**Facial Recognition Attendance System with Anti-Spoofing Technology**

**Abstract**

This document details the design and implementation of a secure, automated attendance system that leverages facial recognition technology combined with robust anti-spoofing measures. Traditional attendance methods are often susceptible to inaccuracies and fraud, while basic facial recognition systems can be deceived by static images or videos. To address these vulnerabilities, this project integrates a liveness detection mechanism that validates the presence of a live individual before marking attendance. The system uses real-time blink detection, analyzing facial landmarks to differentiate between a live person and a spoofing attempt. Upon successful verification, the system logs the user's attendance with a timestamp, creating a reliable and tamper-resistant record. The solution is built using Python with the face\_recognition and mediapipe libraries, offering a practical and effective tool for deployment in academic institutions, corporate environments, and other organizations.

**1. Introduction**

Maintaining accurate and secure attendance records is a fundamental requirement for many organizations. Traditional methods, such as manual sign-in sheets or RFID cards, are prone to issues like human error, time theft, and "buddy punching." While automated facial recognition systems offer a more convenient alternative, they introduce new security challenges. A simple facial recognition system can often be fooled by presenting a photograph or a video of a registered user to the camera, a technique known as a spoofing attack.

This project directly confronts this challenge by developing an intelligent attendance system that not only recognizes who is in front of the camera but also verifies that they are a live person. By implementing an anti-spoofing layer, the system ensures the integrity of the attendance data. The core innovation is the integration of a liveness check that requires the user to perform a natural action—blinking—to prove their presence. This creates a two-factor verification process: identity confirmation via facial recognition and liveness confirmation via blink detection, resulting in a system that is both user-friendly and highly secure.

**2. System Architecture and Workflow**

The system is designed with a clear, user-centric workflow, from initial registration to the final attendance logging. The process is divided into two primary phases: user enrollment and attendance marking.

**2.1 User Registration**

Before attendance can be marked, users must be enrolled in the system. This is a one-time process handled by a dedicated registration script (newFaceRegistration.py).

1. **Image Capture:** The system activates a camera and captures a series of images of the new user's face.
2. **Quality Assurance:** To ensure the accuracy of future recognitions, each captured image is analyzed for quality. A sharpness check using the Laplacian variance is performed to discard blurry images.
3. **Facial Encoding:** For each high-quality image, the system generates a unique 128-dimension facial encoding.
4. **Data Storage:** The collected encodings are saved to a .pkl file, named with a unique user ID, within the Encodings directory. This creates a database of all registered users.

**2.2 Attendance Marking and Anti-Spoofing**

The core functionality of the system is the attendance marking process, which includes the critical anti-spoofing check.

1. **Face Detection and Recognition:** The system continuously monitors the camera feed. When a face is detected, it generates a facial encoding in real-time. This encoding is compared against the database of known encodings to find a match. If a match is found with a sufficient confidence level, the user's ID is identified.
2. **Liveness Detection (Anti-Spoofing):** Upon identifying a registered user, the system immediately initiates a liveness check to prevent spoofing.
   * **Facial Landmark Analysis:** The system uses the MediaPipe Face Mesh library to detect a detailed map of 478 facial landmarks in real-time.
   * **Blink Detection:** It specifically monitors the landmarks around the eyes to calculate the Eye Aspect Ratio (EAR). The EAR is a value that is high when the eyes are open and drops close to zero when the eyes are closed.
   * **Verification:** The user is prompted to blink. The system looks for a rapid drop in the EAR value that persists for a few consecutive frames. If this pattern is detected within a set timeout period, the liveness check is successful. If no blink is detected, the system flags it as a potential spoofing attempt, and attendance is not marked.
3. **Attendance Logging:** If and only if the user is successfully recognized and passes the liveness check, their attendance is recorded. The system logs the user's ID and the current timestamp into a daily CSV file. To prevent duplicate entries, the system checks if the user has already been marked present for that day before adding a new record.

**3. Technical Implementation**

The system is developed in Python and relies on several powerful, open-source libraries for its core functionalities.

* **Facial Recognition:** The face\_recognition library is used for detecting faces, generating 128-d encodings, and comparing faces to identify users.
* **Anti-Spoofing and Liveness Detection:** The MediaPipe library provides the tools for real-time facial landmark detection, which is the foundation of the blink detection mechanism.
* **Image and Video Processing:** OpenCV is used to capture and handle the video feed from the camera, display the output to the user, and perform basic image processing tasks.
* **Data Handling and Manipulation:** NumPy is used for efficient numerical operations, particularly in calculating distances and ratios for the liveness check. Pickle is used for serializing and storing the facial encoding data. The os and datetime libraries are used for file management and timestamping.

The project is structured with a command-line interface that allows an administrator to choose between marking attendance or registering a new user, making it easy to manage and operate.

**4. Conclusion**

This project successfully demonstrates the development of a secure and reliable facial recognition attendance system with integrated anti-spoofing technology. By implementing a blink-based liveness check, the system effectively mitigates the risk of spoofing attacks from static photos, significantly enhancing its security compared to basic recognition systems. The modular architecture, which separates user registration from attendance marking, makes the system scalable and easy to maintain. This solution provides a robust framework that can be deployed in various organizational settings to automate attendance tracking accurately and securely, ensuring that records are both convenient to capture and difficult to falsify.