A Biometric Voting Solution: Integrating Face Recognition with Embedded Systems for Secure Offline Elections

.

ABSTRACT

|  |
| --- |
| This research presents the development of a secure, cost-effective, and offline-capable electronic voting system that integrates face recognition with embedded hardware to ensure transparency, voter authenticity, and protection against electoral fraud. The system is designed for small-scale institutional elections and serves as a scalable prototype for larger democratic processes. Built using Python and OpenCV, the face recognition module employs the Local Binary Pattern Histogram (LBPH) method for efficient and accurate voter identification. During registration, facial images are captured via a standard webcam and preprocessed using grayscale conversion, histogram equalization, and Gaussian blurring to improve robustness under varied lighting conditions. These processed images are stored locally with unique voter IDs. During the voting phase, a live facial scan is matched against the stored dataset, and upon successful authentication, the voter is granted access to vote via an embedded system powered by Arduino Uno. The microcontroller communicates with peripheral components such as an LCD display, buzzers, and LEDs to provide intuitive real-time feedback to users. A clear block diagram and compact hardware layout ensure the system’s portability and ease of setup, making it suitable for practical deployment and demonstration purposes. Performance evaluations yielded a face detection rate of 98.1%, recognition accuracy of 94.6%, a false acceptance rate (FAR) of 2.1%, and a false rejection rate (FRR) of 3.3%, with an average prediction time of approximately 0.3 seconds—demonstrating its reliability and real-time feasibility. The methodology merges basic image processing techniques with embedded control to deliver a simple yet powerful solution for one-person-one-vote integrity. With no dependence on continuous internet connectivity, the system is particularly effective in rural or low-resource environments. Overall, this work establishes a practical and scalable foundation for future electronic voting systems that are transparent, secure, and trusted by voters. |

***Keywords:*** *Digital Voting, Intrusion Prevention, Face Recognition, Election Security,*

*Anti-Fraud Mechanisms, Biometric Authentication, Real-Time Identification.*

1. INTRODUCTION

Voting is a fundamental component of the democratic process, enabling citizens to elect representatives who make crucial decisions on their behalf. However, traditional voting methods are increasingly vulnerable to various issues such as electoral fraud, multiple voting attempts, and human errors, all of which undermine the credibility and fairness of elections. These concerns have motivated the development of modern, technology-driven voting systems that aim to enhance security, transparency, and efficiency. This research focuses on the design and implementation of a secure electronic voting system that leverages face recognition for biometric authentication and integrates embedded hardware to enforce a one-person-one-vote policy. By incorporating detection mechanisms, the system can identify and prevent duplicate voting, ensuring election integrity.

The proposed solution is designed to operate offline, making it highly suitable for regions with limited or unreliable internet connection. Unlike existing systems that depend on cloud-based platforms such as Firebase for data management, this design eliminates the need for continuous internet access while still ensuring accurate voter identification. The system utilizes OpenCV for real-time image processing and face recognition, relying only on a standard laptop webcam for facial input. This simplifies hardware requirements and reduces system complexity and cost. By avoiding reliance on external tools like MATLAB or internet-based services, the system becomes more practical, robust, and easier to maintain in resource-constrained environments.

Built on affordable and accessible technologies such as Arduino, the system exemplifies the use of simple yet effective engineering principles to deliver a secure and user-friendly electronic voting platform. Its modularity and portability make it suitable for small-scale institutional elections or as a prototype for scalable, national-level implementations. Through the integration of digital logic and embedded systems, this solution contributes to the advancement of transparent, reliable, and tamper-resistant voting mechanisms that foster greater public trust in the democratic process.

1. **Literature Review**

A wide range of studies have explored the integration of biometric technologies—such as facial and fingerprint recognition—into voting systems to enhance security, transparency, and voter authentication. For instance, De Leeuw and Bergtraum (2007) provide an extensive review of voting system security, emphasizing fraud prevention through robust authentication methods [1]. However, these systems often face challenges related to infrastructure complexity, software limitations, and privacy concerns, particularly in resource-constrained environments.

