Fingerprint Biometric Identification System - Project Report

This project implements a simple Fingerprint Biometric Identification System using Python. The system allows users to enroll fingerprint images (uploaded from disk) along with personal details (ID, name, gender, age) and later verify identity by matching a supplied fingerprint image against the stored database.

Objectives

Build a user-friendly GUI for enrollment and verification using Tkinter. Extract simple fingerprint features and store them in a local JSON database. Allow verification by comparing fingerprint-derived hashes (simple matching method). Produce a prototype suitable for academic demonstration and further improvement.

Technologies Used

Python, Tkinter (GUI), OpenCV (image processing), NumPy, Pillow (PIL), JSON file storage.

System Design

The system has two main modules implemented as tabs in the GUI: 1. Enrollment: Accepts user details and fingerprint image, processes the image to create a simple hash, and saves the user entry in a JSON file. 2. Verification: Accepts a fingerprint image and checks for an exact hash match against saved fingerprints. On match, it displays user details.

Database (JSON) Structure

The saved JSON stores user entries keyed by user ID. Example: { "U001": { "name": "Alice", "gender": "Female", "age": "21", "fingerprint": 123456789012345 } }

How it Works

1. The fingerprint image is resized to 200x200 pixels and histogram-equalized for contrast normalization. 2. A binary threshold is applied to convert the image to a consistent binary pattern. 3. The binary bytes are hashed (Python built-in hash of the bytes) and stored as the fingerprint feature. 4. Verification compares the stored hash values with the hash of the newly uploaded image. If equal, a match is declared.

Limitations and Future Work

The matching method uses a simple hash comparison and requires near-identical images; it is not robust to rotation, translation, or partial prints. Future improvements: ORB/SIFT-based matching, machine learning (CNN/Siamese) fingerprint matching, support for live scanners, encrypted storage, and DBMS backend.

How to Run

1. Install dependencies: pip install opencv-python pillow numpy reportlab 2. Run the application: python scanner.py 3. Use the GUI to Enroll users (Upload fingerprint image + provide ID/name/gender/age), then Verify by uploading another fingerprint image.

Conclusion

This project serves as a compact prototype demonstrating the basics of fingerprint-based identification using image uploads and simple feature hashing. It is suitable for academic submissions and can be extended as described above.