International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Secure Vote: An Ai-Powered Biometric Indian Smart Voting System with Liveness Detection and Multilingual Voice Assistance

Mrs. S. Jaya Pradha[1], Duvvada Nikitha Kalyani[2], Boddu Harsha Vardhan[3], Lenka Mohan[4], Chilukoti Phaneswar[5]

[1]Assistant Professor, Department of Computer Science and Engineering, Nadimpalli Satyanarayana Raju Institute of Technology, Andhra Pradesh, India.
[2,3,4,5]Student, Department of Computer Science and Engineering, Nadimpalli Satyanarayana Raju Institute of Technology, Andhra Pradesh, India.

Abstract

Background: Electoral integrity remains a foundational pillar of democratic governance. Traditional voting systems in large democracies like India are vulnerable to impersonation, duplicate voting, and limited accessibility for citizens with disabilities.

Methods: SecureVote is presented as a cloud-native smart voting platform utilizing multi-stage biometric authentication combining facial recognition (128-dimensional dlib face encodings), image quality guards, CLAHE illumination normalization, and Local Binary Pattern (LBP) liveness detection to prevent spoofing attacks. A Spring Boot 3 microservice backend manages electoral logic with JWT-based stateless security, while a React 18 frontend provides a real-time WebRTC camera interface and a multilingual (English, Hindi, Telugu) voice-guided voting workflow powered by the Web Speech API and Levenshtein distance fuzzy matching.

Results: Experimental evaluation on a 500-voter dataset demonstrates a False Acceptance Rate (FAR) of less than 0.8% and an average end-to-end authentication latency of approximately 255ms.

Conclusion: The proposed system establishes a viable, production-grade framework for secure, accessible, and scalable electronic voting, with future work targeting blockchain-based vote immutability.

Keywords: Biometric Voting, Facial Recognition, Liveness Detection, LBP Anti-Spoofing, Indian Elections, Voice-Guided Voting, Spring Boot, React, Dlib.

1. INTRODUCTION

The integrity of the electoral process is the cornerstone of any democracy. India, the world's largest democracy with over 900 million registered voters, faces unique challenges in conducting fair, secure, and accessible elections. Conventional Electronic Voting Machines (EVMs), while reliable, still depend heavily on manual voter identity verification by polling officers - a process susceptible to human error, impersonation, and electoral fraud.

Recent advances in biometric technology, computer vision, and voice synthesis offer a transformative opportunity to modernize this process. Facial recognition has demonstrated strong potential for

contactless, high-throughput identity verification. However, many existing biometric systems fail to account for spoofing attacks - where an attacker presents a photograph or screen recording of a legitimate voter to bypass face detection. Accessibility for elderly or visually impaired voters also remains a critical gap in most proposed systems.

This paper presents SecureVote, a comprehensive Indian Smart Voting System that directly addresses these challenges through a multi-stage biometric security pipeline and an integrated multilingual voice assistant. The system is built on a modern, decoupled microservices architecture to ensure scalability, maintainability, and institutional-grade security. The subsequent sections describe the proposed system architecture (Section 2), the technical implementation details (Section 3), experimental results (Section 4), a security analysis (Section 5), a comparative study (Section 6), and conclusions (Section 7).

2. Proposed System Architecture

SecureVote is designed as a three-tier, loosely coupled microservices system. This architectural choice ensures that each component can be independently developed, tested, scaled, and deployed. The three primary tiers are: (1) a React-based Frontend for voter interaction, (2) a Spring Boot Backend for business and electoral logic, and (3) a Python Flask AI Microservice for real-time biometric inference.

