JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ANDROID VOTING SYSTEM USING FACE RECOGNITION

Mr. D. Shravan Kumar[1] | Setty Nitin[2] | T. Manoj Kumar[3] | G. Vishnu Gowtham[4] |Sk. Junaid Ahmad[5] | T. Keerthan[6]

[1]Assistant Professor , [2,3,4,5,6]B.Tech Student

[2,3,4,5,6]Department of Computer Science and Engineering, Narayana Engineering College, Nellore, Andhra Pradesh, India

Abstract : *In the evolving world of technology, the introduction of e-Voting is set to completely transform the voting process. Traditionally, voting has been carried out manually with the use of paper ballots or through electronic voting machines (EVMs that operate on Direct Response Electronic (DRE) or Identical Ballot Boxes offer a more digitalized and convenient method for multinational companies and organizations to voting has been in demand. In order to overcome the weaknesses of the traditional voting system, we have designed the Android Voting System Using Face Recognition. The proposed system will definitely protect the identity and credentials of the voters are verified, and everyone is encouraged to vote for their favourite option candidate.To address these challenges, we have created an Android voting system that uses face recognition. The cutting-edge system utilizes state-of-the-art biometric technology to enhance the security and reliability of the voting process. Through the utilization of face recognition technology, the system can effectively confirm the identity of all voters, thus eradicating the possibilities of impersonation and unauthorized votes. This innovative system not only safeguards the identity and information of all voters but also streamlines the voting procedure, making it more accessible and user-friendly for all individuals.*

Index Terms **- Face Recognition, Android Application, Authentication, Database Integration, Security, Voting Protocols**

I. INTRODUCTION

In today's rapidly evolving digital landscape, biometrics play a crucial role in enhancing security and convenience across various applications. The Android Face Voting System represents an innovative leap in this domain, employing facial recognition technology to revolutionize the voting process. This system aims to deliver a seamless, user-friendly, and highly secure voting experience, setting a new benchmark for electoral integrity.The system leverages facial recognition to link each vote to a person's unique biometric data, thereby minimizing the risk of voter fraud and ensuring the authenticity of every vote. Specifically designed for Android devices, it is widely accessible to a broad range of smartphone users. The primary goal is to enhance security and trust in the voting process by tackling common issues such as voter impersonation and fraud.

Utilizing advanced algorithms, the Android Face Voting System accurately captures and verifies each voter's facial features. When a user wants to cast a vote, the system uses the device's camera to capture their facial image, processes it to extract distinctive features, and matches these features against a pre-registered database. This process ensures that only authorized individuals can vote and prevents multiple voting attempts by the same person. Security is a cornerstone of this system. All biometric data is encrypted and securely stored to protect voter information from unauthorized access and potential breaches. This robust security framework not only safeguards personal data but also enhances the overall integrity of the democratic process. By ensuring that votes are cast by legitimate voters and are accurately counted, the system fosters greater trust among the electorate

.

II. MOTIVATION OF THE PROJECT

The necessity to address the vulnerabilities and inefficiencies in traditional voting methods, such as ballot papers and electronic voting machines (EVMs), has driven the need for innovative solutions. These conventional methods often face issues related to voter fraud, identity theft, and low voter turnout, posing significant challenges to the integrity and efficacy of the electoral process.

Traditional voting methods, while historically foundational to democratic processes, are increasingly susceptible to various forms of electoral fraud. Ballot papers, for instance, can be tampered with, misplaced, or even duplicated, leading to inaccuracies in vote counts. Cases of voter impersonation, where individuals cast votes using the identities ofothers, further compromise the legitimacy of election results. EVMs, although a step towards modernizing voting, are not immune to tampering and hacking. Ensuring that each vote is genuinely cast by an eligible voter and accurately recorded remains a persistent challenge. These vulnerabilities undermine public confidence in the electoral system, necessitating more secure and reliable methods of voter authentication and vote recording.

