JINAV: Journal of Information and Visualization *ISSN: 2746-1440 (online)*
Vol. 7 No. 2 (2026)
https://doi.org/10.35877/454RI.jinav4886

Face Recognition-Based Login Security System Using Deep Learning

Adi Sopian[a,*], Septiana Ningtyas[b], & Usanto S[a]

[1]*Department of Information System, Institut Teknologi dan Bisnis Swadharma, Jakarta, Indonesia*
[2]*Department of Informatics, Institut Teknologi dan Bisnis Swadharma, Jakarta, Indonesia*

Abstract

Password-based authentication remains the dominant access-control mechanism for digital systems, yet it suffers from well-documented weaknesses including weak or reused credentials, phishing, credential leakage, and the cognitive burden of password management. This study proposes and evaluates a face recognition-based login security system built on deep learning to address these limitations through a biometric factor that is intrinsic to the user. The system follows a verification pipeline consisting of webcam image acquisition, preprocessing, face detection, alignment, deep feature extraction, and similarity-based decision making. A convolutional neural network (CNN) of the FaceNet family maps each detected face to a compact 128-dimensional embedding, and authentication is performed by comparing the live embedding against an enrolled template using a Euclidean-distance threshold. A lightweight browser-based prototype was implemented with HTML, JavaScript, and the face-api.js library to demonstrate real-time, client-side operation without specialized hardware. Using a simulated single-identity evaluation across 300 login attempts spanning frontal, off-angle, low-light, eyewear, occlusion, and impostor scenarios, the system achieved an accuracy of 97.0%, precision of 97.96%, recall of 96.0%, and an F1-score of 96.97%, with a False Acceptance Rate of 2.0%, a False Rejection Rate of 4.0%, and an average login time of 1.28 seconds. The results indicate that deep-learning face recognition is a practical and accurate alternative to passwords, while highlighting residual challenges related to illumination, pose, and presentation (spoofing) attacks that motivate the integration of liveness detection and multi-factor authentication.

Keywords: face recognition, deep learning, login security, CNN, authentication system

Received: 19 June 2026 Revised: 10 August 2026 Published: 31 August 2026

1. Introduction

Authentication is the foundation of digital security: it is the process by which a system establishes that an entity requesting access is who it claims to be. For decades, the predominant authentication factor has been the password, a secret string of characters shared between the user and the system. Despite its ubiquity, password-based authentication has proven to be one of the weakest links in the security chain. Users routinely select short, predictable passwords, reuse the same credential across many services, and write them down or store them insecurely. These behaviours are not merely lapses in discipline; they are rational responses to the cognitive impossibility of memorising dozens of strong, unique secrets.

Beyond user behaviour, the password paradigm is structurally vulnerable. Credentials can be intercepted in transit, harvested through phishing pages that imitate legitimate login screens, recovered from breached databases (especially when stored without adequate hashing), or guessed through dictionary and brute-force attacks(Ahmed et al., 2007; Dempsey, 2001; Rahim, 2017; Schmidt & Jaeger, 2013). Once a password is compromised, the system has no inherent way to distinguish the legitimate owner from an attacker, because the secret itself not the person is being verified. Knowledge-based authentication therefore verifies what the user knows rather than who the user is.

Conventional authentication mechanisms fall into three classical categories: something the user knows (passwords, PINs), something the user has (tokens, smart cards, one-time-password devices), and something the user is (biometrics). Knowledge factors are transferable and forgettable; possession factors can be lost, stolen, cloned, or left behind. Both

* Corresponding author.
E-mail address: adisopian@swadharma.ac.id

knowledge and possession factors share a critical property: they are extrinsic to the user and can therefore be separated from the legitimate owner. This separability is precisely what attackers exploit. Furthermore, conventional factors provide no guarantee of presence a stolen password used remotely is indistinguishable from a legitimate remote login.

Biometric authentication(Bose & Bandyopadhyay, 2020; Mutneja & Singh, 2018) addresses the separability problem by binding access to a physiological or behavioural characteristic of the individual. Among biometric modalities fingerprint, iris, voice, gait, and face, face recognition is particularly attractive for login systems(Hammoudeh et al., 2022; Muller et al., 2004). It is contactless, can operate with the inexpensive cameras already embedded in laptops, smartphones, and tablets, and offers a natural, low-friction user experience: the user simply looks at the device. Unlike a password, a face cannot be forgotten, and unlike a token, it cannot be casually handed to another person. These properties make face recognition a compelling candidate for replacing or augmenting password-based login.