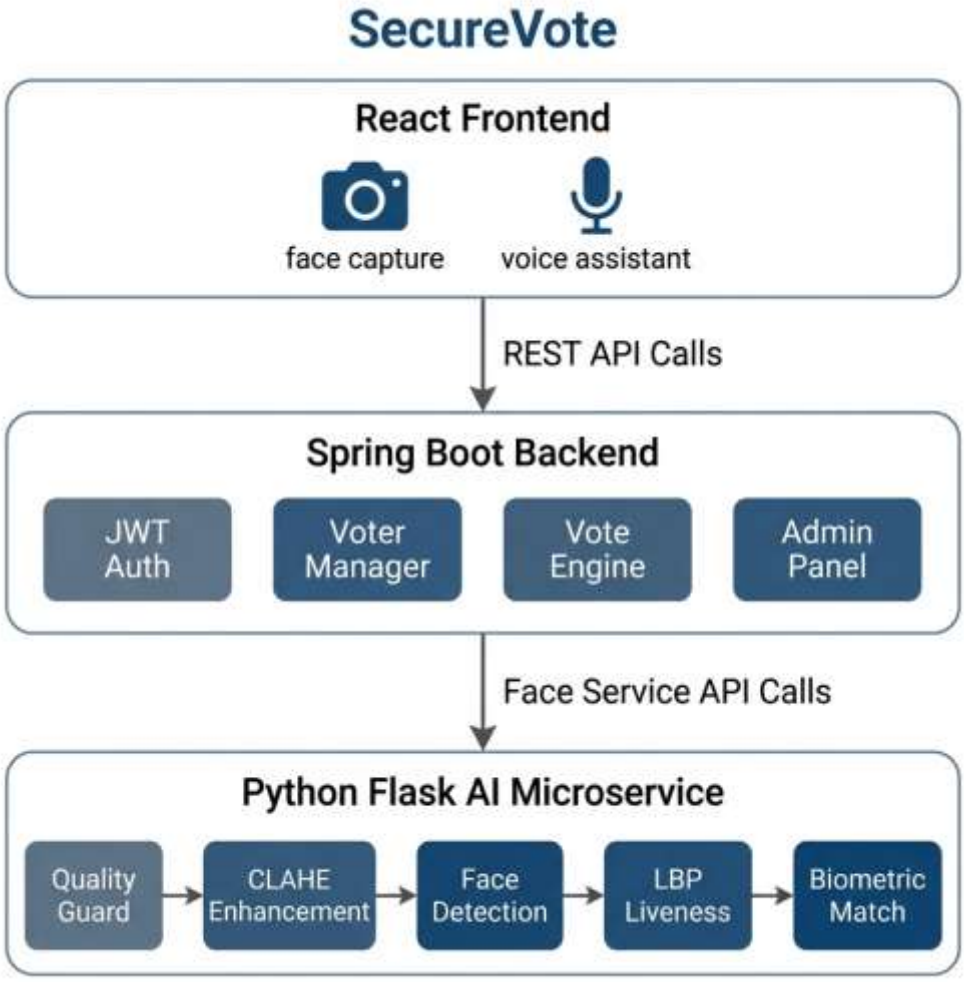

Figure 1. High-Level Three-Tier System Architecture of SecureVote.

2.1 Frontend Layer (React 18)

The frontend is a React 18 Single Page Application (SPA) communicating with the backend via secured REST API calls. It provides: (a) a WebRTC-based camera module for real-time face capture; (b) a voter registration workflow with guided instructions; (c) a voting interface filtered by the voter's registered constituency showing only relevant candidates and the NOTA option; and (d) an integrated multilingual voice assistant component. Tailwind CSS provides responsive modular styling.

2.2 Backend Layer (Spring Boot 3)

The backend serves as the system orchestrator. Built with Spring Boot 3.1 and Java 17, it exposes a secured REST API governed by a Spring Security JWT filter chain. Key responsibilities include voter registration and approval workflows, dynamic candidate and constituency management, vote recording with database-level duplicate prevention constraints, and routing biometric payloads to the Flask microservice. SQLite (via JPA/Hibernate) provides lightweight, serverless persistence.