Another critical issue plaguing traditional voting methods is low voter turnout. Many eligible voters find it inconvenient to travel to polling stations, wait in long lines, or navigate complex voting procedures. This is particularly true for individuals with disabilities, the elderly, and those living in remote or rural areas. Additionally, the time constraints of modern life, where work and personal commitments often take precedence,discourage people from participating in elections. This lack of convenience leads to disengagement and apathy among potential voters, weakening the democratic process.

Abbreviations and Acronyms

- **AVS** - Android Voting System
- **FR** - Facial Recognition
- **AI** - Artificial Intelligence
- **DB** - Database
- **GUI** - Graphical User Interface
- **SDK** - Software Development Kit

Equations

The Microsoft Cognitive Face API is a powerful tool for face recognition, enabling robust and efficient face identification and verification processes. By leveraging this API, the Android Face Voting System can effectively enhance security and accuracy in the voting process. Below is a step-by-step explanation of how the Microsoft Cognitive Face API can be utilized for face recognition in the Android Face Voting System.

 Face Detection

- **API Call**: Detect
- **Function**: This function detects human faces in an image and returns the face locations along with their unique face IDs.
- **Equation**: DetectedFaces=FaceAPI.Detect(image)\text{DetectedFaces} = \text{FaceAPI.Detect(image)}DetectedFaces=FaceAPI.Detect(image)

 Face Registration

- **API Call**: Add Face to Face List
- **Function**: Adds a detected face to a predefined face list for later comparison.
- **Equation**: FaceList←FaceAPI.AddFace(faceId, faceListId)\text{FaceList} \leftarrow \text{FaceAPI.AddFace(faceId, faceListId)}FaceList←FaceAPI.AddFace(faceId, faceListId)

 Feature Extraction

- **API Call**: Embedded within Detect
- **Function**: Extracts unique facial features and encodes them as face embeddings (vectors).
- **Equation**: FaceEmbedding=FaceAPI.ExtractFeatures(faceId)\text{FaceEmbedding} = \text{FaceAPI.ExtractFeatures(faceId)}FaceEmbedding=FaceAPI.ExtractFeatures(faceId)

 Face Verification

API Call: Verify

- **Function**: Compares a detected face's features with a stored face's features to verify identity.
- **Equation**: VerificationResult=FaceAPI.Verify(faceId1, faceId2)\text{VerificationResult} = \text{FaceAPI.Verify(faceId1, faceId2)}VerificationResult=FaceAPI.Verify(faceId1, faceId2) where VerificationResult={Trueif match foundFalseotherwise\text{VerificationResult} = \begin{cases} \text{True} & \text{if match found} \\ \text{False} & \text{otherwise} \end{cases}VerificationResult={TrueFalseif match foundotherwise}

Thresholding for Authentication:
Authenticated={Trueif Similarity Score≥TFalseotherwise\text{Authenticated} = \begin{cases} \text{True} & \text{if } \text{Similarity Score} \geq T \\ \text{False} & \text{otherwise} \end{cases}Authenticated={TrueFalseif Similarity Score≥Totherwise

where TTT is the predefined threshold for successful authentication.

II. PROPOSED WORK

We have come up this cool Android voting system that keeps things super secure for voters when casting their votes. The way the server is set up in our system, it's impossible anyone to mess with the votes. Our goal is to make the electoral system more transparent and reliable with this Android voting system. Plus, we're using Microsoft Cognitive Face Recognizer API to verify the voters' faces.

Defining the modules and their functionalities:

Let's talk about the different parts of this project and what they do:

- Admin
- User

In the Android voting system, leveraging face recognition technology is crucial for accurate and reliable voter identification. Feature extraction plays a key role in this process, as it involves raw facial data to extract the most relevant information necessary for distinguishing one face from another. This step is essential because it directly affects the performance of the subsequent recognition algorithms. All in all, feature extraction is a critical aspect of implementing face recognition in the Android voting system. By efficiently processing raw facial data, it ensures that voter identification is accurate and reliable. Incorporating advanced Azure Services for feature extraction can significantly enhance the overall effectiveness of the system. For ensuring authentication, we are using Microsoft Azure Services to ID voters based on their facial features. We keep all that face data safe and make sure nobody can impersonate others or vote multiple times.

