Naina |220680|Prof.Tushar Sandhan |Department of Electrical Engineering

System and Methods for Fingerprint Anti – spoofing

INTRODUCTION

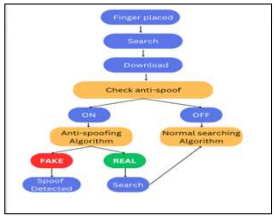
- Fingerprint recognition technology faces spoofing challenges.
- Liveness detection verifies fingerprint authenticity beyond images.
- This project implements a machine learning model on a Raspberry Pi uses ML with Ten Crop for accurate spoof detection, enhancing security in biometrics against fake prints.

OBJECTIVE

- Integrate a pre-trained ResNet18 model for liveness detection into a R - Pi 4B with necessary components.
- Implement a spoof detection algorithm using advanced techniques for accurate fingerprint distinction.
- Conduct thorough testing and prepare for deployment in secure biometric authentication applications.

HARDWARE COMPONENTS

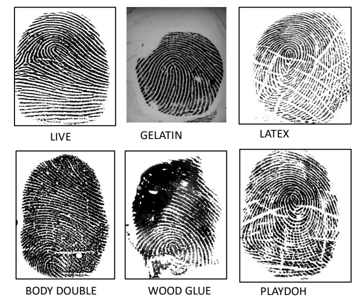
Raspberry Pi 4B,Fingerprint Sensor R307,ST7735 LCD Display,64GB SD Card, USB to UART Serial Converter, RTC Module, Breadboard and Jumper Wires, Push Buttons.



SOFTWARE COMPONENTS

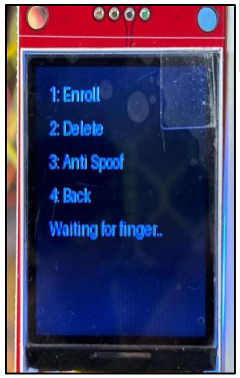
Python Standard Libraries:
time, hashlib, tempfile, threading,collections.OrderedDict
External libraries:
pyfingerprint, RPi.GPIO, torch, torch.nn, torchvision.models, torchvision.transforms, PIL, adafruit_rgb_display.st7735

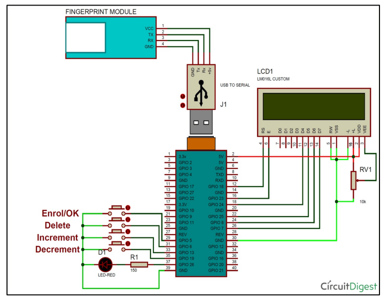
SPOOF IMAGES



METHODOLOGY

Phase 1: Setup and Familiarization
•Tested Raspberry Pi Model 4B using VNC Viewer and managed remotely via PuTTY for command-line operations.
•Configured GPIO pins for hardware components and studied machine learning model deployments.
•Ensured readiness of project components for development.
Phase 2: Sensor Integration and Initial Testing
•Integrated R307 fingerprint sensor with Raspberry Pi, validating functions like enrollment and detection.
•Reviewed and utilized essential libraries (Adafruit, FPM, PySerial, pyfingerprint, TensorFlow, PyTorch) for sensor and machine learning integration.
•Experimented with materials for spoof fingerprint creation and addressed detection challenges.
Phase 3: Image Acquisition and Software Development
•Captured high-quality images for machine learning model training, ensuring clarity and suitability.
•Optimized fingerprint enrollment process, focusing on image quality and sensor functionality.
•Ensured software stability, resolved corruption issues, and ensured smooth operation on both Raspberry Pi and PC.
Phase 4: Model Training and Optimization
•Utilized Google Colab's GPU for efficient model training, enhancing pattern recognition capabilities.
•Deployed digit recognition code on Raspberry Pi, refining performance and addressing initial challenges.
•Implemented spoof detection features to enhance system security and reliability.
•Optimized image processing pipelines, ensuring robust performance of the Ten Crop process.
Phase 5: Finalization and Deployment
•Successfully Deployed Final Code to raspberry Pi .
•Tested system with various spoof attempts, like gelatin, woodglue, latex ,etc.
•Planned future enhancements and improvements based on project outcomes and feedback.





EXPERIMENTAL ANALYSIS

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

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

CONCLUSION

- Our hardware prototype is set up, and the machine learning model achieves 80% accuracy.
- Having achieved smooth operation, our focus shifts to refining device design and enhancing structure for seamless integration and optimal real-world performance of our robust prototype.

REFEERENCES

1. Faizah Alqahtani1, Rachid Zagrouba2, Fingerprint Spoofing Detection Using Machine Learning , Conference 2020
2. Dr, Tushar Sandhan , Vulnerability discovery and liveness detection for improving the security in biometric authentication systems,C3ihub,iitk
3. Yehjune Heo, Generative Adversarial Network Based Fingerprint Anti-Spoofing Limitations, WSAT International Journal of Computer and Engineering, 2021

Water Quality Estimation Devi

INTRODUCTION

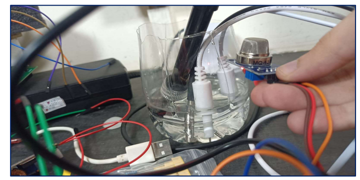
Access to safe drinking water is a global challenge. This project aims to develop a portable device that monitors water quality in real-time, using sensors for pH, temperature, TDS, conductivity, and air quality. The device provides cost-effective and immediate assessments to ensure water safety.

SENSORS AND EQUIPMENTS

- pH Sensor: Measures acidity/alkalinity.
- Temperature Sensor (DS18B20): Measures water temperature.
- MQ135 Air Quality Sensor: Detects air pollutants.
- TDS Sensor: Measures total dissolved solids.
- Conductivity Sensor: Measures electrical conductivity.
- Microcontroller: Arduino Mega.
- Display: ST7735 LCD

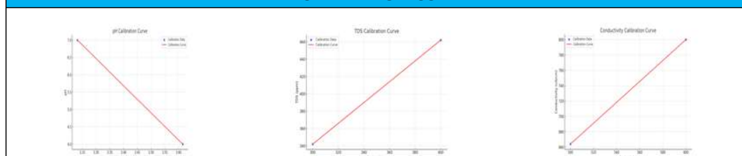
METHODOLOGY

Sensor Integration: Sensors connected to an Arduino microcontroller with data displayed on LCD.
Calibration:
•pH: Using pH 4.0 and pH 7.0 buffer solutions.
•TDS: Using 332 ppm NaCl solution.
•Conductivity: Using the same solution as TDS.
Data Processing: Initial computation delay of 1 second, to be reduced with machine learning.



SENSORS

CALIBRATION CURVE



CONCLUSION AND RESULTS

- The device effectively monitors:
- pH: Accurate readings within 6.5 to 8.5.
- Temperature: Crucial for water quality.
- Air Quality: Detects pollutants indicating water contamination.
- TDS Levels: Above 500 ppm is unsuitable for drinking.
- Conductivity: Above 2000 µS/cm indicates high pollution.
- The prototype is cost-effective for real-time monitoring. Future improvements with machine learning will reduce delays and enhance accuracy.

FUTURE IMPROVEMENTS

Machine Learning: To reduce computation delay of 1 sec and improve accuracy.
Extended Testing: Broader range of water samples to enhance reliability.

REFERENCES

1. Mohamed cheniti* and Belaout Abdesslam, An Arduino-based Water Quality Monitoring System Using pH, Temperature, Turbidity, Tds Sensors
2. World Health Organization : WHO (2023, March 24)
3. Wong Jun Hong, Water Quality Monitoring with Arduino Based Sensors, MDPI Journal List