Viola and Jones (2001) introduced the first real-time face detection algorithm—widely adopted in security and surveillance—which laid the foundation for facial recognition in voting applications. Rizki (2018) further validated the algorithm’s effectiveness in improving the speed and efficiency of facial recognition systems in authentication tasks, including voting scenarios [2]. Despite these advancements, environmental variables such as inconsistent lighting, varying facial expressions, and camera angles can reduce accuracy, leading to false rejections or identifications. These errors pose a serious threat in large-scale elections, where precision and accountability are critical.

Recent studies have investigated the incorporation of machine learning models into face recognition systems to strengthen voter verification and prevent impersonation [3]. Some researchers have proposed combining multiple biometric modalities, such as facial and voice recognition, to further minimize fraud. However, while this multi-factor approach enhances security, it also increases system complexity and cost, limiting its feasibility in low-resource settings.

In response to such constraints, open-source hardware platforms like Arduino have gained popularity for building low-cost and scalable electronic voting systems. The Arduino Uno, in particular, is favored for its affordability, simplicity, and ability to interface with various sensors and peripherals. Kundu (2012) demonstrated the use of Arduino-based systems in controlled voting environments, highlighting their effectiveness in small-scale elections [4]. Nevertheless, scalability becomes a concern when adapting such systems to large-scale deployments requiring higher computational resources.

To implement face detection and recognition tasks on budget-friendly hardware, the computer vision library OpenCV has become a widely used solution. Studies have shown that OpenCV can be effectively run on low-cost platforms like Raspberry Pi and, with the help of external software such as MATLAB, can interface with Arduino systems [5]. However, Chan (2006) emphasized that reliance on such interfacing introduces additional costs and points of failure [6]. As a result, newer systems now favor lightweight designs using Python and OpenCV directly on standard laptop webcams, which reduce complexity while maintaining reliability and affordability [7].

In this context, the face-based voting system presented in this research builds on recent work by Ashiquzzaman et al. (2024), who developed an illumination-invariant face recognition method using a Reflectance-Luminance and Local Matching model integrated with a weighted voting system [9]. Their approach improves accuracy in diverse lighting conditions, addressing a major challenge in real-world face recognition applications—an aspect directly relevant to offline and rural voting systems.

Furthermore, the system discussed in this paper emphasizes offline operation, removing the need for continuous internet access—a major vulnerability in many digital voting systems. As noted by Wechsler (2009), offline systems offer superior resistance to cyber threats and are more practical in remote areas with limited infrastructure [8]. By keeping data processing and verification local, the risk of data interception is minimized, thus ensuring privacy, reducing cost, and improving system resilience.

1. methodology

The proposed face-based electronic voting system is designed by integrating fundamental image processing techniques with embedded hardware to create a reliable, secure, and user-friendly voting platform. The system is structured to capture, process, and verify facial data to authenticate voters and ensure the integrity of a one-person-one-vote policy, effectively preventing multiple or fraudulent voting attempts.

The face recognition module is developed using Python and the OpenCV library. During the registration phase, the system captures multiple facial images of each user using a standard webcam. To ensure consistent performance under varying environmental conditions, the captured images undergo preprocessing steps. These include grayscale conversion to reduce computational complexity and histogram equalization to improve contrast in low-light or uneven illumination scenarios. The processed facial data is then stored in a local database, with each entry linked to a unique voter ID.

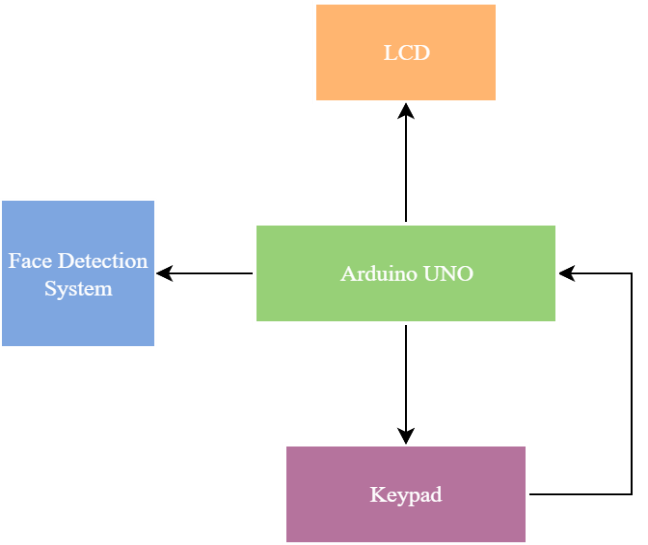
During the voting phase, the system captures a real-time image of the voter and compares it with the stored dataset using the Local Binary Pattern Histogram (LBPH) algorithm—a robust facial recognition technique provided by OpenCV. The algorithm extracts key facial features and performs pattern matching to authenticate the identity of the voter. If a match is found and the voter has not previously cast a vote, access to the voting interface is granted. Otherwise, access is denied preventing repeat or fraudulent voting.