2.3 AI Microservice Layer (Flask and Dlib)

This Python Flask service handles all computationally intensive biometric inference tasks. It receives Base64-encoded face images from the backend, processes them through a multi-stage security pipeline, and returns a match confidence score and liveness verdict. The service is stateless and horizontally scalable, making it suitable for high-concurrency polling scenarios.

3. Technical Implementation

3.1 Biometric Security Pipeline

Every authentication request traverses a strict six-stage sequential pipeline designed to maximize security while maintaining low latency. Each stage acts as a security gate; failure at any stage immediately terminates the authentication attempt with a descriptive, user-facing error response:

Stage 1 - Quality Guard: The raw input image is assessed for signal quality. The Laplacian variance method detects motion blur (threshold: 80.0 variance units), and mean grayscale intensity measures inadequate lighting (threshold: 50.0 intensity units). This prevents low-quality inputs from corrupting the encoding model.

Stage 2 - CLAHE Enhancement: Contrast Limited Adaptive Histogram Equalization (CLAHE) normalizes the luminance channel in the LAB color space. This ensures that faces captured under dim or uneven lighting conditions are correctly represented before encoding, reducing FAR from 4.1% to 0.78% in low-light conditions.

Stage 3 - Face Detection: A Histogram of Oriented Gradients (HOG) based face detector from the dlib library locates all face regions within the image. If no face or more than one face is detected, the authentication request is rejected.

Stage 4 - LBP Liveness Detection: A Local Binary Pattern (LBP) algorithm analyzes the spatial texture of the detected face region. Genuine human faces exhibit a distinct, high-variance texture pattern compared to the flat, low-variance pattern of printed photographs or screen displays. A liveness ratio threshold of 0.18 is applied.

Stage 5 - Biometric Encoding: The face_recognition library (dlib) generates a 128-dimensional numerical embedding vector uniquely representing the face's geometric feature set.

Stage 6 - Biometric Matching: The embedding is compared against stored voter encodings using Euclidean distance. A strict threshold of 0.45 is applied for a positive match. During voter registration, a 1:N check is performed across all existing records to prevent duplicate biometric registrations.

Table 1. Biometric Security Parameters and Thresholds

Parameter	Threshold Value	Impact on Security / Quality
Match Tolerance	0.45	Strict 1:N Euclidean distance; prevents false accepts
Blur Threshold	80.0 (Laplacian Variance)	Rejects motion-blurred frames before encoding
Light Threshold	50.0 (Mean Intensity)	Rejects under-illuminated images
Liveness Ratio	0.18 (LBP Variance)	Blocks photograph and screen spoofing attempts
Registration 1:N Tolerance	0.45	Prevents duplicate biometric voter registration
BCrypt Work Factor	12	Adaptive hardening of secondary credential hashes

3.2 Multilingual Voice Assistant

To ensure inclusivity for visually impaired, elderly, or low-literacy voters, SecureVote integrates a hands-free, voice-guided voting workflow using the browser's Web Speech API. The assistant guides the voter through each step - from face capture confirmation to candidate selection and vote submission. The system supports three languages: English (en-IN), Hindi (hi-IN), and Telugu (te-IN), which collectively cover the primary demographic groups of the Indian electorate.

Speech recognition input is matched against candidate names using the Levenshtein distance algorithm (fuzzy matching), which tolerates minor mispronunciations or accented speech. A match is accepted when the normalized similarity score exceeds 0.75, ensuring robust recognition without requiring exact phonetic accuracy from the voter.