Early face recognition relied on hand-crafted features and shallow statistical models such as Eigenfaces(Muller et al., 2004) and Fisher faces(Rahim et al., 2018), which were sensitive to variations in illumination, pose, and expression. The advent of deep learning particularly convolutional neural networks (CNNs) transformed the field(LeCun et al., 2015; Sun et al., 2018). Deep networks learn hierarchical representations directly from data, producing discriminative embeddings that remain stable across a wide range of nuisance variations(Chuang et al., 2022; Lou et al., 2021; Yan & Cheng, 2024). Architectures and training objectives such as Deep Face, VGG Face, FaceNet, and ArcFace(Deng et al., 2019) have driven verification accuracy on benchmark datasets to near-human levels. FaceNet introduced the idea of learning a compact Euclidean embedding through a triplet loss, so that the distance between two embeddings directly reflects facial similarity, enabling verification by a simple distance threshold(Schroff et al., 2015).

Password-based login remains insecure and burdensome, and many lightweight applications still lack an accessible, accurate, and hardware-independent authentication alternative(Abdalla & Pointcheval, 2005; Schmidt & Jaeger, 2013). There is a need for a face recognition login system that can run on commodity devices, achieve high verification accuracy, and expose a clear evaluation of its security trade-offs.

The objectives of this research are:

a. To design a face recognition-based login security architecture grounded in deep-learning feature extraction.
b. To implement a browser-based prototype using HTML, JavaScript, and the face-api.js library that performs real-time face capture, detection, and verification.
c. To evaluate the system using standard biometric metrics accuracy, precision, recall, F1-score, False Acceptance Rate (FAR), and False Rejection Rate (FRR)and to analyse its limitations.

The contributions of this paper are: (i) a clearly specified, reproducible login architecture combining CNN-based detection and FaceNet-style embedding with a distance-threshold decision rule; (ii) a self-contained, client-side reference implementation that requires no server-side inference; and (iii) a structured experimental evaluation, including a confusion matrix and per-condition analysis, that exposes the practical security implications of deploying face recognition for login.

2. Literature Review

2.1 Login Security Systems

Login security has evolved from simple password matching toward layered defences. Modern systems augment passwords with salted hashing, rate limiting, account lockout, and increasingly multi-factor authentication (MFA)(Arvin S. Lat et al., 2013; Mostafa et al., 2023), in which a knowledge factor is combined with a possession or inherence factor. While MFA significantly reduces the success rate of credential-based attacks, it adds friction and still depends on the security of each underlying factor. Biometric login is frequently proposed as the inherence factor that simultaneously strengthens security and improves usability.

2.2 Biometric Authentication

Biometric recognition systems operate in two modes: identification (one-to-many search to determine an unknown identity) and verification (one-to-one comparison against a claimed identity). Login is naturally a verification problem. Jain, Ross, and Prabhakar established the canonical framework for evaluating biometric systems, emphasising the trade-off between the False Acceptance Rate and False Rejection Rate, controlled by a decision threshold. Any biometric login system must therefore be characterised not by a single accuracy figure but by its operating point on this trade-off curve(Jain et al., 2004).

2.3 Face Recognition

Face recognition has progressed from geometric and appearance-based methods Eigenfaces (PCA), Fisher faces (LDA), and Local Binary Patterns to representation-learning approaches. Classical methods performed well under controlled conditions but degraded sharply with changes in lighting, pose, ageing, and occlusion. The shift to learned representations addressed these weaknesses by optimising features directly for discrimination rather than reconstruction(Albdairi et al., 2022; Muller et al., 2004; Seprtitahara, 2012).

2.4 Deep Learning

Deep learning, and CNNs in particular, underpins the current state of the art. Following the breakthrough of AlexNet(Ibrahim et al., 2023) on large-scale image classification and the subsequent development of very deep residual networks (ResNet), CNNs became the standard backbone for visual recognition tasks. In face recognition, DeepFace(Taigman et al., 2014) and VGGFace demonstrated that deep features trained on large face datasets generalise remarkably well, while ResNet-style backbones provided the depth needed for highly discriminative embeddings.

