Facelog: Login System with User Authentication Toolkit Utilizing Convolutional Neural Network Algorithm

Jenefer P. Bermusa., Reagan B. Ricafort

AMA University, Philippines

DOI: https://doi.org/10.51244/IJRSI.2026.13010074

Received: 08 January 2026; Accepted: 13 January 2026; Published: 31 January 2026

ABSTRACT

The study introduces FaceLog, a two-layer authentication framework developed to add digital security by incorporating biometric authentication and multi-factor authentication (MFA). The first security layer utilizes a Convolutional Neural Network (CNN)–based facial recognition model with liveness detection to verify user authenticity in real time. Using the Eye Aspect Ratio (EAR) method, the system detects natural eye blinks to distinguish live users from spoofing attempts involving static or digital images. Once facial verification is successful, the system proceeds to second layer of protection, either a One-Time Password (OTP) or a Time-Based One-Time Password (TOTP) for identity confirmation. This structure ensures that even if one authentication factor is compromised, unauthorized access remains effectively prevented. Evaluation results demonstrate high accuracy, precision, recall, and F1-score, supported by excellent ratings in functionality, usability, and compatibility based the criterion of ISO/IEC 25010 software quality model. The findings affirm that combining biometric authentication with multi-factor verification provides a robust, efficient, and user-centered approach to secure modern login systems, addressing the growing challenges of cybersecurity in digital platforms.

Keywords: Facial Recognition, Convolutional Neural Network (CNN), Liveness Detection, Multi-Factor Authentication (MFA), ISO/IEC 25010.

INTRODUCTION

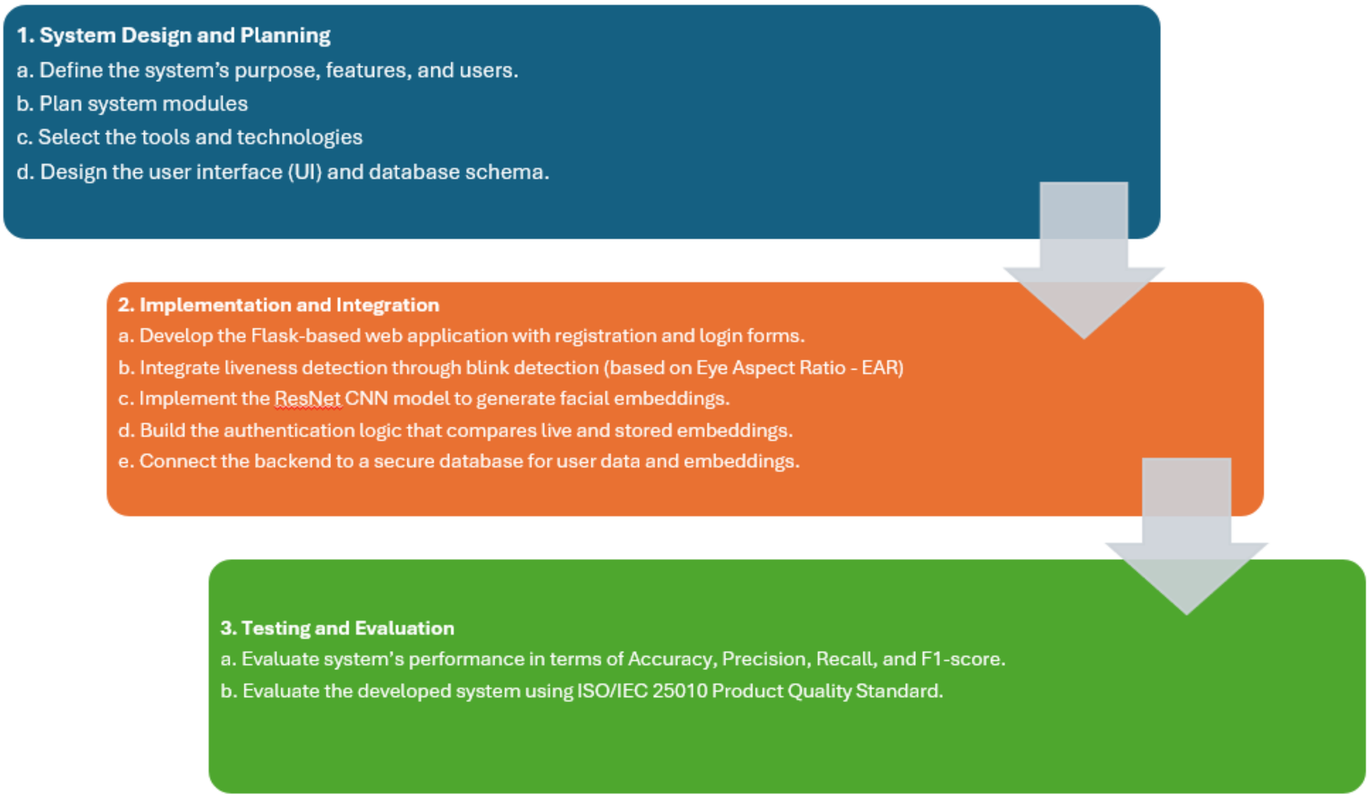
Securing online accounts is a critical concern in today's digital environment (Soriano et.al., 2025)[1], where many services and transactions are now performed through the internet (Albshaier et.al.,2024)[2]; (Khando, et.al., 2022); (Patra et.al., 2022) [3][4]. As cyberthreats continue to evolve (Salim et.al., 2024)[5], traditional login methods such as usernames and passwords have become more vulnerable(Varshney et.al., 2025)[6] to attacks like phishing (Sturman et.al.,2025)[7], brute force attempts (Ha et.al., 2023)[8], and social engineering (Holthouse, et. al.,2025); (Chakraborty, et.al., 2024); (Mahdad, et.al., 2024)[9][10][11]. To address these risks, new technologies are being developed to strengthen authentication processes and improve protection of user credentials (Ismail et.al., 2024); (Khan et.al,2024);(Maraveas et.al.,2024)[12][13][14].

One of the most promising innovations in this field is the use of deep learning-based facial recognition (Amirgaliyev, et.al, 2025); (Gill, et.al, 2024); (Dhakal, et.al,2024)[15][16][17]. This technique allows a system to verify a user's identity by analyzing unique facial features through trained neural network models(Nosrati, et.al, 2024); (Malempati, M., 2024)[18][19]. However, facial recognition alone is not enough, as it can still be exploited using photos of the user (Shukla, et.al, 2025); (Erdogmus, N., & Marcel, S. 2014); (George, A., et., al.,2019)[20][21][22]. For this reason, researchers have also introduced real-time liveness detection(Rehman, Y., et. a., 2019); (Yu, C., et., al., 2019)[23][24], which can identify whether a face is from a live person rather than from a static image. In addition, multi-factor security mechanisms enhance this process by requiring more than one step of verification, making unauthorized access more difficult(Al-Mutairi, A., et., al.,2024)[25].

