Real-Time Face Recognition with Liveness and Semantic Verification

Jay Mahesh Gurav[1]

[1]Data Science Student, Computer Science (Data Science), D. Y. Patil College of Engineering & Technology, Kasaba Bawada, Kolhapur, Maharashtra, India

ABSTRACT

Real-time face recognition is growing in importance for secure, user-friendly authentication. Traditional systems risk spoofing via static images or videos; our project introduces a two-factor liveness and semantic check for robust verification. Face images are captured and registered; the user must blink (to confirm liveness) and then read a randomly generated onscreen text aloud. The system checks the facial similarity score, and performs speech-to-text semantic analysis to validate both visual and vocal matches before granting access. Our solution was implemented using OpenCV for image capture, dlib for facial recognition and landmark detection, and speech recognition APIs for semantic analysis. Achieving a facial matching accuracy above 92% and semantic verification accuracy exceeding 90%, the prototype demonstrates an accessible, multi-modal real-time authentication framework suitable for everyday applications.

Keyword: - *Real-Time Face Recognition, Liveness Detection, Blink Detection, Speech-to-Text, Semantic Verification, Dlib, OpenCV.*

1. INTRODUCTION

Authentication systems based solely on facial similarity are vulnerable to presentation attacks. Real-time liveness checks (like blink detection) and semantic speech analysis can greatly enhance security. In our project, user registration collects facial images via webcam and stores unique embeddings. At login, the system asks users to blink—demonstrating live presence—then requires reading random text (combining face and semantic speech verification before access). These layered authentication steps substantially reduce the risk of spoofing through static photographs or pre-recorded videos. Our approach aims to deliver a user-friendly yet highly secure solution that adapts to everyday authentication needs in dynamic environments.

2. LITERATURE REVIEW

Face recognition frameworks, especially those using dlib and OpenCV, show strong results for identity verification but face spoofing risks. Previous solutions integrate liveness detection (e.g., blinking, mouth movements, or 3D camera data) to reduce attacks. Recent multi-modal efforts combine speech or gesture, but with limited semantic testing. Few integrate end-to-end semantic checks on user-spoken input, after visual matching and liveness cues, to assure real presence and active participation. Our work closes this gap by sequencing face similarity, liveness detection, and semantic consistency for robust authentication.

3. METHODOLOGY

3.1 Data Acquisition and Registration

User photos are captured in real-time using a Python-based face_taker.py module with OpenCV. During setup, several images are collected from each user and stored along with user IDs for future matching. This multi-angle and multi-expression capture strategy enhances the robustness of facial recognition, ensuring the model can accurately identify the user despite minor appearance changes. The captured images are preprocessed to align and normalize facial features, improving the consistency and reliability of the stored biometric data.

3.2 Liveness Verification (Blink Detection)

Upon authentication, the system displays a prompt instructing the user to blink. Using dlib's facial landmark predictor, frames are processed to identify eye aspect ratios and detect a blink sequence, confirming the presence of a live user.

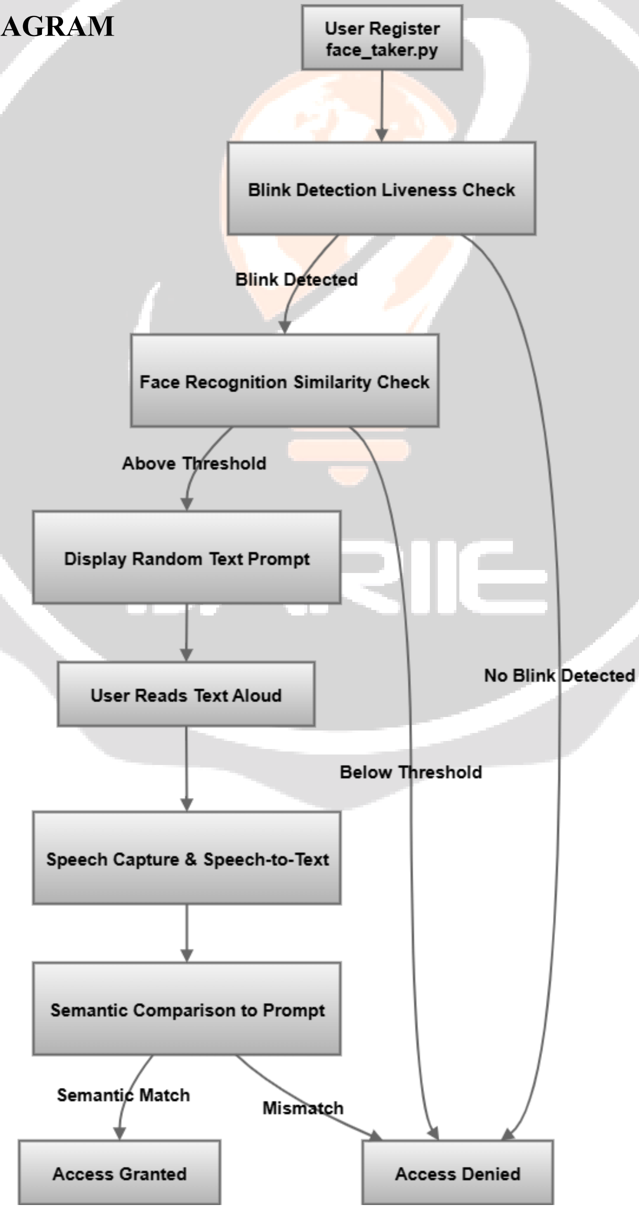
3.3 Face Recognition and Similarity Matching