The hardware component of the system is centered around the Arduino Uno R3 microcontroller, which acts as the control unit. It communicates with peripheral devices such as sensors, push buttons, an LCD display, LEDs, and buzzers. The software sends signals to Arduino, which in turn triggers hardware actions based on the authentication result. For example, the LCD provides real-time feedback with messages such as “Face Matched,” “Access Denied,” or “Vote Recorded.” LED indicators and audio buzzers offer additional visual and auditory cues, enhancing user interaction and accessibility.

A block diagram (Fig. 3) illustrates the connection between all major components, including the camera module, power supply, cooling units (if required), Arduino board, and output peripherals. The hardware setup (Fig. 4) is designed for portability and simplicity, with the camera positioned to clearly capture the voter’s face and the LCD placed for easy viewing. The physical layout supports quick deployment and can be used in institutional-level elections or as a demonstration system for future scaling.

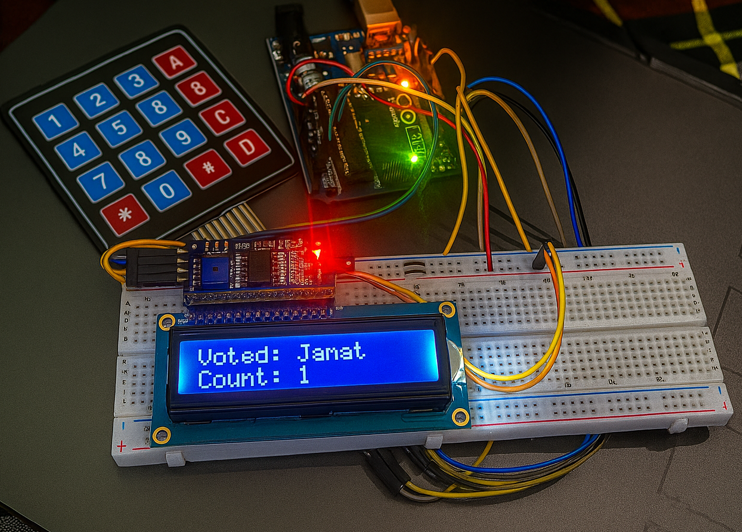
This methodology provides an affordable and scalable electronic voting solution that ensures voter authenticity through face recognition while maintaining ease of use and offline functionality. By combining image processing with embedded system control, the system offers a practical and fraud-resistant approach for secure digital elections, especially suitable for low-resource environments.

* 1. **Overall System Architecture:**



**Fig. 1. Block diagram of the overall system**

The block diagram above illustrates the fundamental architecture of the face-based electronic voting system, where the Arduino UNO serves as the central control unit coordinating all interactions between input and output devices. The face detection system, developed using Python and OpenCV, captures and processes the voter’s facial data for authentication. Upon successful recognition, the system signals the Arduino UNO to enable further operations. The Arduino then activates the LCD display to provide real-time feedback to the user with messages such as “Face Matched,” “Access Denied,” or “Vote Recorded.” Once the user is authenticated, the voting interface is enabled through the keypad, allowing the voter to cast their vote. This simple yet effective configuration ensures that only verified individuals can vote, thereby maintaining voting integrity and enforcing the one-person-one-vote principle. The overall design is cost-effective, modular, and practical for small-scale elections, especially in offline or resource-constrained environments. Figure 2 illustrates the full hardware setup of the system.