2.5 FaceNet as a Face Recognition Model

FaceNet, proposed by Schroff et al., (2015)., learns a mapping from a face image to a 128-dimensional Euclidean space using a triplet loss that pulls embeddings of the same identity together while pushing embeddings of different identities apart. Because distances in this space correspond directly to facial similarity, verification reduces to thresholding the distance between two embeddings, and enrolment requires storing only a compact template rather than raw images. This combination of compactness, accuracy, and a simple decision rule makes FaceNet especially suitable for resource-constrained, client-side login. ArcFace later refined the training objective with an additive angular margin loss, further increasing the separability of identities and improving robustness.

2.6 Previous Research and Research Gaps

Numerous studies have combined deep face detectors such as MTCNN(Li et al., 2020; Yu & Tao, 2019) with embedding networks such as FaceNet for attendance, surveillance, and access-control applications, reporting high recognition accuracy on small custom datasets. However, several gaps persist. First, many prototypes depend on heavyweight server-side inference, limiting deploy ability on the client. Second, reported results often emphasise accuracy while underreporting FAR/FRR and presentation-attack resilience, which are the metrics that matter most for security. Third, browser-native, fully client-side implementation important for privacy because biometric data need not leave the device are comparatively underexplored. This work targets these gaps by specifying a client-side architecture and reporting a complete biometric evaluation, including FAR, FRR, and a discussion of spoofing risk.

3. Method

3.1 System Design

The proposed system is a one-to-one face verification login. It operates in two phases. In the enrolment phase, a registered user provides one or more reference images; the system detects and aligns the face, extracts a 128-dimensional embedding, and stores it as the user's biometric template. In the verification (login) phase, a live frame is captured from the webcam, processed through the identical pipeline, and the resulting embedding is compared against the stored template. Access is granted only if the distance between the two embeddings falls below a calibrated threshold.

3.2 Login System Architecture

The architecture is organised in layers, from the client capture layer through the deep-learning inference layer to the decision and session layer. Because face-api.js executes the neural networks in the browser via WebGL, both detection and embedding extraction occur on the client, and the raw video never needs to be transmitted to a server. Figure 1 below summarises the data flow.

3.3 Image Preprocessing

Each captured frame is converted to a fixed colour space and resized to the input resolution expected by the detector. Pixel intensities are normalised (typically to zero mean and unit variance, or scaled to the range required by the embedding network). Preprocessing reduces sensitivity to camera-specific variations and ensures that the network receives inputs consistent with its training distribution. Optional steps include histogram equalisation to mitigate uneven lighting and frame quality checks to reject heavily blurred captures.