Voter Security and Privacy are our top priorities:

- **Data Encryption:** Protects personal and biometric data.
- **Secure Processing:** Makes sure all data processing is safe from breaches.
- **Privacy Compliance:** Follows regulations to keep voter info confidential.

We want to make voting easy for everyone :

- **Mobile Integration:** Vote from your Android phone.
- **User-Friendly Interface:** Simple and easy for everyone to use, regardless of age.
- **Accessibility Features:** Helps voters with disabilities participate too.

Making sure everything is fair and honest:

- Secure tech so votes can't be changed once cast.
- Shows a clear record of the voting process.
- Counts every vote accurately.

Increasing voter involvement:

- Mobile app voting increases convenience.
- Sends reminders about elections.
- Provides candidate info for an informed decision.

Tracking votes in real-time:

- **Live Vote Tracking:** Know how many votes are coming in.
- **Data Analytics:** See trends in voting behavior.
- **Reporting Tools:** Detailed reports on election results.

III. WORKING

An Android voting system using face recognition leverages facial recognition technology to authenticate voters and ensure secure and fair elections. Here's a high-level overview of how such a system would work:

Key Components

- **Mobile Application**: An Android app that voters use to cast their votes.
- **Face Recognition Technology**: Software integrated into the app to authenticate the identity of voters.
- **Backend Server**: Manages voter data, processes votes, and stores results securely.
- **Database**: Stores voter information, face recognition data, and election results.
- **Security Measures**: Encryption, secure communication protocols, and anti-tampering measures to ensure data integrity.

Workflow

1. Voter Registration :

- **Voter Enrollment**: Voters register themselves in the system by providing personal information and capturing their facial image through the app's camera.
- **Face Recognition Model Training**: The captured facial images are processed and used to train the facial recognition model, which generates unique facial features and stores them in the database.

2. Voter Authentication

- **Login:** On election day, voters log into the app using their credentials (e.g., voter ID).
- **Face Verification:** The app prompts voters to take a live photo. This photo is compared against the stored facial features to verify the voter's identity.
- **Liveness Detection:** To prevent spoofing (using a photo), the app employs liveness detection techniques, such as requiring blinking, head movement, or using infrared sensors to ensure the presence of a real person.

3. Voting Process

- **Ballot Display:** Once authenticated, the app displays the ballot with available candidates or options.
- **Vote Casting:** Voters select their choices and submit their vote. The app confirms the submission and encrypts the vote before sending it to the backend server.

4. Vote Storage and Counting

- **Secure Transmission:** Votes are transmitted to the backend server over a secure, encrypted channel.
- **Storage:** The server stores the encrypted votes in a secure database.
- **Counting:** Once voting is complete, the votes are decrypted and counted. The results are then processed and stored.

Security Measures

- **Encryption**: Both at rest and in transit, ensuring that sensitive data is protected.
- **Authentication**: Multi-factor authentication (MFA) for additional security.
- **Audit Logs:** Detailed logs of all activities for transparency and auditing purposes.
- **Tamper Detection:** Mechanisms to detect and respond to any tampering attempts.

Implementation Technologies

- **Android SDK:** For developing the mobile application.
- **Face Recognition Libraries:** Libraries like OpenCV, TensorFlow, or specialized services like Amazon Rekognition or Microsoft Azure Face API.
- **Backend Technologies:** Node.js, Django, or Spring Boot for the server-side application.
- **Database:** Fire Base database for storing data.
- **Encryption Protocols:** SSL/TLS for secure data transmission.

Example Workflow

1. **Registration Phase:**
    - User opens the app and fills in personal details.
    - The app captures a facial image.
    - The image is processed, and facial features are extracted and stored.
2. **Voting Phase:**
    - User logs in and is prompted to verify their identity with a live selfie.
    - The app performs face recognition and liveness detection.
    - Upon successful authentication, the user accesses the ballot.
    - User casts their vote, which is encrypted and sent to the server.
3. **Post-Voting Phase:**
    - Votes are securely stored and eventually decrypted for counting.
    - Results are calculated and published securely.