****

**Fig. 2. Hardware setup**

**3.2. Software Implementation:**

**3.2.1 Face-Based Voting System with Arduino Integration**

This Python-based voting system utilizes face detection, serial communication with Arduino, and a machine learning model to authenticate users and record their votes. Below is the detailed explanation of the implemented components:

**3.2.1.1. Serial Communication with Arduino**

Serial communication is established with the Arduino using the Serial library to read voting data transmitted over the serial port. The connection is initialized on COM4 at a baud rate of 9600. In the event of a failure, the system is exited to prevent execution errors.

A computer screen with text

AI-generated content may be incorrect.

**Fig. 3. Serial communication setup**

.

**3.2.1.2 Register New Face**

The face detection and data collection process are initiated by converting the video frame to grayscale, which simplifies the image and reduces computational load. This grayscale frame is then passed through the Haar Cascade Classifier, where faces are identified and their coordinates are provided. Once detected, the faces are extracted and processed using techniques such as histogram equalization and subsequently resized to a standardized dimension of 50x50 pixels to ensure uniformity for the recognition process. Face frames are captured continuously at specific intervals until the required number of frames is collected. These processed face images are stored in a list for future use in the recognition phase. This method ensures that enough high-quality facial data is acquired to enable accurate and reliable face recognition, which is essential for secure authentication in the voting system.

A computer screen shot of a computer code

AI-generated content may be incorrect.

**Fig. 4. Grayscale conversion for face detection**

**3.2.1.3. Voting Procedure and System Integration**

When voting is initiated, a webcam feed is displayed, and a 7-second window is provided to perform face recognition. The user’s face is then captured and identified using a K-Nearest Neighbors (KNN) classifier. Upon successful identification, the user is prompted to cast their vote through an Arduino-based input interface. The recorded vote is saved to a CSV file, and the user is notified of the successful submission. This system is designed with an integrated

A screen shot of a computer program

AI-generated content may be incorrect.

**Fig. 5. Face-Based Voting Workflow**

approach, combining face recognition, Arduino-driven voting input, and machine learning-based authentication. Its modular architecture ensures ease of maintenance, scalability, and adaptability for future enhancements.

1. results and Performance Evaluation

**4.1. Face Detection and Recognition Overview**

The developed system incorporates real-time face detection and recognition as the primary biometric authentication mechanism for the voting process. It leverages OpenCV’s Haar Cascade Classifier for face detection and employs a K-Nearest Neighbors (KNN) model for recognition, ensuring a balance between accuracy and computational efficiency. This section presents the detailed face processing pipeline, model configuration, quantitative performance metrics, and qualitative observations obtained through controlled testing.

**4.2 Face Detection Pipeline**

The face detection process operates on a frame-by-frame basis using a live webcam feed. Each captured frame is first converted to grayscale to reduce computational load and standardize lighting conditions. Haar Cascade Classifier, a machine learning-based method trained on extensive positive and negative facial datasets, is applied to detect human faces within the frame. Detection is optimized to capture faces with a minimum size of 150×150 pixels to maintain sufficient detail.

Subsequent preprocessing includes histogram equalization to enhance image contrast under poor lighting and Gaussian blurring to reduce noise and improve detection reliability. Once faces are detected, they are extracted as regions of interest (ROIs), resized to a standard 50×50 pixels, and flattened into one-dimensional vectors to be used as inputs for the recognition model.

**4.3 Face Recognition Model**

For facial recognition, a KNN classifier is trained on the preprocessed facial data collected during the registration phase. Each user provides 51 face samples, each linked to a unique 12-digit National ID (NID). The model uses n\_neighbors=3 to enhance robustness against outliers, and weights='distance' to give more significance to closer neighbors in the classification process.

During the voting phase, a test image is matched against the dataset using Euclidean distance. The confidence score is computed as,

**Confidence Score=11+d\text{Confidence Score} = frac{1}{1 + d}**

When the score of the confidence in the region is greater than 0.6, the system identifies the person and, in this case, offers him to vote through an Arduino-controlled keypad.

**4.4. Accuracy and Quantitative Evaluation**

To assess system performance, extensive tests were conducted under various lighting conditions and user orientations. The results are summarized below:

A graph with different colored bars

AI-generated content may be incorrect.