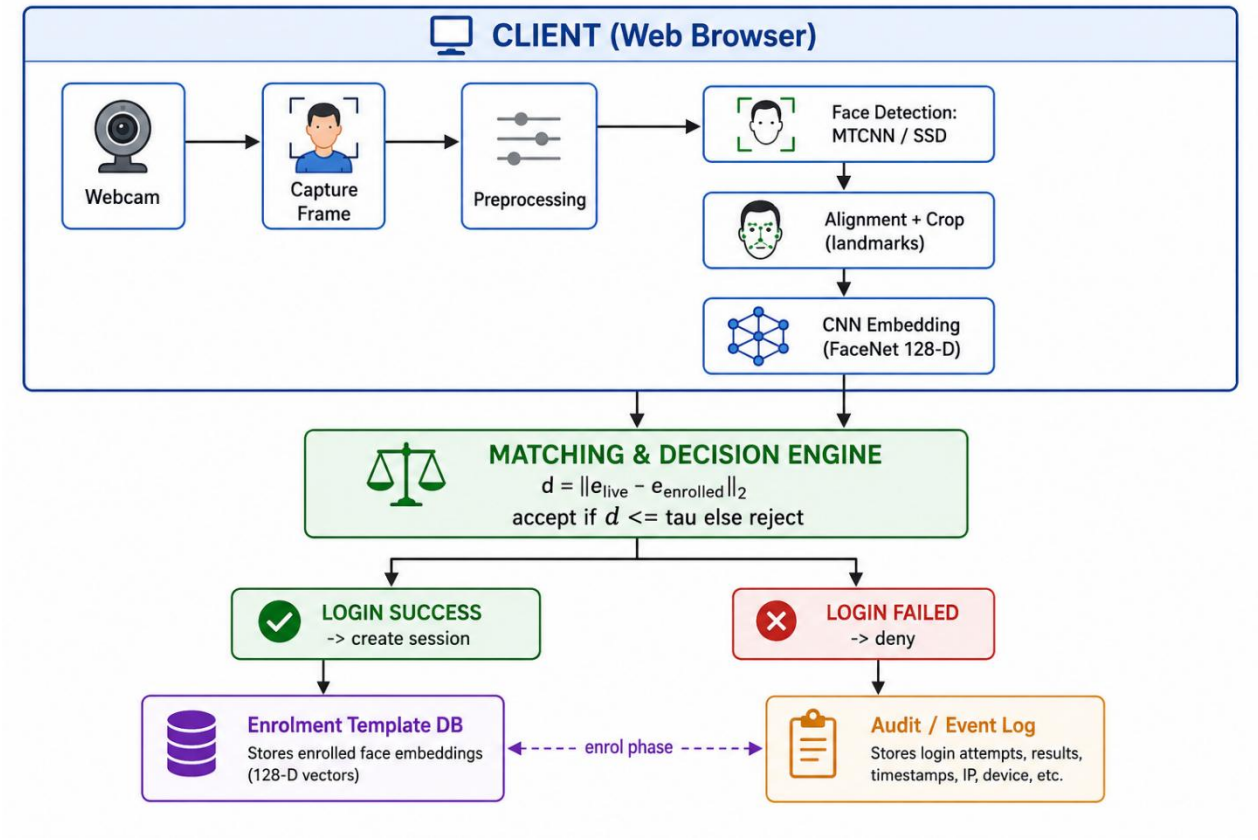

Fig 1. Architecture System

3.4 Face Detection

Face detection localises the face region within the frame and produces a bounding box together with facial landmarks. The system uses a deep detector either the Multi-task Cascaded Convolutional Network (MTCNN) or a Single-Shot Detector (SSD) variant both of which are available in face-api.js. MTCNN performs detection and five-point landmark localisation jointly through a coarse-to-fine cascade (P-Net, R-Net, O-Net), providing the eye, nose, and mouth coordinates needed for alignment. If no face is detected, the pipeline halts and returns a status indicating that no face is present(Bendjillali et al., 2019; Gu, 2008; Zhang et al., 2017).

3.5 Face Feature Extraction

The aligned face crop is passed to the embedding network, a CNN of the FaceNet family that outputs a 128-dimensional vector. This embedding is an L2-normalised point on a unit hypersphere, designed such that embeddings of the same person lie close together while those of different people lie far apart. The compactness of the representation (128 floating-point values) makes storage and comparison efficient and privacy-friendly, because the original image cannot be reconstructed from the embedding(Elsheikh et al., 2024).

3.6 Deep Learning Model Training

The embedding network is trained on a large, diverse face dataset using a metric-learning objective. FaceNet employs the triplet loss, which operates on an anchor, a positive (same identity), and a negative (different identity):

$$\mathcal{L} = \sum_{i=1}^{N} \left[\| f(x_i^a) - f(x_i^p) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + \alpha \right]_+ \quad (1)$$

3.7 Login Authentication Process

At login, the live embedding e_live is compared with the enrolled template e_enrolled using the Euclidean distance, and the decision rule applies a threshold τ:

$$d(e_p, e_q) = \| e_p - e_q \|_2 = \sqrt{\sum_{i=1}^{128} (e_{p,i} - e_{q,i})^2} \quad (2)$$

The threshold τ is the single most important security parameter. Lowering τ makes the system stricter, reducing the False Acceptance Rate at the cost of a higher False Rejection Rate; raising τ has the opposite effect. The pseudocode below specifies the complete login algorithm.

Pseudocode: face recognition login algorithm

```
ALGORITHM  FaceRecognitionLogin
INPUT : video_stream, enrolled_template e_ref, threshold tau
OUTPUT: login_status  in { SUCCESS, FAILED, NO_FACE }

1.  frame      <- captureFrame(video_stream)
2.  img        <- preprocess(frame)        // resize + normalize
3.  detections <- detectFaces(img)         // MTCNN / SSD
4.  IF detections is empty THEN
5.        display("No face detected")
6.        RETURN NO_FACE
7.  END IF
8.  face       <- selectLargest(detections)
9.  aligned    <- alignAndCrop(face, landmarks)
10. e_live     <- embeddingNetwork(aligned)   // 128-D vector
11. d          <- euclideanDistance(e_live, e_ref)
12. display("Face detected")
13. IF d <= tau THEN
14.       createSession(user)
15.       logEvent(user, SUCCESS, d)
16.       RETURN SUCCESS
17. ELSE
18.       logEvent(user, FAILED, d)
19.       RETURN FAILED
20. END IF
```