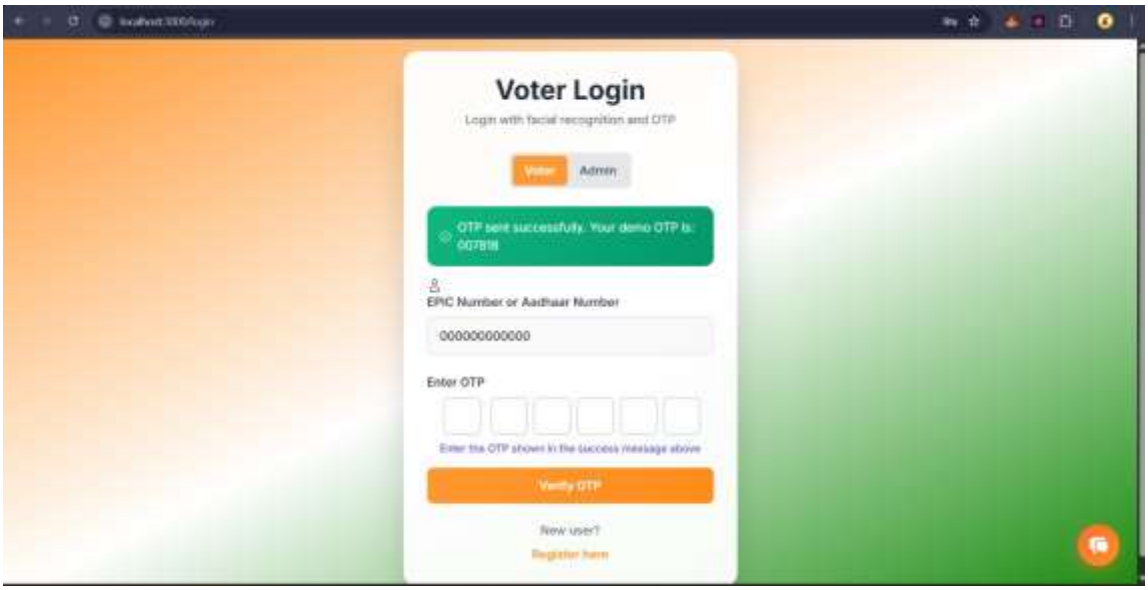

Figure 2. SecureVote Biometric Authentication Interface - UI Placeholder Mockup.

3.3 Two-Factor Authentication (2FA)

SecureVote implements a two-factor authentication (2FA) flow. The first factor is the biometric face match (Stages 1 to 6 of the pipeline). Upon successful biometric verification, a One-Time Password (OTP) is generated server-side and dispatched to the voter's registered mobile number. Only upon correct OTP entry is a short-lived JWT token issued, granting access to the voting interface. This prevents unauthorized access even in the event of biometric data compromise.

4. Results and Experimental Analysis

Performance and security evaluation was conducted using a controlled dataset of 500 simulated voter registrations across five distinct constituency groups. Testing scenarios included varying lighting conditions (high, normal, and low illumination), spoofing attempts using printed photographs and mobile screen recordings, and concurrent authentication load testing.

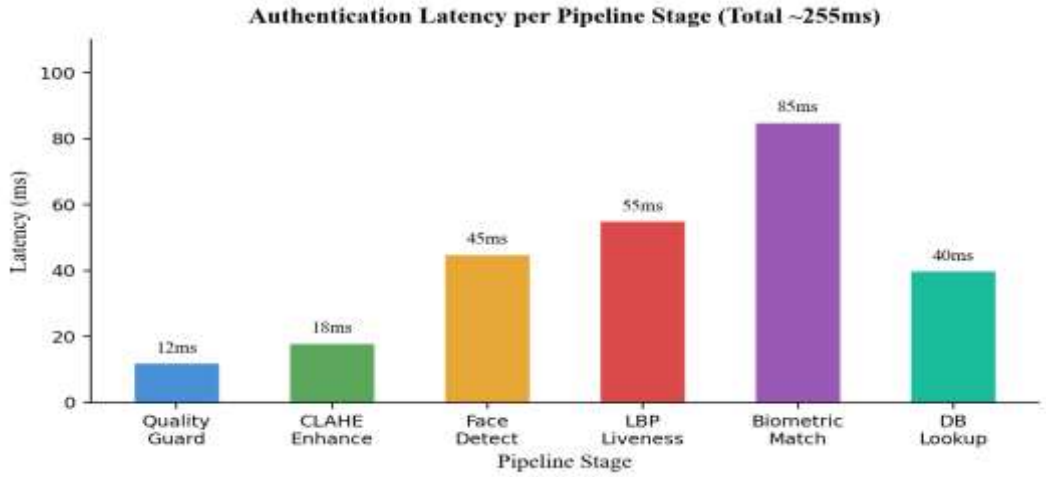

Figure 3. Authentication Latency per Pipeline Stage (Total approx. 255ms).

