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, specifically designed for rural and connectivity-limited areas, integrating 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 R3. 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 [5].

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 growing body of research has explored the integration of biometric technologies into electronic voting systems to enhance security, transparency, and voter authentication. De Leeuw and Bergstra [1] provided a foundational analysis of information security in digital systems, emphasizing the necessity of robust authentication mechanisms to mitigate electoral fraud. However, their study also noted infrastructural and implementation challenges, particularly in under-resourced environments.

Viola and Jones introduced a breakthrough in real-time face detection through their cascade classifier method, which remains widely used in security applications. Rizki et al. [2] applied this method to improve authentication processes in institutional attendance systems. Their findings demonstrated faster and more accurate face detection compared to manual identification; however, the system’s performance was limited under poor lighting and occlusion, raising concerns about its suitability for high-stakes applications like national elections.Alam et al. [3] explored enhancements to Haar cascade-based detection using skin color filtering, kernel sizing, and adaptive thresholding. Their system performed well in controlled environments but lacked robustness in dynamic, real-world settings. These findings suggest that additional mechanisms—such as adaptive preprocessing or environmental calibration—are necessary for reliable deployment.

Kundu et al. [4] evaluated the use of Arduino in small-scale electronic voting systems. Their work highlighted Arduino’s affordability and ease of integration with peripheral sensors, making it a promising candidate for low-budget, rural elections. However, they noted limitations in scalability and processing power, which restrict applicability to large-scale elections requiring real-time biometric verification. Rajashekar and Soni [5] developed a smart voting system using facial recognition on embedded platforms, showing that lightweight models could accurately identify users and prevent duplicate voting. While their prototype proved effective in lab conditions, the lack of evaluation under diverse environmental conditions (e.g., varying light or facial orientation) limits generalizability.

Wang, Chan, and Wang [6] emphasized that the use of external toolchains like MATLAB for hardware interfacing increases both cost and failure points in biometric systems. To overcome this, Gowda et al. [7] proposed a streamlined approach using OpenCV and Python to execute face detection on low-end laptops. Their design offered real-time processing without internet dependency, though it sacrificed advanced features like deep learning-based face matching due to hardware constraints.

Ashiquzzaman et al. [9] introduced a novel illumination-invariant face recognition approach based on reflectance-luminance separation and a local matching model. Their weighted voting algorithm significantly improved recognition accuracy under varied lighting, addressing one of the primary challenges in rural or outdoor polling environments. While effective, the model's computational demand may hinder its performance on low-power devices like Arduino.Wechsler [8] argued that offline biometric systems provide enhanced data privacy and resistance to cyber threats. This is particularly relevant for rural and remote areas with limited infrastructure, where offline operation eliminates dependency on cloud services and mitigates data interception risks.

Taken together, the reviewed literature underscores a clear trajectory toward more secure, low-cost, and decentralized voting solutions. However, most studies either fall short on scalability, environmental robustness, or affordability. This research addresses these gaps by designing an offline, face-based voting system tailored for rural deployment—combining cost-effective embedded hardware, lightweight machine vision, and illumination-resilient recognition models.

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.

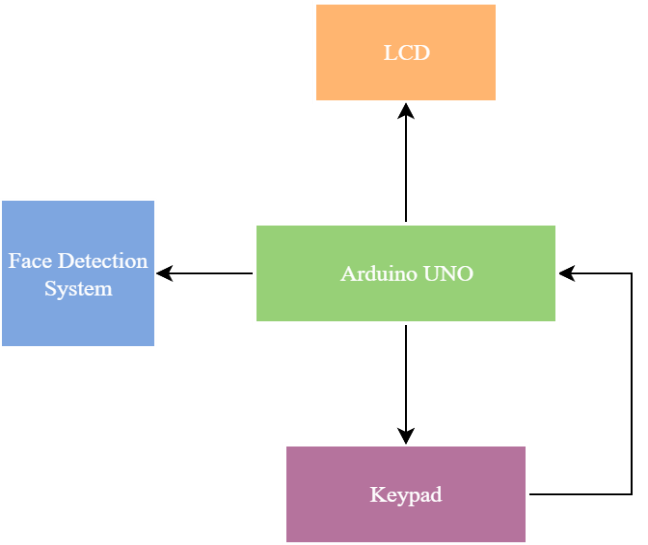
During the voting phase, the system captures a real-time image of the voter and compares it with the stored dataset using the Local Binary Pattern Histogram (LBPH) algorithm—a robust facial recognition technique provided by OpenCV. The algorithm extracts key facial features and performs pattern matching 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:**