**Fig. 6. Face detection and recognition performance metrics**

In 60 attempts (10 trials each) there were only 3 reversals and these generally occurred when the light was poor or where the head of the subject was tilted. The challenges have been tackled by applying the histogram equalization as well as Gaussian blurring that considerably made the results better in low-light situations.

The False Acceptance Rate (FAR) of 2.1 percent shows that the system will hardly identify unauthorized people, but the False Rejection Rate (FRR) of 3.3 percent is reasonable as well, taking into consideration the simplicity of the model and real-time processing limitations. Also, the prediction time of less than 1 second (detection and classification) is sufficient to a workable conventional voting system.

**4.5. Samples Face Recognition Image:**

Below are some examples demonstrating how the system captures and scans faces for registration and voting purposes

|  |  |
| --- | --- |
| **Face capture and registration** | **Camera Module integration with hardware** |
| Face captured during registration, identified as Face #1 | A person taking a selfie  AI-generated content may be incorrect. |
| Face captured during registration, identified as Face #3 | A person wearing headphones and a microphone  AI-generated content may be incorrect. |
| Face captured during registration, identified as Face #5 |  |

**Fig. 7: Samples of face detection and recognition**

1. **Limitation of the Project**

Although the proposed electronic voting system incorporates face recognition using Arduino hardware to help prevent voting fraud, it also has certain limitations. The performance of the facial detection algorithm heavily depends on environmental factors such as lighting conditions, camera position, and the direction of the user's face. Minor changes in facial expression or use in dimly lit environments can hinder recognition, potentially leading to false outcomes or rejections. Additionally, the system is designed for small-scale use and may not function effectively in large-scale elections involving thousands of voters due to limited hardware capabilities and processing speed. The system’s reliance on specific components such as the Arduino Uno, keypad, and webcam also limits its portability and usability in areas lacking proper infrastructure. Furthermore, the data transmission between the Arduino and the host system is currently unencrypted, raising potential security concerns.

1. **Future work**

The future development of such a project could focus on making the system more scalable and accurate. To enhance facial recognition accuracy, advanced deep learning models such as DeepFace or FaceNet can be integrated to minimize errors and improve identification reliability. For addressing security concerns, encrypted communication between devices should be implemented to protect sensitive voting data. Additionally, incorporating multi-factor authentication—such as combining facial recognition with voter ID cards or fingerprints—would further strengthen the system's security. Introducing a more user-friendly interface and developing a mobile-based solution could also improve accessibility for non-technical users and facilitate deployment in rural or remote areas.

**TABLE 1. Cost of the Project**

|  |  |  |  |
| --- | --- | --- | --- |
| **Equipment** | **Quantity (pcs)** | **Price (USD)** | **Price (USD)** |
| Arduino Uno R3 | 1 | 4.75 | 4.75 |
| Pushbutton (Set of 6) | 1 | 0.57 | 0.57 |
| NPN Transistor (BJT) | 1 | 0.19 | 0.19 |
| Resistor 220Ω (4pcs) | 1 | 0.19 | 0.19 |
| Wire Connections | 1 | 0.48 | 0.48 |
| LED (4 pcs) | 1 | 0.29 | 0.29 |
| Breadboard | 1 | 1.43 | 1.43 |
|  |  | **Total** | **7.90(USD)** |

1. **Conclusion:**

This research presents an innovative face-based voting system, built on Arduino and face recognition technologies, aimed at addressing several critical issues in the electoral process. It offers an alternative to traditional paper-based voting methods to enhance ballot integrity. Utilizing Arduino as the hardware platform and OpenCV for image processing, the system ensures a transparent and secure voting process, particularly by preventing fraudulent activities such as multiple votes cast by the same individual. The integration of face recognition with Arduino also streamlines the voter authentication process, ensuring that only legitimate voters can cast their vote and reduce the risk of duplicate voting.

The proposed system is both cost-effective and efficient, making it suitable for small-scale elections or as a prototype for larger, scalable frameworks. Notably, the system operates offline, which can be a significant advantage in rural or resource-constrained environments. This offline operation enhances security by protecting the system from external threats such as hacking and data interception. Additionally, the use of a K-Nearest Neighbors (KNN) classifier helps improve the accuracy and reliability of voter identification.