The performance and trustworthiness of facial recognition systems are often measured using two critical metrics False Acceptance Rate (FAR) and False Rejection Rate (FRR) (Liu, Qin, Wu, & Liang, 2022)[26]. The FAR represents the rate of not permitted users that the system incorrectly accepts, while the FRR indicates the rate of authentic users who are incorrectly rejected. These measures provide a quantitative basis for evaluating the trade-off between security and usability in biometric authentication(Reichinger, D., et.al., 2021);(Yang, W., et. al., 2019); (Iskandar, A., et al. 2024); (Yang, W., et al., 2021)[27][28][29][30[[31]. A lower FAR signifies stronger resistance to spoofing or impersonation, whereas a moderate FRR ensures that genuine users can still access their accounts with minimal inconvenience. In biometric authentication, system performance is primarily evaluated through the False Acceptance Rate (FAR) and False Rejection Rate (FRR). Industry frameworks, particularly the FIDO Biometric Certification Requirements, specify that consumer-grade authenticators should operate at FAR levels around 0.01% (equivalent to 10^{-4} or 1 in 10,000) while maintaining FRR values below 7% for BioLevel 1/1+ and below 5% for BioLevel 2/2+. (FIDO Alliance, 2023)[32]. These thresholds are consistent with recent recommendations from the National Institute of Standards and Technology (NIST), which emphasize adopting stricter biometric error limits to ensure more robust and reliable authentication systems (NIST, 2022)[33].

Despite these advances, most existing systems apply these technologies separately rather than as a unified solution. This research addresses that gap by combining deep learning–based facial recognition, liveness detection, and multi-factor authentication into a single integrated framework. In addition, the system's False Acceptance Rate (FAR) and False Rejection Rate (FRR)(Abdelfatah, R. I. 2024)[34] were examined to ensure an optimal balance between security and usability, where FAR measures how often unauthorized users are mistakenly granted access and FRR indicates how often legitimate users are incorrectly denied(Shinde, S. R.,et., al.,(2025); (Prasad, P., et., al.,(2023); (Khairnar, S.,et. al., 2025)[35][36][37]. To achieve this, the researcher focused on the following objectives: To develop a face authentication system (1); To integrate liveness detection into the authentication process (2); To integrate multi-factor authentication (MFA) option such One-Time Password (OTP) or Time-Based OTP (TOTP)(3);To evaluate the system in terms of F1-score, recall, and precision (4); and To assess the overall system quality using the ISO/IEC 25010 software product quality standard (5). This study aimed to contribute to the development of more secure, reliable, and user-friendly authentication systems by combining deep learning-based facial recognition, real-time liveness detection, and multi-factor security mechanisms into a unified login framework.

METHODOLOGY



1. System Design and Planning
a. Define the system's purpose, features, and users.
b. Plan system modules
c. Select the tools and technologies
d. Design the user interface (UI) and database schema.

2. Implementation and Integration
a. Develop the Flask-based web application with registration and login forms.
b. Integrate liveness detection through blink detection (based on Eye Aspect Ratio - EAR)
c. Implement the ResNet CNN model to generate facial embeddings.
d. Build the authentication logic that compares live and stored embeddings.
e. Connect the backend to a secure database for user data and embeddings.

3. Testing and Evaluation
a. Evaluate system's performance in terms of Accuracy, Precision, Recall, and F1-score.
b. Evaluate the developed system using ISO/IEC 25010 Product Quality Standard.

Fig. 1. Conceptual Design

Figure 1 shows the conceptual design of FaceLog: Login User Authentication Utilizing a Convolutional Neural Network (CNN). The framework composed of three major phases: System Design and Planning, Implementation and Integration, and Testing and Evaluation (Hossain, M., 2023)[38]. The first phase defines the system's objectives, features, and target users while identifying key modules. It also includes the identification of suitable development tools as well as the design of the user interface (UI) and database schema to ensure both efficiency and usability. The second phase, Implementation and Integration, covers the actual system development, including the creation of the Flask-based web application, integration of liveness detection using the Eye Aspect Ratio (EAR)(Mahmood, et. al, 2025)[39], and implementation of the ResNet CNN model for facial feature extraction and verification (Sheng, et. al, 2024)[40]. This phase also includes the development of authentication logic for comparing live and stored embeddings and connecting the backend to a secure database for encrypted data management. The final phase, Testing and Evaluation, validates the system's performance through unit, integration, and system testing. Its effectiveness is measured in terms of Accuracy, Precision, Recall, and F1-score, and evaluated under the ISO/IEC 25010 Product Quality Standard to confirm that the developed system meets global software quality and cybersecurity requirements.