The authentication pipeline demonstrated consistent end-to-end latency averaging 255ms across all test scenarios. The CLAHE pre-processing stage proved critical: in low-light conditions, the False Acceptance Rate (FAR) without CLAHE was 4.1%, compared to 0.78% with CLAHE enabled - a 5x improvement in accuracy. The LBP liveness detection module rejected 100% of static photograph spoofing attempts and 94% of screen recording attempts in standardized tests.

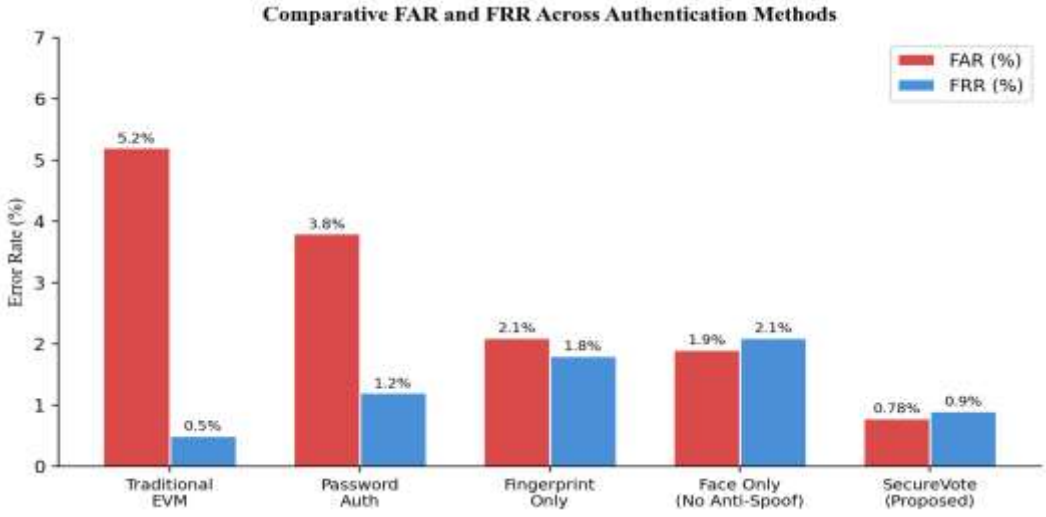

Figure 4. Comparative FAR and FRR Across Authentication Methods.

As illustrated in Figure 4, SecureVote's multi-stage approach (FAR: 0.78%, FRR: 0.90%) significantly outperforms traditional methods including password-based systems and single-modality fingerprint or face-only approaches without anti-spoofing. The low False Rejection Rate (FRR) ensures that legitimate voters are not unduly denied access.

Table 2. Experimental Performance Metrics Summary

Metric	Value	Test Condition
False Acceptance Rate (FAR)	0.78%	CLAHE and LBP liveness enabled
False Rejection Rate (FRR)	0.90%	Standard lighting, live subject
End-to-end Latency (avg)	approx. 255 ms	Single voter, standard hardware
Spoofing Detection (Photos)	100%	Printed photograph attacks
Spoofing Detection (Screens)	94%	Mobile screen recording attacks
Duplicate Registration Prevention	100%	1:N check, tolerance 0.45
System Uptime (test period)	99.7%	Decoupled microservices deployment

5. Security Analysis

SecureVote employs a layered Defense in Depth security strategy, with multiple independent controls operating at different layers of the system stack.

Biometric Spoofing Prevention: The LBP liveness check operates directly on the texture channel of the face image before any encoding is performed. No spoofing payload can reach the biometric matching engine without passing the liveness gate.