Facial Recognition Technologies

- **Feature Extraction:** Utilizes algorithms like Histogram of Oriented Gradients (HOG), Convolutional Neural Networks (CNNs), or Deep Face for extracting unique facial features.
- **Face Matching:** Compares live facial features against stored templates using metrics such as Euclidean distance or cosine similarity.

- **Accuracy Improvement:** Techniques like transfer learning on pre-trained models (e.g., VGGFace, FaceNet) to enhance recognition accuracy.

IV. RESULTS AND DISCUSSION

4.1. Admin:

Fig : 01

Fig:02

Fig:01

- **Login Details Entry:** The admin login page allows administrators to enter their user ID and password to access the system.
- **User Selection:** Admins can choose their role (Admin) before logging in, with an option for users to register if they don't have an account

Fig:02

- **Election Setup:** Configuring election parameters such as start and end times, eligible voter lists, and election rules.
- **Candidate Management:** Adding, editing, or removing candidates or options on the ballot.
- **Voter Enrollment Oversight:** Verifying and approving voter registrations, including facial image verification and data validation.

Post-Election Tasks

- **Vote Counting and Verification**: Ensuring that all votes are accurately counted, verifying the integrity of the results, and handling any recounts if necessary.
- **Result Publication:** Publishing the election results in a transparent and accessible manner.

Fig : 03

Fig 03: Poll Creating Screen

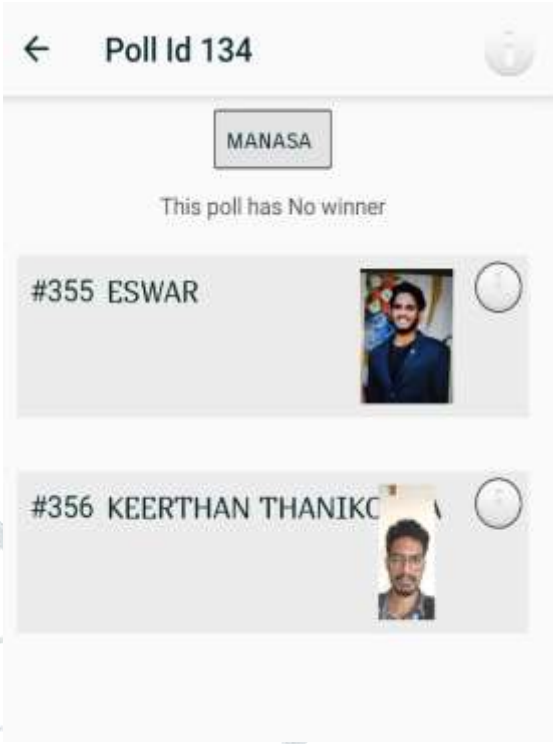

Fig : 04

Fig 04: Contestants

Fig:03

- **Name :** It specifies the name of the current poll.
- **Description :** It specifies the descriptive part of the poll which we are going to create such as type of election, etc.
- **Start Date** : It specifies the starting date of the poll.
- **Start time** : It specifies the time at which we start the poll.
- **End Date** : It specifies the ending date of the poll.
- **End time** : It specifies the time at which the polls end.
- **Result Date** : It specifies the date on which the results will released.

Fig:04

- This figure shows the candidates name and details who are participated in the poll and also shows who is the winner of that poll.

Creating a poll in an Android voting system involves designing the user interface (UI), implementing backend functionality, and adding optional features for enhanced usability and security. Face recognition is now commonly used to control access in many areas, such as public security, everyday life, the military, and banks. It uses computer technology to identify people by their facial features.

The facial recognition system matches a person's face captured in a digital image against a database of stored faces, typically used for user authentication and ID verification services. These days, face recognition is seen as a dependable and precise way to control access. It's often used alongside other biometric methods for added security. Facial recognition involves the process of identifying a person's face from a collection of stored face samples within the system.