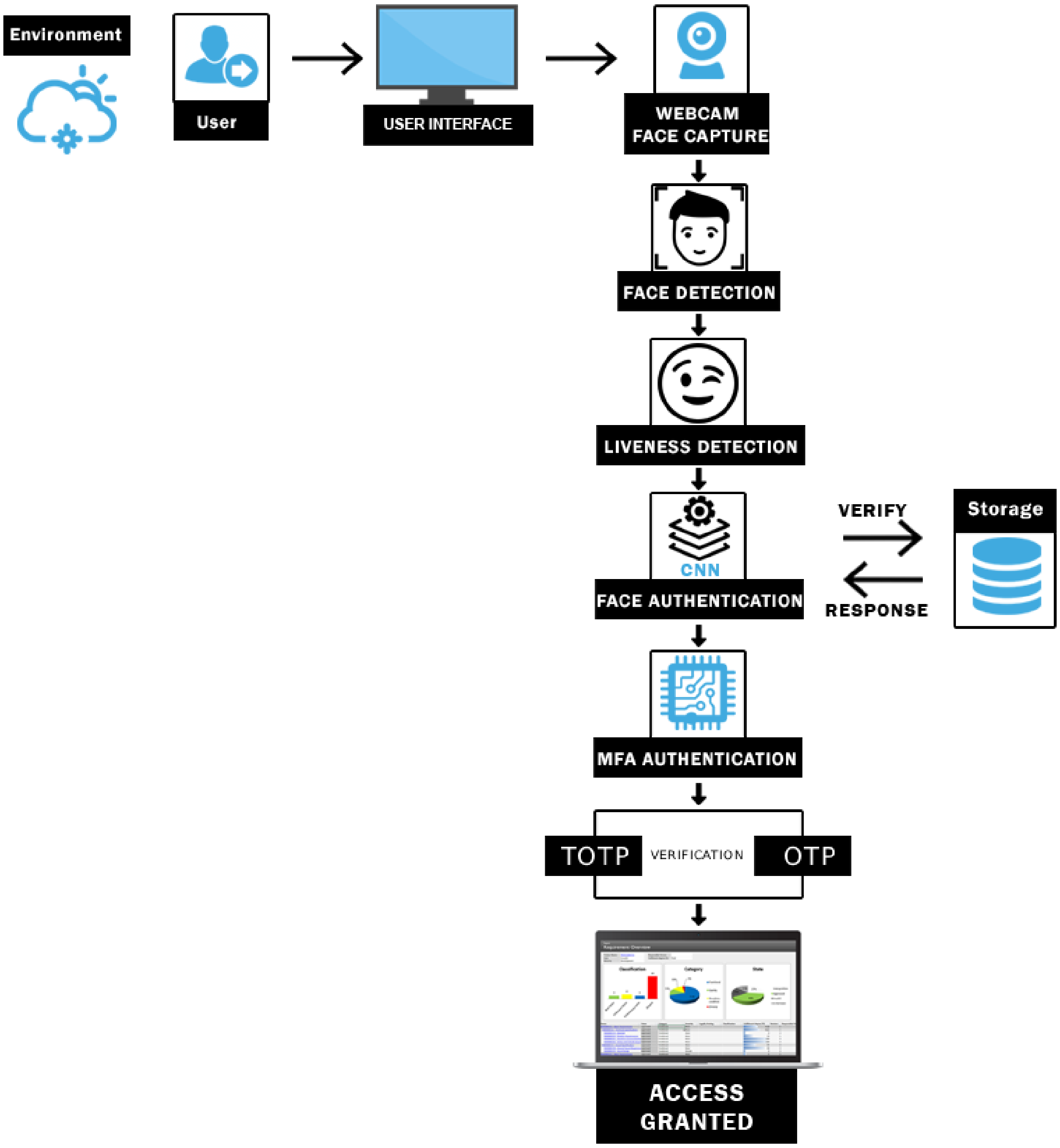


Fig. 2. System Architecture Overview

The diagram shows that the user sends interact with the user interface. The system captures face input via the web camera. The system detects human face and verifies liveness. The captured human face extracted in the CNN layer. In this layer, deep learning models was utilized. The face alignment layer, this layer Normalize the detected faces so they are in a consistent orientation and scale, improving the accuracy of subsequent steps. It utilizes techniques that identifies key points (eyes, nose, mouth). The feature extraction layer, convert the aligned face into a numerical representation (feature vector called embedding) that captures essential facial characteristics. It applies deep learning models (CNN) to extract rich feature representations. The extracted features matched to the database of known faces to identify or verify the individual. If the scanned face matches on the stored data on the database, the system allows sends One-Time Password via email or prompt the user to enter TOTP using Google authenticator. Once OTP or TOTP is verified the user have gained access to the web application system.

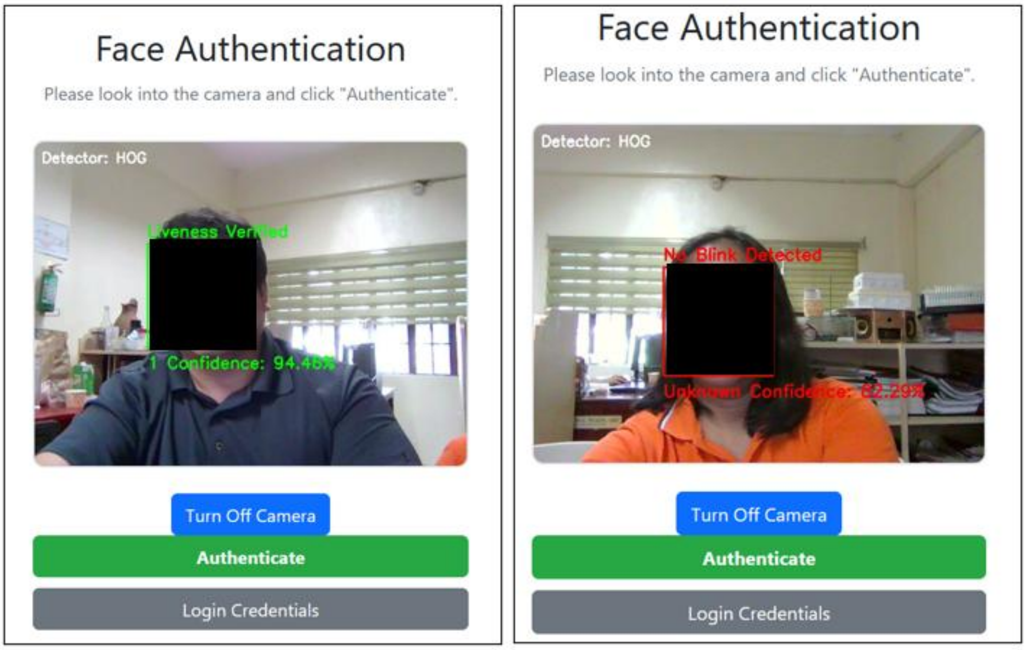
Table 1. Summary of Testing Dataset and Experimental Conditions

Category	Description
Number of Participants	System Users = 52, System Administrators = 18
Total Authentication Attempts	100 face-based authentication attempts
Authentication Mode	Face recognition with liveness detection and MFA
Liveness Mechanism	Blink-based eye aspect ratio (EAR) verification
Camera Type	Standard webcam
Camera Distance	Approximately 40–60 cm from the subject
Lighting Conditions	Controlled indoor lighting, no extreme shadows or backlighting
Face Orientation	Near-frontal pose with minor natural head movement
Occlusion	No masks, sunglasses, or facial obstructions
Environment Type	Controlled indoor environment
Confidence Threshold	90% similarity threshold

Table 1 summarizes the testing dataset and experimental conditions used to evaluate the proposed FaceLog system. The evaluation involved 70 participants, including 52 system users and 18 system administrators, and consisted of 100 face-based authentication attempts. Facial recognition was combined with liveness detection and multi-factor authentication (MFA), with liveness verified using a blink-based eye aspect ratio (EAR) technique. All authentication attempts were performed using a standard webcam positioned approximately 40–60 cm from the subject under controlled indoor lighting conditions, with participants maintaining a near-frontal facial orientation and minimal natural head movement. The experiments were conducted in a controlled indoor environment to establish baseline system performance; therefore, the reported results should be interpreted within the context of these controlled experimental conditions.

RESULTS

A. Design and develop a face authentication that utilizes a Convolutional Neural Network (CNN) algorithm.


Face Authentication

Please look into the camera and click "Authenticate".