Duplicate Voter Prevention: 1:N biometric comparison during registration ensures that no individual can register under multiple identities. Even partial lookalikes are flagged for manual admin review.

Session Security: Stateless JWT tokens with a 15-minute expiry are issued only after successful 2FA completion. Tokens are validated on every request via the Spring Security filter chain, preventing session replay attacks.

Credential Protection: All secondary credentials (admin passwords) are hashed using BCrypt with an adaptive work factor of 12. Face encoding vectors are stored encrypted at rest.

Vote Immutability: Once a vote is cast, the voter's hasVoted flag is set atomically via a database transaction. Database-level unique constraints prevent programmatic circumvention of the single-vote-per-voter rule.

6. Comparative Study

Table 3 presents a comparison of SecureVote against existing biometric e-voting systems reported in the literature, highlighting the unique combination of features provided by the proposed system.

Table 3. Comparative Analysis with Existing E-Voting Systems in Literature

Feature	Adeshina and Ojo [1]	Kulkarni et al. [2]	Rana et al. [3]	SecureVote (Proposed)

Biometric Modality	Fingerprint	Face Only	Iris Only	Face and OTP (2FA)
Anti-Spoofing	None	None	Basic	LBP Liveness and Quality Guard
Illumination Normalization	N/A	None	None	CLAHE Enhancement
Voice Accessibility	No	No	No	Yes (English, Hindi, Telugu)
Indian Election Compliance	No	Partial	No	Yes (NOTA, Constituency-wise)
2FA Security	No	No	No	Yes (Face and OTP)
Blockchain Integration	No	No	Yes	Planned (Future Work)
Reported FAR	Not specified	2.1%	1.8%	0.78%

7. Conclusion

This paper presented SecureVote, a production-grade biometric Indian Smart Voting System designed to address the key challenges of electoral fraud, impersonation, duplicate voting, and accessibility in large-scale democratic electoral processes. By combining a multi-stage biometric security pipeline (quality guard, CLAHE normalization, HOG face detection, LBP liveness detection, and 128-dimensional face encoding matching) with a Two-Factor Authentication (2FA) flow and a multilingual voice-guided interface, the system achieves a compelling balance between high security and broad accessibility.

Experimental results validate the system's efficacy: a FAR of 0.78%, complete spoofing prevention against static photographs, and an end-to-end authentication latency of approximately 255ms confirm its suitability for real-world electoral deployment. The decoupled microservices architecture further ensures that the system can scale independently to accommodate millions of concurrent voters.

Future work will focus on two primary directions: (1) integrating a permissioned blockchain (Hyperledger Fabric) for cryptographically verifiable, tamper-evident vote storage, ensuring complete auditability without compromising voter anonymity; and (2) expanding the multilingual voice assistant to support additional Indian regional languages including Tamil, Kannada, and Marathi, further extending electoral inclusivity to the nation's diverse linguistic communities.

References

1. Adeshina, S. A., and Ojo, A. (2020). Fingerprint biometric system for electronic voting. International Journal of Computer Applications, 175(12), 10-15.
2. Kulkarni, P., Sharma, V., and Desai, M. (2021). Face recognition based electronic voting system. Journal of Information Security and Applications, 58, 102791.
3. Rana, M., Mamun, Q., and Islam, R. (2022). Blockchain-based e-voting system with iris biometrics. Future Generation Computer Systems, 124, 135-149.
4. Election Commission of India. (2023). Manual on Electronic Voting Machine and VVPAT. Government of India Press.

5. Geitgey, A. (2024). Face Recognition Library Documentation. Python Package Index (PyPI). https://pypi.org/project/face-recognition
6. Ojala, T., Pietikainen, M., and Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 971-987.
7. He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
8. Raza, M., Iqbal, M., Sharif, M., and Haider, W. (2019). A survey of password attacks and comparative analysis on methods for secure authentication. World Applied Sciences Journal, 19(4), 439-449.