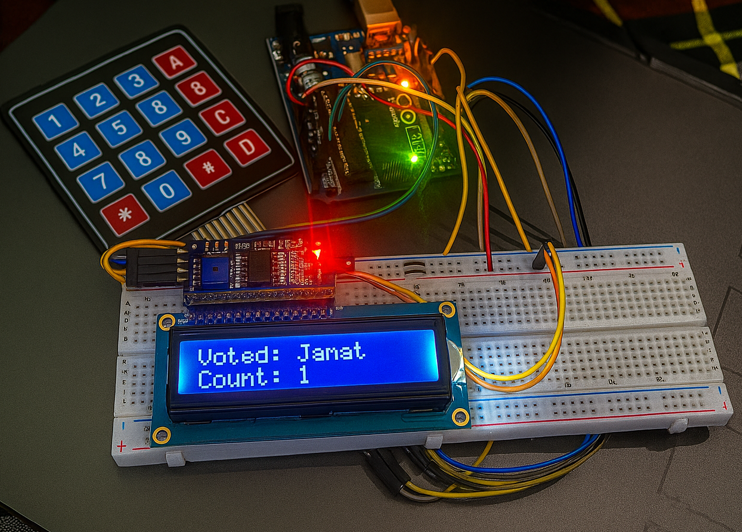
**Fig. 1. Block diagram of the overall system**

The block diagram in figure 1 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.

Table 1 lists the essential components used in building the prototype of the face-based electronic voting system. It includes the Arduino Uno R3 as the main controller, supported by basic electronic components such as LEDs, resistors, transistors, and pushbuttons to simulate voter interaction and feedback. A breadboard and connecting wires were used to assemble the circuit without soldering, ensuring flexibility during prototyping and testing.

**TABLE 1. List of Hardware Component**

|  |  |
| --- | --- |
| **Name of the Components** | **Quantity (pcs)** |
| Arduino Uno R3 | 1 |
| Pushbutton (Set of 6) | 1 |
| NPN Transistor (BJT) | 1 |
| Resistor 220Ω (4pcs) | 1 |
| Wire Connections | 1 |
| LED (4 pcs) | 1 |
| Breadboard | 1 |

****

**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.

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. 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 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. Face recognition accuracy was calculated using the equation

*Accuracy = Correct Recognitions Total Attempts×100\text{Accuracy}*

*= \frac{\text{Correct Recognitions}}{\text{Total Attempts}} \times 100*

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

**TABLE 2. Samples Face Recognition Image**

|  |  |
| --- | --- |
| **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 |  |

1. **Conclusion:**

This research successfully developed a secure, low-cost, and offline-capable face-based electronic voting system designed to address key challenges in the electoral process, particularly in rural or resource-constrained settings. The system meets its core objectives by replacing traditional paper-based methods with a digital solution that enhances transparency, prevents duplicate voting through facial authentication, and operates without reliance on internet connectivity, thereby reducing risks of hacking or data interception. Built using Arduino and OpenCV, the prototype demonstrates practical feasibility for small-scale elections and lays the groundwork for scalable implementations. While limitations such as sensitivity to lighting conditions and camera positioning exist, the study provides a strong foundation for future enhancements, including the integration of more advanced recognition algorithms and additional security layers. Overall, this work contributes to the development of reliable, tamper-resistant electronic voting systems that support more trustworthy and inclusive democratic processes.

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.

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.

Disclaimer

Authors hereby declare that NO generative AI technologies such as Large Language Models (ChatGPT, COPILOT, etc.) and text-to-image generators have been used during the writing or editing of this manuscript.