Face Authentication

Please look into the camera and click "Authenticate".

Figure 3. Developed face authentication

The figure above illustrates the developed face authentication system, which enables users to verify their identity through a webcam before accessing their accounts. The system first detects whether the input is a human face, then performs liveness verification, and finally measures the confidence level of the detected face, which is set at 90%. This prototype demonstrates the practical application of facial recognition technology in platforms that require enhanced login security.

The authentication module was implemented in Python using core libraries for image processing, facial feature extraction, and identity validation. Below is the library used in the development of this technology.

Table 2: Library used in the development of the face authentication

Library / Module	**Description**
cv2 *(OpenCV)*	Captures images from the camera and performs basic image preprocessing (resizing, color conversion).
dlib	Detects facial landmarks (eyes, nose, mouth) and supports CNN-based face detection.
face_recognition	Generates 128-D facial embeddings and performs face comparison and identification.
numpy	Handles numerical operations on face embeddings (averaging, distance calculation).
scipy.spatial.distance	Computes Euclidean distance between embeddings to validate identity.
faiss	Speeds up face embedding searches and similarity matching for large datasets.
pyotp	Validates Time-based One-Time Passwords (TOTP) for multi-factor authentication.
flask_bcrypt	Hashes and verifies passwords securely alongside face-based validation.

qrcode	Generates QR codes for linking the system to Google Authenticator (TOTP setup).
mysql	Stores and manages user data, facial embeddings, authentication logs, and performance logs.

Pseudo Code of the Developed System

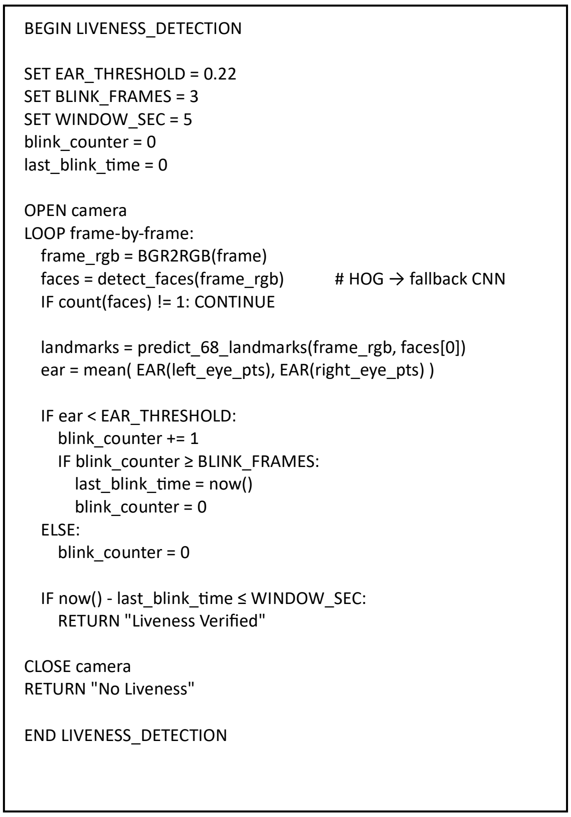
```
BEGIN USER_REGISTRATION
   # --- Libraries used (where applied) ---
   # flask: routing, session, render/redirect, flash
   # mysql.connector: duplicate checks, INSERT user
   # cv2 (OpenCV): camera capture, BGR→RGB conversion, drawing (optional)
   # face_recognition: face_locations (HOG/CNN), face_encodings (128-D)
   # numpy: mean of embeddings, L2 normalization
   # pyotp: TOTP secret + verification
   # qrcode, base64, io: provisioning QR generation + in-page display
   # flask_bcrypt: password hashing
   # json: store 128-D encoding as JSON string

   ROUTE /register [GET, POST]

   # --- Access control ---
   IF NOT (session.admin_id EXISTS AND session.admin_2fa_verified == TRUE) THEN
      flash("Unauthorized access"); REDIRECT to admin_login_face
   ENDIF
     IF request.method == POST THEN

      step ← form.step


      IF step == "register" THEN

         # 1) Read and validate fields

         firstname, lastname, username, password, phone, email, role ← read form fields

         IF any field missing OR role ∉ {"admin","user"} THEN

            flash("Invalid input"); REDIRECT /register

         ENDIF

         # 2) Uniqueness checks (DB)

         IF EXISTS(system_users.username==username OR system_users.email==email) OR

           EXISTS(admin_users.username==username OR admin_users.email==email) THEN

            flash("Username or email already exists"); REDIRECT /register

         ENDIF

# 3) Capture face and compute 128-D template

         video ← cv2.VideoCapture(0)

         frames_collected ← []

         embeddings ← []


         REPEAT total_images times (>=1; e.g., 1–5):

            success, frame_bgr ← video.read()

            IF NOT success: CONTINUE

            frame_rgb ← cv2.cvtColor(frame_bgr, BGR2RGB)

              # HOG first (fast), fallback to CNN (robust)

            boxes ← face_recognition.face_locations(frame_rgb, model="hog")

            IF boxes EMPTY: boxes ← face_recognition.face_locations(frame_rgb, model="cnn")

            IF boxes EMPTY: CONTINUE

            encs ← face_recognition.face_encodings(frame_rgb, boxes)

            IF encs NOT EMPTY: embeddings.APPEND(encs[0])
        video.release()

IF embeddings EMPTY THEN
            flash("Failed to capture face"); REDIRECT /register
         ENDIF

         avg ← numpy.mean(embeddings, axis=0).astype(float32)
         norm ← L2_NORM(avg) + 1e-8
         template_128d ← (avg / norm)   # L2-normalized 128-D vector
         face_encoding_json ← json.dumps(template_128d.tolist())
        # 4) Hash password (bcrypt)
         hashed_password ← bcrypt.hash(password)
         # 5) Create TOTP (server-side) + QR provisioning
         totp_secret ← pyotp.random_base32()
         otp_uri ← TOTP(totp_secret).provisioning_uri(name=email, issuer="FaceAuthApp")
         qr_png ← qrcode.make(otp_uri)
         qr_data_b64 ← base64.encode( png_bytes(qr_png) )
         # 6) Render TOTP verify step (keep secrets server-side)
         RENDER register.html with step="verify_totp" and show qr_data_b64
         STORE pending fields (firstname, lastname, username, hashed_password,
                    phone, email, role, face_encoding_json, totp_secret)
         IN SERVER SESSION (not in client form)
      ELSE IF step == "verify_totp" THEN
         # 7) Verify TOTP
         code ← form.otp
         totp_secret ← session.totp_secret
         IF TOTP(totp_secret).verify(code, valid_window=1) == FALSE THEN
            flash("Invalid Google Authenticator code")
            RENDER verify_totp page again
         ELSE
            # 8) Commit to DB (admin_users or system_users)
            table ← ("admin_users" IF role=="admin" ELSE "system_users")
            INSERT INTO table (firstname, lastname, username, password, phone_number,
                    email, face_encoding, role, status, totp_secret, twofa_enabled)
                  VALUES (firstname, lastname, username, hashed_password, phone,
                      email, face_encoding_json, role, 'active', totp_secret, TRUE)
            db.commit()
            # 9) Optionally cache in-memory (known_faces[username] = template_128d)
            #    (Ensure it’s the normalized vector)
            flash("Registration successful")
            RENDER register.html with step="success"
         ENDIF
      ENDIF
   ELSE
      # GET: show empty registration form
      RENDER register.html with step="register"
   ENDIF
```

