, and liveness detection using Python.
- We implemented seamless communication between hardware components, including the fingerprint sensor, LCD display, and peripherals, while designing a user-friendly interface on the LCD for intuitive user interaction. The interface features the following functions:

    1. Enroll a new fingerprint.
    2. Delete an existing fingerprint.
    3. Toggle anti-spoof detection (enable/disable) using machine learning.
    4. Return to the main menu.

6.5 Conclusion and Result

The model is 80 percent accurate with the dataset we are having already

Model	Dataset	Accuracy
Resnet-18	CrossMatch	around 80
	GreenBit	around 80
	Average Cross Sensor	around 80

Figure 5: Comparison of models and accuracies

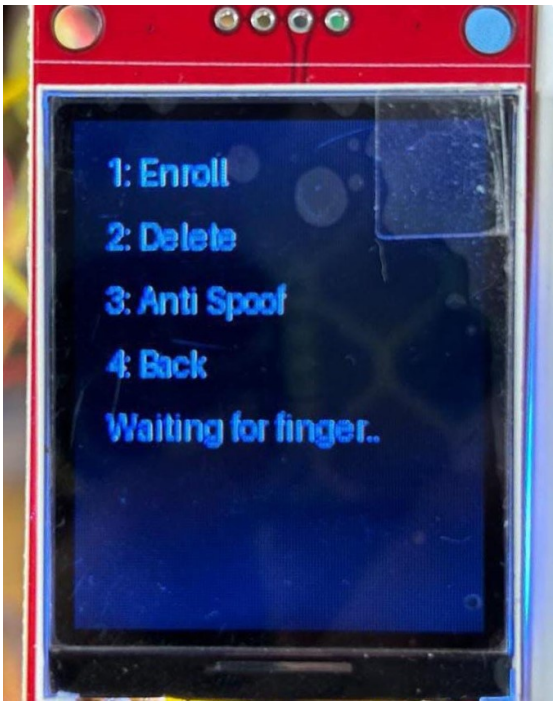

Figure 6: ST7735 DISPLAY

Indian Institute of Technology Kanpur

PIL-logo.png

SURGE INTERNSHIP REPORT

Water Quality Estimation Device

Student Name	Student ID
Naina	2430527

Guide:

Dr. Tushar Sandhan

Submission Date: 13/07/2024

7 Abstract

The Water Quality Estimation Device project aims to develop a portable device for monitoring key water quality parameters.This device is designed to be easily transportable, ensuring practical utility in real-life situations. Our approach involved conducting extensive research on essential water quality parameters. Subsequently, we carefully selected the most pertinent and suitable characteristics to assess water quality effectively. We also included air quality characteristics to gain insights into the underlying water quality. The device uses Arduino programming to connect with sensors and process data, ensuring accurate measurements through calibration. Its user-friendly interface makes it easy for both professionals and non-experts to interpret data in water quality monitoring. Currently, the device experiences a computation delay of 1 second, which we aim to minimize using machine learning models.

8 Introduction

Access to clean and safe water is essential for human health and well-being. According to the World Health Organization (WHO), approximately 2.2 billion people worldwide lack access to safely managed drinking water services, and 785 million still lack a basic one. Contaminated water significantly contributes to waterborne diseases and is responsible for an estimated 485,000 deaths yearly. The report suggests improving how governments manage water, investing more money, teaching people new skills, making better data systems, and encouraging new ideas. It asks governments to focus on fair and long-lasting water services, deal with climate change effects, and make sure everyone gets the help they need.

9 Literature Survey

Previous studies and projects have explored various methods for monitoring water quality. Traditional laboratory-based methods, although accurate, are time-consuming and expensive. Portable devices have been developed to provide quicker results, but they often lack comprehensive analysis and affordability. Recent advancements in sensor technology and machine learning offer new opportunities for developing cost-effective and efficient water quality monitoring devices. By integrating multiple sensors and leveraging data analysis techniques, it is possible to create a device that provides reliable real-time water quality assessments.

10 Equipments Used

11 Equipments Used

- pH sensor
- Temperature Sensor (DS18B20)
- MQ135 Air Quality Sensor

- TDS Sensor
- Conductivity Sensor
- Arduino Mega
- 16x2 LCD Display
- Multimeter
- Arduino IDE
- Power Supply (9V battery)
- Push Buttons
- Buck Converter

Calibration Solutions

- pH 4.0 and pH 7.0 buffer solutions
- 332 ppm NaCl Solution
- Standard Conductivity Solution

12 Methodology

12.1 Sensors and their calibration

12.1.1 pH Sensor

A pH sensor is used to measure the hydrogen-ion activity in water-based solutions, indicating its acidity or alkalinity.

Figure 7: pH Sensor

- Its Calibration:

  The pH sensor needs regular calibration using standard buffer solutions to ensure accuracy. The calibration process involves adjusting the sensor to known pH values.

The sensor was first immersed in a neutral buffer solution (pH 7.0) until the reading stabilized. The meter was then adjusted to match this known pH value. Next, the sensor was rinsed thoroughly and immersed in a second buffer solution (pH 4.0). After the reading stabilized, the meter was again adjusted to match this known pH value.