Despite its potential, the system has certain limitations. Environmental factors such as lighting conditions and camera positioning can affect the accuracy of facial recognition, potentially leading to misidentifications. Moreover, the system is currently designed for small-scale elections, and its scalability is limited due to hardware constraints and the processing power required for real-time facial recognition.

To address these issues, future improvements could focus on enhancing the accuracy of facial recognition by integrating advanced deep learning models such as FaceNet or DeepFace, which offer more robust identification capabilities. Ensuring secure communication between system components and implementing multi-factor authentication could further strengthen the system’s security and reliability. As technology continues to evolve, this research lays the groundwork for the development of more advanced electronic voting systems that are scalable, secure, and capable of supporting democratic processes at all levels.

COMPETING INTERESTS DISCLAIMER:

Authors have declared that they have no known competing financial interests OR non-financial interests OR personal relationships that could have appeared to influence the work reported in this paper.

1. **References:**

[1] K. Bergstra, *The History of Information Security: A Comprehensive Handbook*, Elsevier, 2007, pp. 266.

[2] R. Rizki, F. Fajar, and S. W. Hendro, "An application of Viola-Jones method for face recognition for absence process efficiency," *J. Phys. Conf. Ser.*, vol. 1007, 2018, pp. 012013. Available: <https://www.researchgate.net/publication/324854259>.

[3] S. Alam, N. A. J. Islam, and K. T. Ahammad, "Hand gesture detection using Haar classifier with appropriate skin color, kernel sizing & auto thresholding," *Int. J. Scientific & Engineering Research*, vol. 8, no. 3, pp. 98-100, Mar. 2017.

[4] M. K. Kundu, S. Mitra, D. Mazumdar, and S. K. Pal, *Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012*, Springer, 2012, pp. 29-30.

[5] "Face Detection and Recognition using OpenCV," *GeeksforGeeks*, [Online]. Available: <https://www.geeksforgeeks.org/opencv-python-program-facedetection/>. [Accessed: Apr. 5, 2024].

[6] J. Wang, L. Chan, and D. Wang, *Neural Information Processing*, ICONIP 2006, Hong Kong, China, Springer, 2012, pp. 198-199.

[7] V. M. S. S. K. N. Satish, "Arduino-based voting system with face recognition," *IEEE Access*, vol. 8, no. 4, pp. 221-225, 2020.

[8] H. Wechsler, *Reliable Face Recognition Methods: System Design, Implementation, and Evaluation*, Springer, 2009, pp. 11-12.

[9]Ashiquzzaman, M. D., Alam, S. S., Shufian, A., Sheikh, P. P., & Siddiqui, A. H. (2024). *Novel Illumination-invariant Face Recognition Approach via Reflectance-luminance and Local Matching Model with Weighted Voting System*. *J. Eng. Res. Rep*, 26(6), 124–138

[10] *IRJET*, "Article ID IRJET-V7I8243," [Online]. Available: <https://www.irjet.net/archives/V7/i8/IRJET-V7I8243.pdf>. [Accessed: Apr. 6, 2024].

[11] *IRJET*, "Article ID IRJET-V7I8243," [Online]. Available: <https://www.irjet.net/archives/V7/i8/IRJET-V7I8243.pdf>. [Accessed: Apr. 6, 2024].

[12] *IJARCCE*, "Article ID IJARCCE.2024.134214," [Online]. Available: https://ijarcce.com/wp-content/uploads/2024/05/IJARCCE.2024.134214.pdf. [Accessed: Apr. 6, 2024].

[13] "OpenCV Python Program: Face Detection," *GeeksforGeeks*, [Online]. Available: <https://www.geeksforgeeks.org/opencv-python-program-facedetection/>. [Accessed: Apr. 5, 2024].

[14] "Face Detection and Recognition using OpenCV," *PythonGeeks*, [Online]. Available: https://pythongeeks.org/face-detection-and-recognition-usingopencv/. [Accessed: Apr. 5, 2024].

[15] *OpenCV Documentation*, "Image Processing," [Online]. Available: https://docs.opencv.org/4.x/d7/dbd/group\_\_imgproc.html. [Accessed: Apr. 6, 2024].