The pseudo code presented the algorithmic workflow of the developed face-recognition module integrated into the FaceLog authentication. It outlines the sequential processes involved in capturing, analyzing, and validating a user's facial data during login. Initially, the system activates the webcam and identifies the presence of a human face inside the frame using a convolutional neural network–based detector. The pseudo code also integrates liveness verification through eye-blink detection before proceeding to facial matching, thereby enhancing resistance against spoofing attacks. Once a face is identified, the algorithm extracts key facial landmarks and computes numerical embeddings that represent the unique features of the individual. These embeddings are then matched with the saved reference vectors in the database to measure similarity based on Euclidean distance. A confidence threshold of 90 percent is applied to determine a valid match, ensuring accurate identification and minimizing false acceptances. Presenting this pseudo code provides a clear procedural understanding of how the system operationalizes the face-recognition algorithm, linking theoretical design with its practical software implementation. The structured logic demonstrated in the pseudo code confirms the efficiency, security, and transparency of the authentication process developed in Python.

B. Integrate liveness detection into the authentication process

Pseudo Code

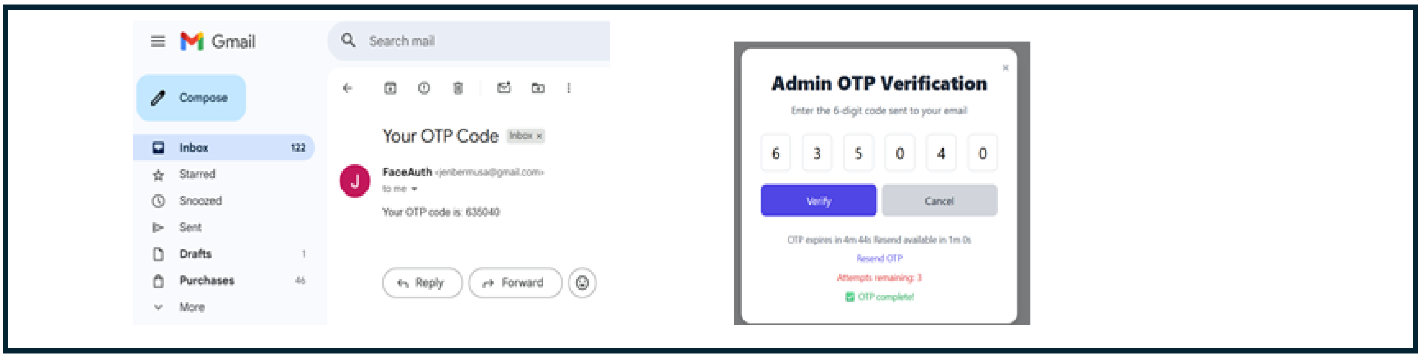
The developed face authentication system includes a liveness detection feature to ensure that only real human users can access the system. The process begins with face detection using two models for higher accuracy. When the captured image is clear and stable, the system applies the Histogram of Oriented Gradients (HOG) model because it is fast and efficient. If the image is blurred or affected by lighting, the system automatically switches to a Convolutional Neural Network (CNN) model, which performs better in complex visual conditions. To confirm that the detected face belongs to a live person, the system uses the Eye Aspect Ratio (EAR) algorithm. This algorithm detects natural eye blinking, which cannot be replicated by static photos or recorded videos. The EAR is calculated from points around the user's eyes, and a drop in the ratio indicates a blink. When a valid blink occurs within the set time window, the system marks the user as "live" and proceeds with facial recognition. If no blink is detected, the process ends and access is denied to prevent spoofing. The pseudocode shows how the liveness detection process is integrated into the face authentication workflow of the developed system.



```
BEGIN LIVENESS_DETECTION

SET EAR_THRESHOLD = 0.22
SET BLINK_FRAMES = 3
SET WINDOW_SEC = 5
blink_counter = 0
last_blink_time = 0

OPEN camera
LOOP frame-by-frame:
   frame_rgb = BGR2RGB(frame)
   faces = detect_faces(frame_rgb)          # HOG → fallback CNN
   IF count(faces) != 1: CONTINUE

   landmarks = predict_68_landmarks(frame_rgb, faces[0])
   ear = mean( EAR(left_eye_pts), EAR(right_eye_pts) )

   IF ear < EAR_THRESHOLD:
      blink_counter += 1
      IF blink_counter ≥ BLINK_FRAMES:
         last_blink_time = now()
         blink_counter = 0
   ELSE:
      blink_counter = 0

   IF now() - last_blink_time ≤ WINDOW_SEC:
      RETURN "Liveness Verified"

CLOSE camera
RETURN "No Liveness"

END LIVENESS_DETECTION
```

C. To integrate Multi-factor-authentication OTP and TOTP

After a user passes the face-liveness check, the system proceeds to multi-factor authentication. The user can choose either a one-time password sent to email or a time-based one-time password (TOTP). Examples of the OTP and TOTP flows are shown below.