To determine the relationship between pH and voltage, the slope (m) was calculated using the formula:

$$m = \frac{(pH_2 - pH_1)}{(V_2 - V_1)}$$

where $pH_1 = 7$ and $V_1 = 3.232\,V$, $pH_2 = 4$ and $V_2 = 3.615\,V$. This results in:

$$m = \frac{(4-7)}{(3.615-3.232)} = \frac{-3}{0.383} = -7.8328\,\text{pH/V}$$

This calibration ensures the sensor's readings are accurate and reliable.

12.1.2 MQ135 Sensor

The MQ135 is a gas sensor that detects various pollutants in the air. It's commonly used to measure air quality in environments such as homes, workplaces, and industrial settings.

Figure 8: MQ135 Sensor

- Its Calibration:

  To determine the relationship between the analog output and gas concentration, the slope (m) was calculated using known gas concentrations and corresponding output voltages.

  The sensor was exposed to a standard gas with a known concentration of 400 ppm CO2, giving an output voltage of 0.1 V. It was then exposed to a concentration of 1000 ppm air (which was muddy algae-covered standing water air behind Hall 6), resulting in an output voltage of 2.0 V.

  Using these values, the slope was calculated with the following formula:

$$m = \frac{(C_2 - C_1)}{(V_2 - V_1)}$$

  where: - $C_1 = 400\,\text{ppm}$, $V_1 = 0.1\,\text{V}$ - $C_2 = 1000\,\text{ppm}$, $V_2 = 2.0\,\text{V}$

  Substituting the values:

$$m = \frac{(1000-400)}{(2.0-0.1)} = \frac{600}{1.9} = 315.79\,\text{ppm/V}$$

  This calibration ensures that the sensor accurately converts its analog output voltage to the corresponding gas concentration.

12.1.3 TDS Sensor

A TDS (Total Dissolved Solids) sensor measures the concentration of dissolved solids in water. These solids can include salts, minerals, metals, and other substances.

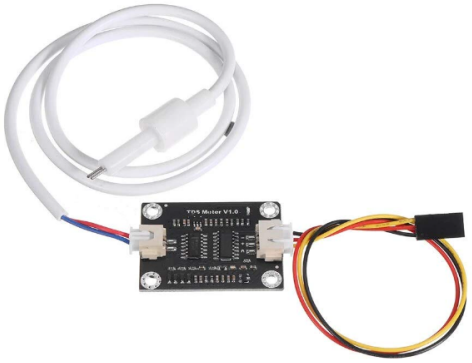

Figure 9: TDS Sensor

- Its Calibration:

  The TDS sensor was calibrated using a 332 ppm NaCl (sodium chloride) solution. During calibration, the analog values produced by the sensor were recorded and compared with the known concentration of the NaCl solution. A calibration factor of 342/462 was calculated to adjust the sensor's analog output to accurately reflect the actual TDS (Total Dissolved Solids) value in water.

12.1.4 Temperature Sensor(DS18B20)

The DS18B20 temperature sensor measures water temperature, which is critical for monitoring chemical reactions, dissolved oxygen levels, and the metabolism of aquatic life.

Figure 10: Temperature Sensor

- Its Calibration:

  The DS18B20 temperature sensor is usually already calibrated by the manufacturer, so users typically don't need to calibrate it themselves.

12.1.5 Conductivity Sensor

A conductivity sensor measures a solution's electrical conductivity, indicating the presence of dissolved ions, such as salts, acids, and bases.

- Its Calibration:

  The calibration factor for the conductivity sensor, based on the 332 ppm NaCl solution with an estimated conductivity of 664 µS/cm, was determined to be 664 µS/cm. This factor is used to calibrate the sensor's readings to measure the conductivity of solutions in various applications accurately. *conductivity calibration factor : 664 µS/cm*

13 Our Prototype and Results

The sensors will be enclosed in a compact, portable casing. The device is designed to provide a user-friendly interface for real-time monitoring of water quality.

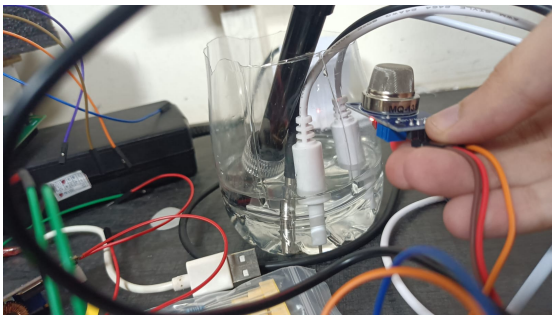

Figure 11: All Sensor Together

Figure 12: ST 7735 display

14 Future Improvements

- Further testing with a broader range of water samples will enhance the device's reliability and usability, ensuring it meets diverse water quality monitoring needs.
- By using machine learning models, we can speed up data processing and improve sensor accuracy by predicting values based on past data and trends. This allows for faster analysis and real-time monitoring.