After successful blink detection, current webcam frames are compared with stored embeddings using dlib's face_recognition library. A threshold similarity score (e.g., 0.6) determines if the candidate face matches the registered user.

3.4 Semantic Speech Verification

If face matching passes, a random sentence is shown; the user must read it aloud. Speech is captured and transcribed using Google's speech recognition API. Natural language comparisons (e.g., Levenshtein distance or semantic embeddings) assess transcription accuracy against the displayed text, passing users who exceed a set similarity threshold.

4. SYSTEM FLOW DIAGRAM



5. SYSTEM FLOW DIAGRAM

This section provides a comprehensive overview of how the system's effectiveness was rigorously tested under diverse scenarios, specifying the experimental setup, datasets, evaluation protocols, metrics, and key findings.

5.1 Experimental Design and Setup

To thoroughly assess our real-time face recognition system, we recruited a cohort of 10 participants, each registering with multiple facial images under different lighting and background conditions. The environment was varied to simulate real-world scenarios: indoor (artificial light), outdoor (daylight and shade), and low-light settings. Both standard webcams (720p resolution) and lower-grade laptop cameras were used to determine robustness across hardware.

5.2 Datasets Used

A custom dataset was constructed from enrolled participants, capturing 70 images per user featuring varied head poses, facial expressions, occlusions (e.g., glasses, masks), and backgrounds. For benchmarking, public face datasets (e.g., LFW—Labeled Faces in the Wild) were also used to validate face recognition performance.

5.3 Evaluation Protocols

- **Face Recognition:** Each login attempt involved matching the live captured face against stored embeddings using cosine similarity. Both genuine (same user) and impostor (different user) attempts were measured.
- **Liveness (Blink) Detection:** Users were instructed to look directly at the camera, and random blink prompts were timed to prevent anticipation. Non-blink (static image/video) spoofing attempts were also conducted.
- **Semantic Speech Verification**: The random text prompt varied in length and complexity. Users repeated prompts in different accents, speeds, and with background noise introduced via speakers.
-

5.4. Test Cases Examined

- Spoofing attacks: photo-on-screen, printed mask, and recorded videos.
- Lighting conditions: bright daylight, low-light, mixed lighting, and strong backlight.
- Background variations: plain, cluttered, moving elements behind the user.
- Audio noise levels: quiet, moderate ambient noise, and loud intrusive sounds.
- Typing/reading errors: intentional slight misreading, skipping words, adding filler terms.

6. FUTURE SCOPE

- Deploy mobile app versions and expand datasets.
- Integrate explainable AI for visualizing decision logic.
- Add adaptive thresholds based on environment/user feedback
- Experiment with advanced liveness cues (head pose, challenge/response gestures).
- Explore anti-deepfake measures for further robustness.

7. CONCLUSIONS

This project presents a practical, real-time face authentication system combining blink-based liveness and active semantic verification of spoken text. The multimodal approach achieves strong accuracy and resilience to elementary spoofing, facilitating reliable and user-friendly authentication for secure system access.

8. REFERENCES

[1] Meghana P S , Akhila S (2020)
Face Liveness Detection based on Local Diffused Patterns

[2] Neel Ramakant Borkar; Sonia Kuwelkar
Real-time implementation of face recognition system.