Your OTP Code

FaceAuth <jenbermusa@gmail.com>
to me

Your OTP code is: 635040

Admin OTP Verification

Enter the 6-digit code sent to your email

6 3 5 0 4 0

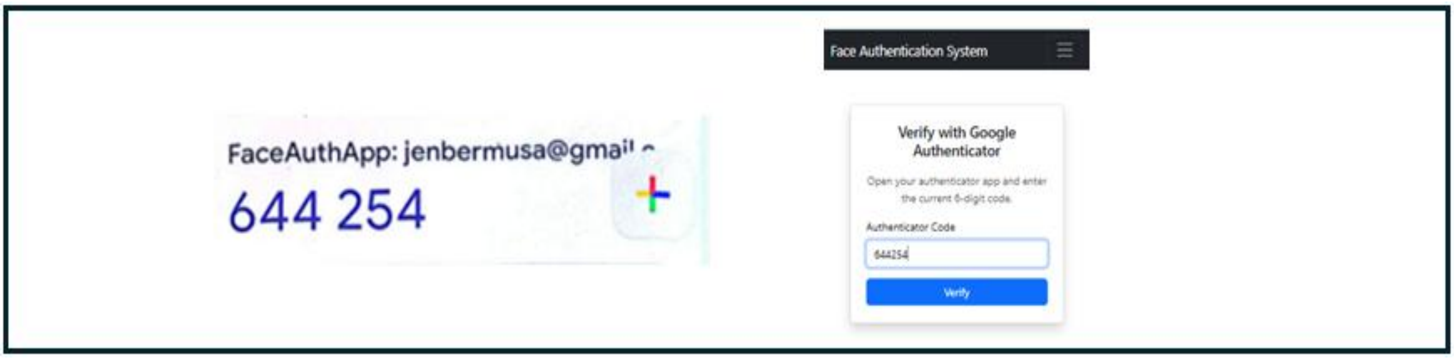
OTP expires in 4m 44s Resend available in 1m 0s

Resend OTP

Attempts remaining: 3

☑ OTP complete!

Fig. 4. Integration of One-Time-Password



FaceAuthApp: jenbermusa@gmai

644 254

Face Authentication System

Verify with Google Authenticator

Open your authenticator app and enter the current 6-digit code.

Authenticator Code: 644254

Fig. 5. Integration of Time-Based-One-Time-Password

Figure 4 and 5 shows the integration of Multi-Factor Authentication (MFA) into the developed face authentication system was successfully implemented using both One-Time Password and Time-Based One-Time Password mechanisms. After the user's identity is verified through face liveness detection, the system prompts an additional verification step where the user can choose between OTP sent via email or TOTP generated through Google Authenticator. This layered integration ensures that even in cases where facial recognition data are intercepted or spoofed, access is denied without successful secondary factor authentication.

During testing, OTP verification functioned reliably through email delivery, with each code expiring after 300 seconds to prevent reuse. The TOTP method synchronized successfully with authenticator applications, demonstrating secure token generation based on time intervals. Both methods verified user authenticity before granting access, thereby strengthening the security of the overall authentication process.

The result confirms that the integration of multi-factor authentication added an essential security layer beyond biometrics. OTP provided accessibility for general users, while TOTP offered stronger protection through encrypted, time-based keys. This balance supports both usability and advanced security, making the system adaptable for different operational environments.

The integration of OTP and TOTP into the face authentication system demonstrates a practical improvement in web-based login security. It provides a layered verification approach that combines biometric recognition with token-based authentication, minimizing risks of unauthorized access, phishing, or replay attacks.

For practical applications, this approach is suitable for secure platforms such as academic portals, e-banking systems, and institutional logins where high security and user trust are essential. From a research perspective, the result validates that combining deep learning-based facial recognition with multi-factor authentication creates a more resilient and user-centered security model, bridging the gap between convenience and protection.

D. System Evaluation in terms of recall, precision and F1-score

The system's performance was measured using the F1-score, which evaluates both precision and recall determining accuracy in face authentication. The F1-score provides a balanced view of how well the system identifies real users and rejects impostors. The computed value was 1.0, derived from the formula:

Solution 1:

$$\boldsymbol{Precision} = \frac{\mathbf{100}}{\mathbf{100+0}} = \mathbf{1.0}\ \boldsymbol{or}\ \mathbf{100\%}$$

Solution 2:

$$\boldsymbol{Recall} = \frac{\mathbf{100}}{\mathbf{100+0}} = \mathbf{1.0}\ \boldsymbol{or}\ \mathbf{100\%}$$

Solution 3:

$$\boldsymbol{F1 - Score} = \mathbf{2}\ \boldsymbol{x}\ \frac{(\boldsymbol{Precision} \times \boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})} = \mathbf{2}\ \boldsymbol{x}\ \frac{\mathbf{1}\boldsymbol{x}\mathbf{1}}{\mathbf{1+1}} = \mathbf{1.0}\ \boldsymbol{or}\ \mathbf{100\%}$$

This result means the system achieved perfect precision and recall, correctly identifying all genuine users without producing false matches or rejections. The score confirms that the model effectively recognized every real face while preventing unauthorized access.

The high accuracy demonstrates the reliability of the developed system that integrates convolutional neural network-based face recognition with liveness detection and multi-factor authentication. The result indicates that combining these security layers improved both recognition performance and system resilience against spoofing attacks.

Under the controlled indoor testing conditions described in Table 1, the system maintained consistent authentication performance at a 90% decision threshold without producing false acceptances or false rejections.

The reported F1-score of 1.0 and corresponding 100% accuracy were obtained under controlled indoor laboratory conditions designed to establish baseline system performance. Consequently, these results represent best-case performance, and system accuracy may vary in real-world environments due to factors such as lighting variability, pose changes, occlusion, and device differences.

In the FaceLog system, identity verification is performed using a similarity-based decision threshold set at 90%, which defines the minimum confidence level required for a successful facial match. This threshold was selected through empirical observation to achieve a balance between system security and user accessibility during baseline testing. At the 90% confidence threshold, the system recorded no false acceptances and no false rejections under controlled indoor laboratory conditions, yielding an F1-score of 1.0. However, the selection of a decision threshold has a direct impact on the balance between the False Acceptance Rate (FAR) and the False Rejection Rate (FRR). Reducing the threshold increases tolerance to facial variation and may lower FRR, but it also raises FAR and increases the risk of unauthorized access. In contrast, increasing the threshold applies stricter matching criteria, which can reduce FAR while simultaneously increasing FRR and negatively affecting user convenience. Since threshold sensitivity was not evaluated across multiple operating points in this study, the reported results reflect best-case performance under controlled conditions.

E. Assessment of system quality using the ISO/IEC 25010 software product quality standard

Table 3. System evaluation summary result(IT Expert).

Category	Mean Score	Qualitative Rating
A. Functional Suitability	4.71	Very Great Extent
B. Performance Efficiency	4.42	Very Great Extent
C. Compatibility	4.50	Very Great Extent
D. Interaction Capability	4.67	Very Great Extent
E. Reliability	4.59	Very Great Extent
F. Security	4.65	Very Great Extent
G. Maintainability	4.62	Very Great Extent
Overall Mean	**4.59**	**Very Great Extent**

The table shows the overall mean score of 4.59 confirms that the developed system meets international software quality standards to a Very Great Extent. It shows a high level of functional accuracy, operational efficiency, user interaction quality, reliability, security, and maintainability. Users rated the system as robust, efficient, and user-friendly, confirming its readiness for real-world deployment and institutional use. The consistent Very Great Extent ratings across all categories highlight both the technical soundness and practical usability of the system, demonstrating its success in meeting its objectives and exceeding user expectations.