This is achieved by converting images into digital binary representations and extracting relevant features using various techniques. Here we have used Microsoft Azure Services which has emerged as a prominent approach in the field of face recognition. As discussed in research, face recognition entails the automated detection and recognition of objects such as human faces, animals, or other entities using computerbased.

User Interface (UI)

1. **Poll Creation Screen:** Includes fields for the poll title, description, multiple options, optional date pickers for start and end dates, and a create button.
2. **Voting Screen:** Displays the poll title, description, options as radio buttons or checkboxes, and a vote button.
3. **Results Screen:** Shows the poll title, description, and a graphical representation of the poll results.

Backend Implementation

1. **Data Model:** Define Poll and Option classes to represent poll data.**Database:** Use Firebase to store poll and option data.
2. **Poll Creation Logic:** Validate inputs, and save the poll to the database.
3. **Voting Logic**: Retrieve the poll, increment the vote count for the selected option, and update the database.
4. **Results Display**: Retrieve poll data, calculate results, and display them graphically.

4.2 User :

Fig : User Registration Page

Fig : Poll Details Page

Fig : Face Authentication Page

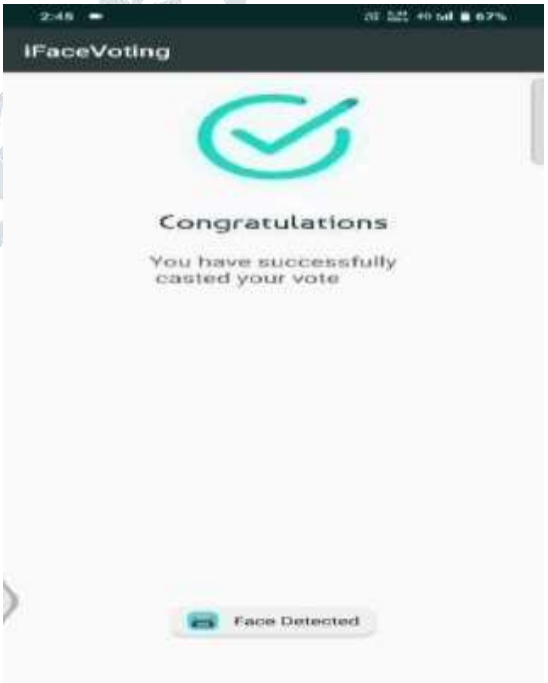

Fig : Successfully Voted Page

The registration page for our Android voting system leverages face recognition technology for secure voter authentication. Designed for simplicity and security, it ensures accessible enrollment and enhances the integrity of electoral processes, fostering trust and inclusivity in democratic participation.

1. Register or Log In

- **Register**: If you're a new user, register by entering your details such as name, date of birth, and a valid ID.
- **Log In**: If you're already registered, log in using your username and password.

2. Face Recognition Authentication

- **Initiate Authentication**: The app will prompt you to initiate face recognition for authentication.
- **Scan Your Face**: Position your face within the camera frame. Ensure good lighting and follow any instructions provided.
- **Verification**: The Microsoft Cognitive API will verify your identity based on your registered profile.

3. Select the Election

- **Choose Election**: Tap on the election you wish to participate in from the list of available options.

4. View Candidates or Options

- **Review Choices**: The app will display the candidates or voting options for the selected election. Scroll through to review details about each option.

5. Cast Your Vote

- **Make Selection**: Tap on your preferred candidate or voting option to make your choice. Some apps may require you to confirm your selection before proceeding.

6. Submit Your Vote

- **Confirm and Submit**: After making your selection, find and tap the "Submit Vote" button. This action will finalize your vote.