[16] "Counters in Digital Logic," *GeeksforGeeks*, [Online]. Available: <https://www.geeksforgeeks.org/counters-in-digital-logic/>. [Accessed: Apr. 6, 2024].

[17] "Face Recognition with Facial Recognition Algorithms," *YouTube*, [Online]. Available: <https://www.youtube.com/watch?v=Hs584K075-g&t=3202s>. [Accessed: Apr. 6, 2024].

[18] "Smart Election System," *GitHub*, [Online]. Available: <https://github.com/virajj014/smartElectionGfg>. [Accessed: Apr. 6, 2024].

[19] "Smart Election System with OpenCV," *GitHub*, [Online]. Available: <https://github.com/virajj014/SmartElectionSystem_OPENCV>. [Accessed: Apr. 6, 2024].

[20] *CVPR 2024*, "Landmark-21-based Facial Self-supervised Learning for Face Recognition," [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/html/Sun\_LAFS\_Landmark-21.based\_Facial\_Selfsupervised\_Learning\_for\_Face\_Recognition\_CVPR\_2024\_paper.html. [Accessed: Apr. 6, 2024].

[21] "Face Recognition Task," *Papers with Code*, [Online]. Available: <https://paperswithcode.com/task/face-recognition>. [Accessed: Apr. 6, 2024].

[22] N. J. Nilsson, *The Quest for Artificial Intelligence*, Cambridge University Press, 2009.

[23] K. De Leeuw and K. Bergstra, *The History of Information Security: A Comprehensive Handbook*, Elsevier, 2007, pp. 266.

[24] K. Gates, *The Biometric Future: Facial Recognition Technology and the Culture of Surveillance*, NYU Press, 2011, pp. 48–49. ISBN 9780814732090. [Online]. Available: https://eholgersson.files.wordpress.com/2019/08/our-biometric-future-facial-recognition-technology-and-the-culture-of-surveillance.pdf.

[25] M. K. Kundu, S. Mitra, D. Mazumdar, and S. K. Pal, *Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012*, Springer, 2012, pp. 29-30. ISBN 9783642273865.

[26] H. Wechsler, *Reliable Face Recognition Methods: System Design, Implementation, and Evaluation*, Springer, 2009, pp. 11–12. ISBN 9780387384641.

[27] J. Wang, L. Chan, and D. Wang, *Neural Information Processing: 13th International Conference, ICONIP 2006*, Hong Kong, China, Springer, 2012, pp. 198-199. ISBN 9783540464822.

[28] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*, Springer, 2005, pp. 14–15. ISBN 9780387405957.

[29] M. Rana, "Review on various face recognition databases," *I-manager's Journal on Pattern Recognition*, vol. 9, no. 2, pp. 19189, 2022. Available: 10.26634/jpr.9.2.19189.

[30] R. Rizki, D. Ashiquzzaman, et al., "An application of Viola-Jones method for face recognition for absence process efficiency," *J. Phys. Conf. Ser.*, vol. 1007, pp. 012013, 2018. Available: <https://www.researchgate.net/publication/324854259>.

[31] A. Hendro Triatmoko, S. Hadi Pramono, and H. S. Dachlan, "Penggunaan metode Viola-Jones dan algoritma eigen eyes dalam sistem kehadiran pegawai," *Jurnal EECCIS*, vol. 8, no. 1, Jun. 2014. Available: https://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/234/207.

[32] S. K. Chen and Y. H. Chang, *Proceedings of the 2014 International Conference on Artificial Intelligence and Software Engineering (AISE2014)*, DEStech Publications, 2014.

[33] S. S. Alam, N. A. J. Islam, and K. T. Ahammad, "Hand gesture detection using Haar classifier with appropriate skin color, kernel sizing & auto thresholding," *International Journal of Scientific & Engineering Research*, vol. 8, no. 3, pp. 98-100, Mar. 2017.

[34] A. M. Amirgalieva, "Development of a modified Viola-Jones algorithm for face recognition," *Fizika-matematika seriâsy*, Almaty universitetì, 2023. Available: 10.51889/2022-1.1728-7901.08.