Table 4. System evaluation result(Intended Users)

Category	Mean Score	Qualitative Rating
A. Functional Suitability	4.34	Very Great Extent
B. Performance Efficiency	4.57	Very Great Extent
C. Compatibility	4.50	Very Great Extent
D. Interaction Capability	4.41	Very Great Extent
Overall Mean	4.59	**Very Great Extent**

The table presents the overall system evaluation results based on four major quality characteristics from the ISO/IEC 25010 Product Quality Standard: Functional Suitability, Performance Efficiency, Compatibility, and Interaction Capability. All categories received an Excellent qualitative rating, with mean scores ranging from 4.34 to 4.57 and an overall mean of 4.59, indicating outstanding system performance and quality. Among these, Performance Efficiency obtained the highest mean score (4.57), showing that the system performs operations efficiently with fast response times and optimal resource utilization. Compatibility followed with a mean score of 4.50, confirming that the system integrates smoothly with other applications and platforms without interference. Interaction Capability achieved a mean score of 4.41, demonstrating that the system offers a user-friendly, intuitive, and responsive interface. Meanwhile, Functional Suitability recorded a mean of 4.34, signifying that the system effectively fulfills its intended purpose and meets user requirements. Overall, the results verify that the face authentication system is highly functional, efficient, compatible, and user-centered, meeting user expectations and fully aligning with the standards set by the ISO/IEC 25010 product quality model.

CONCLUSIONS

The study successfully developed a face authentication system that integrates facial recognition, liveness detection, and multi-factor authentication to improve login security. The developed system verifies user identity through a webcam by detecting facial features, confirming liveness through eye blinking, and validating access through additional authentication options such as OTP and TOTP. The combination of deep learning-based face recognition and secondary authentication layers demonstrated the system's effectiveness in warranting secure and reliable access control. The integration of these technologies provides a strong defense against unauthorized access and spoofing attempts while maintaining usability and convenience for legitimate users. The findings confirm that the developed system can serve as a practical security framework for web-based applications requiring both accuracy and resilience. It demonstrates the potential of combining biometric and token-based authentication for safer digital environments.

ACKNOWLEDGMENT

The researcher extends sincere gratitude to all individuals and experts who contributed to the completion of this study. Special appreciation is given to the librarians of Isabela State University for their active participation and support. The researcher also acknowledges the IT specialists of the College of Computing Studies, Information and Communication Technology (CCSICT) for their technical expertise and valuable insights, which significantly improved the system's functionality and design. Profound thanks are likewise extended to Dr. Reagan B. Ricafort of AMA University for his expertise, guidance, and continued support. The contributions of all who generously shared their time, knowledge, and experience are deeply appreciated, as they were essential to the successful completion of this study.

REFERENCES

1. Soriano, C. R., Tintiangko, J., & Panaligan, J. H. (2025). Social infrastructures of mobility: Relational encounters between foreign and local digital nomads in the Philippines. City Culture and Society, 43, 100667. https://doi.org/10.1016/j.ccs.2025.100667
2. Albshaier, L., Almarri, S., & Hafizur Rahman, M. M. (2024). A review of blockchain's role in E-Commerce transactions: Open challenges, and future research directions. Computers, 13(1), 27. https://doi.org/10.3390/computers13010027
3. Khando, K., Islam, M. S., & Gao, S. (2022). The emerging technologies of digital payments and associated challenges: A systematic literature review. Future Internet, 15(1), 21. https://doi.org/10.3390/fi15010021
4. Patra, G. K., Rajaram, S. K., Boddapati, V. N., Kuraku, C., & Gollangi, H. K. (2022). Advancing digital payment systems: Combining AI, big data, and biometric authentication for enhanced security. International Journal of Engineering and Computer Science, 11(08), 10–18535. http://dx.doi.org/10.18535/ijecs/v11i08.4698
5. Salim, M. M., Camacho, D., & Park, J. H. (2024). Digital Twin and federated learning enabled cyberthreat detection system for IoT networks. Future Generation Computer Systems, 161, 701–713. https://doi.org/10.1016/j.future.2024.07.017
6. Varshney, G., Raj, A., Sangwan, D., Abuadbba, S., Mishra, R., & Gao, Y. (2025). A login page transparency and visual similarity-based zero-day phishing defense protocol. Computers & Security, 158, 104598. https://doi.org/10.1016/j.cose.2025.104598
7. Sturman, D., Bell, E. A., Auton, J. C., Breakey, G. R., & Wiggins, M. W. (2024). The roles of phishing knowledge, cue utilization, and decision styles in phishing email detection. Applied Ergonomics, 119, 104309. https://doi.org/10.1016/j.apergo.2024.104309
8. Ha, G., Jia, C., Ge, X., Yuan, J., Chen, H., & Li, M. (2023). Efficient and anonymous password-hardened encryption services. Information Sciences, 653, 119771. https://doi.org/10.1016/j.ins.2023.119771
9. Holthouse, R., Owens, S., & Bhunia, S. (2025). The 23andMe data breach: Analyzing credential stuffing attacks, security vulnerabilities, and mitigation strategies. arXiv preprint arXiv:2502.04303. https://doi.org/10.48550/arXiv.2502.04303
10. Chakraborty, S., Jackson, C., Frazier, M., & Clark, K. (2024, March). A study on password protection and encryption in the era of cyber attacks. In SoutheastCon 2024 (pp. 460–465). IEEE. https://doi.org/10.1109/SoutheastCon52093.2024.10500214
11. Mahdad, A. T., Jubur, M., & Saxena, N. (2024, December). Breaching security keys without root: Fido2 deception attacks via overlays exploiting limited display authenticators. In Proceedings of the 2024 on

ACM SIGSAC Conference on Computer and Communications Security (pp. 1686–1700). ACM. https://doi.org/10.1145/3658644.3690286