V. FUTURE SCOPE:

Android Voting System Using Face Recognition" encompasses the development of an secure voting system that leverages face recognition for voter authentication, ballot processing, and result generation. It includes secure and accessible voting methods, real-time data analysis, and a userfriendly interface. The system aims to enhance the security, efficiency, and transparency of the voting process, catering to both traditional and remote voting scenarios.This project aims to revolutionize the voting process by integrating advanced facial recognition technology for voter authentication. This system ensures that each vote is cast by a legitimate voter, thus enhancing the integrity and security of elections. By leveraging Microsoft Cognitive API for face recognition, the project employs Java programming language for backend processing, XML for frontend design, and Firebase as the database. This combination ensures a robust, scalable, and user-friendly voting solution.

1. Secure Voter Authentication : Utilizes Microsoft Cognitive API to accurately verify voter identities, preventing impersonation and fraud.
2. Efficient Ballot Processing : Processes votes instantly, providing real-time updates and analytics to election officials. Automates ballot counting and validation, minimizing human errors and ensuring accuracy.
3. Result Generation and Transparency : Compiles and displays election results in real-time, allowing for prompt announcement and reducing waiting times. Generates detailed reports and audit trails, enhancing the transparency and trustworthiness of the election process.
4. Accessibility and Inclusivity : Designed with XML to create an intuitive and easy-to-navigate user interface, accessible to all voters, including those with disabilities.
5. Support for Traditional and Remote Voting : Accommodates both in-person and remote voting, allowing voters to cast their ballots from anywhere securely. Enables offline voting where internet access is limited, with data synchronization once connectivity is restored.
6. Enhanced Security Measures : Ensures all data transmitted between the voter's device and central servers is encrypted, protecting against data breaches. Conducts periodic security assessments to identify and mitigate potential vulnerabilities. By focusing on these aspects, the "Android Voting System Using Face Recognition" aims to deliver a secure, efficient, and transparent voting experience, addressing the needs of modern electoral processes and building public confidence in digital voting technologies.

VI. CONCLUSION :

The development of an Android voting system that leverages advanced facial recognition technology enhances the security, efficiency, and transparency of the voting process. The system includes a comprehensive registration feature that captures frontal facial images of new users, ensuring accurate voter authentication. User details must be verified by an admin, maintaining a high level of security and integrity. Once a new user is registered, the facial recognition models are retrained by the admin to incorporate the new data, ensuring the system remains up-to-date and effective their vote.

This prevents duplicate voting, as the system does not permit anyone to vote more than once. By using Microsoft Cognitive Services Face API for facial recognition, the system benefits from reliable and accurate authentication capabilities. The use of Android UI via Java provides a user-friendly interface, making the voting process accessible and straightforward for all users. Real-time data analysis and transparent result

generation further enhance the system's functionality, providing immediate insights and fostering public trust. The integration of secure and accessible voting methods ensures that the system can cater to a wide range of voters, including those in remote locations. In conclusion, the Android Voting System Using Face Recognition represents a significant advancement in digital voting technology. By combining facial recognition with secure data handling and a user-friendly interface, the system not only improves the security and efficiency of elections but also enhances voter confidence and participation. Continuous improvements and the incorporation of emerging technologies will further solidify the system's role in modernizing the voting process.

VII. REFERENCES

[1] Kohno, T., Stubblefield, A., Rubin, A. D., & Wallach, D. S. (n.d.). Analysis of an electronic voting system. IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004. doi:10.1109/secpri.2004.1301313 [2] D. Chaum, "Secret-ballot receipts: True voter-verifiable elections," in IEEE Security & Privacy, vol. 2, no. 1, pp. 38- 47, Jan.-Feb. 2004, doi: 10.1109/MSECP.2004.1264852. [3] Evans, D., & Paul, N. (2004). Election security: perception and reality. IEEE Security & Privacy Magazine, 2(1), 24–31. doi:10.1109/msecp.2004.1264850 [4] Jefferson, D., Rubin, A. D., Simons, B., & Wagner, D. (2004). Analyzing internet voting security. Communications of the ACM, 47(10), 59. doi:10.1145/1022594.1022624 [5] Evans, D., & Paul, N. (2004). Election security: perception and reality. IEEE Security & Privacy Magazine,2(1),24–31.doi:10.1109/msecp.2004.12