1. **References:**

[1] K. De Leeuw and K. Bergstra, *The History of Information Security: A Comprehensive Handbook*. Elsevier, 2007. [Online]. Available: <https://books.google.com.bd/books?hl=en&lr=&id=nUJdAAAAQBAJ&oi=fnd&pg=PT7>

[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, p. 012013, 2018. [Online]. Available: <https://iopscience.iop.org/article/10.1088/1742-6596/1007/1/012013/meta>

[3] 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," *Int. J. Sci. Eng. Res.*, vol. 8, no. 3, pp. 98–100, Mar. 2017. [Online]. Available: <https://www.researchgate.net/profile/Khandoker-Tanjim-Ahammad/publication/318661516>

[4] M. K. Kundu, S. Mitra, D. Mazumdar, and S. K. Pal, *Perception and Machine Intelligence: First Indo-Japan Conference, PerMIn 2012*. Springer, 2012. [Online]. Available: <https://books.google.com.bd/books?hl=en&lr=&id=YpJlxfK3rhUC&oi=fnd&pg=PR4>

[5] R. P. Rajashekar and A. H. Soni, "Smart Voting System Using Facial Recognition," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2021, pp. 1–5, doi:10.1109/ICCCI50826.2021.9402541.

[6] J. Wang, L. Chan, and D. Wang, *Neural Information Processing: 13th International Conference, ICONIP 2006, Hong Kong, China*. Springer, 2012. [Online]. Available: <https://link.springer.com/book/10.1007/11893295>

[7] N. Gowda, J. H. Tharun, A. B. N., D. Lamani, and P. J. Patil, "Smart voting system using Face Recognition," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 7, no. 6, 2020.

[8] H. Wechsler, *Reliable Face Recognition Methods: System Design, Implementation, and Evaluation*. Springer, 2009, pp. 11–12. [Online]. Available: <https://link.springer.com/chapter/10.1007/978-0-387-38464-1_7>

[9] M. D. Ashiquzzaman, S. S. Alam, A. Shufian, P. P. Sheikh, and A. H. Siddiqui, "Novel illumination-invariant face recognition approach via reflectance-luminance and local matching model with weighted voting system," J. Eng. Res. Rep., vol. 26, no. 6, pp. 124–138, 2024.

[10] J. Gouda, J. Mani, K. Shashank, and K. Mishra, "Fraud vote detection using facial recognition," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 13, no. 4, pp. 1–6, 2024. [Online]. Available: <https://ijarcce.com/wp-content/uploads/2024/05/IJARCCE.2024.134214.pdf>

[11] GeeksforGeeks, "OpenCV Python Program: Face Detection," [Online]. Available: <https://www.geeksforgeeks.org/python/opencv-python-program-face-detection/>

[12] PythonGeeks, "Face Detection and Recognition using OpenCV," [Online]. Available: <https://pythongeeks.org/face-detection-and-recognition-using-opencv/>

[13] Papers with Code, "Face Recognition Task," [Online]. Available: <https://paperswithcode.com/task/face-recognition>

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

[15] K. Gates, *The Biometric Future: Facial Recognition Technology and the Culture of Surveillance*. NYU Press, 2011, pp. 48–49. [Online]. Available: <https://www.degruyterbrill.com/document/doi/10.18574/nyu/9780814732090.001.0001/html>

[16] 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.

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

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

[19] S. Z. Li and A. K. Jain, *Handbook of Face Recognition*. Springer, 2005, pp. 14–15. [Online]. Available: <https://link.springer.com/book/10.1007/978-0-85729-932-1>

[20] M. Rana, "Review on various face recognition databases," *I-manager's J. Pattern Recogn.*, vol. 9, no. 2, pp. 19189, 2022. [Online]. Available: <https://www.proquest.com/openview/efe2ba56205f09cab01082d820813c68>

[21] R. Rizki and D. Ashiquzzaman, "An application of Viola-Jones method for face recognition for absence process efficiency," *J. Phys. Conf. Ser.*, vol. 1007, p. 012013, 2018.

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

[23] S. K. Chen and Y. H. Chang, *Proc. Int. Conf. on Artificial Intelligence and Software Engineering (AISE 2014)*. DEStech Publications, 2014. [Online]. Available: <https://books.google.com.bd/books?hl=en&lr=&id=gOMNAwAAQBAJ&oi=fnd&pg=PA1>

[24] 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," *Int. J. Sci. Eng. Res.*, vol. 8, no. 3, pp. 98–100, Mar. 2017.

[25] A. M. Amirgalieva, "Development of a modified Viola-Jones algorithm for face recognition," *Fizika-matematika seriâsy*, Almaty Universitetì, 2023. [Online]. Available: <https://bulletin-phmath.kaznpu.kz/index.php/ped/article/view/400>