12. Ismail, M., Madathil, N. T., Alalawi, M., Alrabaee, S., Bataineh, M. A., Melhem, S., & Mouheb, D. (2024). Cybersecurity activities for education and curriculum design: A survey. Computers in Human Behavior Reports, 16, 100501. https://doi.org/10.1016/j.chbr.2024.100501
13. Khan, K., Khurshid, A., & Cifuentes-Faura, J. (2024). Is artificial intelligence a new battleground for cybersecurity? Internet of Things, 28, 101428. https://doi.org/10.1016/j.iot.2024.101428
14. Maraveas, C., Rajarajan, M., Arvanitis, K. D., & Vatsanidou, A. (2024). Cybersecurity threats and mitigation measures in agriculture 4.0 and 5.0. Smart Agricultural Technology, 9, 100616. https://doi.org/10.1016/j.atech.2024.100616
15. Amirgaliyev, B., Mussabek, M., Rakhimzhanova, T., & Zhumadillayeva, A. (2025). A review of machine learning and deep learning methods for person detection, tracking and identification, and face recognition with applications. Sensors, 25(5), 1410. https://doi.org/10.3390/s25051410
16. Gill, A., Jain, D., Sharma, J., Kumar, A., & Garg, P. (2024, May). Deep learning approach for facial identification for online transactions. In 2024 International Conference on Emerging Innovations and Advanced Computing (INNOCOMP) (pp. 715–722). IEEE. https://doi.org/10.1109/INNOCOMP63224.2024.00123
17. Dhakal, S., Parvez, A., Kathuria, S., & Chanti, Y. (2024, May). Face recognition technology powered by artificial intelligence for enhanced security measures. In 2024 Parul International Conference on Engineering and Technology (PICET) (pp. 1–6). IEEE. https://doi.org/10.1109/PICET60765.2024.10716186
18. Nosrati, L., Bidgoli, A. M., & Javadi, H. H. S. (2024). Identifying people's faces in smart banking systems using artificial neural networks. International Journal of Computational Intelligence Systems, 17(1), 9. https://doi.org/10.1007/s44196-023-00383-7
19. Malempati, M. (2024). Generative AI-driven innovation in digital identity verification: Leveraging neural networks for next-generation financial security. SSRN. https://doi.org/10.2139/ssrn.5204991
20. Shukla, S., Varshney, G., Singh, S., & Goel, S. (2025). A passwordless MFA utilizing biometrics, proximity, and contactless communication. Information Security Journal: A Global Perspective, 1–22. https://doi.org/10.1080/19393555.2025.2536033
21. Erdogmus, N., & Marcel, S. (2014). Spoofing face recognition with 3D masks. IEEE Transactions on Information Forensics and Security, 9(7), 1084–1097. https://doi.org/10.1109/TIFS.2014.2322255
22. George, A., Mostaani, Z., Geissenbuhler, D., Nikisins, O., Anjos, A., & Marcel, S. (2019). Biometric face presentation attack detection with multi-channel convolutional neural network. IEEE Transactions on Information Forensics and Security, 15, 42–55. https://doi.org/10.1109/TIFS.2019.2916652
23. Rehman, Y. A. U., Po, L. M., Liu, M., Zou, Z., Ou, W., & Zhao, Y. (2019). Face liveness detection using convolutional-features fusion of real and deep network generated face images. Journal of Visual Communication and Image Representation, 59, 574–582. https://doi.org/10.1016/j.jvcir.2019.02.014
24. Yu, C., Yao, C., Pei, M., & Jia, Y. (2019). Diffusion-based kernel matrix model for face liveness detection. Image and Vision Computing, 89, 88–94. https://doi.org/10.1016/j.imavis.2019.06.009
25. Al-Mutairi, Abdulrahman, & Al-Sahli, Rayan. (2024). Secure Authentication System based on Multi-Factor Authentication. https://doi.org/10.13140/RG.2.2.24880.74247
26. Liu, J., Qin, H., Wu, Y., & Liang, D. (2022). AnchorFace: Boosting TAR@FAR for practical face recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, 36(2), 1371–1379. https://doi.org/10.1609/aaai.v36i2.20063
27. Reichinger, D., Lechner, U., & Pichler, A. (2021). Continuous mobile user authentication using combined behavioral and physiological biometrics. Applied Sciences, 11(24), 11756. https://doi.org/10.3390/app112411756
28. Yang, W., Wang, S., Hu, J., Zheng, G., & Valli, C. (2019). Security and Accuracy of Fingerprint-Based Biometrics: A Review. Symmetry, 11(2), 141. https://doi.org/10.3390/sym11020141
29. Iskandar, A., et al. (2024). Biometric systems for identification and verification scenarios using threshold-based matching and performance trade-offs. Neural Computing & Applications. https://doi.org/10.1007/s00521-023-09390-3
30. Yang, W., et al. (2021). Biometrics for Internet-of-Things Security: A Review. Sensors (special issue). https://doi.org/10.3390/s21082728

31. FIDO Alliance. (2023). FIDO Biometric Certification Requirements: Performance Criteria for FAR and FRR. FIDO Alliance.https://fidoalliance.org/specs/biometric/requirements/Biometrics-Requirements-v3.0-fd-20230111.html
32. National Institute of Standards and Technology (NIST). (2022). Guidelines for Biometric Performance and Error Rate Management. U.S. Department of Commerce.
33. Abdelfatah, R. I. (2024). Robust biometric identity authentication scheme using quantum voice encryption and quantum secure direct communications for cybersecurity. Journal of King Saud University - Computer and Information Sciences, 36(5), 102062. https://doi.org/10.1016/j.jksuci.2024.102062
34. Shinde, S. R., Bongale, A. M., Dharrao, D., & Thepade, S. D. (2025). An enhanced light weight face liveness detection method using deep convolutional neural network. MethodsX, 14, 103229. https://doi.org/10.1016/j.mex.2025.103229
35. Prasad, P., Lakshmi, A., Kautish, S., Singh, S., Shrivastava, R., Almazyad, A., ... & Ali, M. (2024). Robust Facial Biometric Authentication System Using Pupillary Light Reflex for Liveness Detection of Facial Images. Computer Modeling in Engineering & Sciences, 139(1), 725. https://doi.org/10.32604/cmes.2023.030640
36. Khairnar, S., Gite, S., Pradhan, B., Thepade, S., & Alamri, A. (2025). Optimizing CNN architectures for face liveness detection: performance, efficiency, and generalization across datasets. Computer Modeling in Engineering & Sciences, 143(3), 3677. https://doi.org/10.32604/cmes.2025.058855
37. Hossain, M. I. (2023). Software development life cycle (SDLC) methodologies for information systems project management. International Journal for Multidisciplinary Research, 5(5), 1–36. E-ISSN: 2582-2160
38. Mahmood, M., Tasnim, N., Saimu, T. M., Rahman, M. A., Quadir, H. S., Rashed, M. G., & Das, D. (2025, February). A novel approach to liveness detection: Real-time eye blink, head movement, and lip movement analysis. In 2025 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1–6). IEEE. https://doi.org/10.1109/ECCE64574.2025.11014007
39. Sheng, H., & Lau, M. (2024). Optimising real-time facial expression recognition with ResNet architectures. Journal of Machine Intelligence and Data Science (JMIDS), 5(1), 33–45. https://doi.org/10.11159/jmids.2024.005