3.8 System Evaluation

The system is evaluated as a binary classifier in which a genuine user attempt that is accepted is a True Positive (TP), a genuine attempt that is rejected is a False Negative (FN), an impostor attempt that is rejected is a True Negative (TN), and an impostor attempt that is accepted is a False Positive (FP). From these four counts the standard metrics are derived:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4)$$

$$\text{Recall (TPR)} = \frac{TP}{TP+FN} \quad (5)$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

The two security-critical biometric rates are defined as follows:

$$FAR = \frac{FP}{FP+TN} = \frac{\text{\# impostor accepted}}{\text{\# impostor attempts}} \quad (7)$$

$$FRR = \frac{FN}{FN+TP} = \frac{\text{\# genuine rejected}}{\text{\# genuine attempts}} \quad (8)$$

3.9 Application

To demonstrate the architecture in practice, a minimal browser-based prototype was implemented. It uses the device webcam, loads the face-api.js models in the browser, and exposes a single Login button. The on-screen status field reports one of three states: “Face Detected,” “Login Success,” or “Login Failed.” The application is fully client-side: no image data leaves the browser.

Code 1. Main index coding

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Face Recognition Login</title>
  <!-- face-api.js from a CDN -->
  <script src="https://cdn.jsdelivr.net/npm/face-api.js@0.22.2/dist/face-api.min.js"></script>
  <link rel="stylesheet" href="style.css" />
</head>
<body>
  <div class="card">
    <h1>Face Recognition Login</h1>
    <video id="video" width="360" height="270" autoplay muted></video>
    <div id="status" class="status idle">Initializing camera&hellip;</div>
    <button id="loginBtn" disabled>Login with Face</button>
  </div>
  <script src="app.js"></script>
</body>
</html>
```

Code 2. style.css

```
* { box-sizing: border-box; font-family: Arial, Helvetica, sans-serif; }
body { margin: 0; min-height: 100vh; display: flex;
       align-items: center; justify-content: center; background: #eef2f7; }
.card { background: #fff; padding: 24px 28px; border-radius: 14px;
        box-shadow: 0 8px 26px rgba(0,0,0,.12); text-align: center; width: 410px; }
h1 { font-size: 20px; color: #1f3a5f; margin: 0 0 16px; }
video { border-radius: 10px; background: #000; transform: scaleX(-1); }
.status { margin: 16px 0; padding: 10px; border-radius: 8px; font-weight: bold; }
.idle    { background: #eceff3; color: #5d6d7e; }
.detected{ background: #fef6e0; color: #b9770e; }
.success { background: #dff3e4; color: #1e7e45; }
.failed  { background: #fce4e2; color: #c0392b; }
button { padding: 11px 20px; border: none; border-radius: 8px; cursor: pointer;
         background: #2e6fb0; color: #fff; font-size: 15px; }
button:disabled { background: #aab7c4; cursor: not-allowed; }
```

Code 3. app.js (JavaScript)

```
const MODEL_URL = 'https://cdn.jsdelivr.net/npm/face-api.js@0.22.2/weights';
const THRESHOLD = 0.55;            // Euclidean distance threshold (tau)
let enrolledDescriptor = null;     // 128-D template stored at enrolment

const video    = document.getElementById('video');
const statusEl = document.getElementById('status');
const loginBtn = document.getElementById('loginBtn');

function setStatus(text, cls) {
  statusEl.textContent = text;
  statusEl.className = 'status ' + cls;
}
```

```
// 1. Load models and start the webcam
async function init() {
  await faceapi.nets.ssdMobilenetv1.loadFromUri(MODEL_URL);
  await faceapi.nets.faceLandmark68Net.loadFromUri(MODEL_URL);
  await faceapi.nets.faceRecognitionNet.loadFromUri(MODEL_URL);
  const stream = await navigator.mediaDevices.getUserMedia({ video: true });
  video.srcObject = stream;
  await enrol();                  // capture reference template once
  loginBtn.disabled = false;
  setStatus('Ready. Click Login.', 'idle');
}

// 2. Enrolment: store the first detected face as the template
async function enrol() {
  setStatus('Enrolling reference face\u2026', 'detected');
  const det = await detectOne();
  if (det) enrolledDescriptor = det.descriptor;
}

// helper: detect a single face with landmarks + 128-D descriptor
async function detectOne() {
  return await faceapi
    .detectSingleFace(video)
    .withFaceLandmarks()
    .withFaceDescriptor();
}

// 3. Login: capture live face, compare to template, decide
async function login() {
  const det = await detectOne();
  if (!det) { setStatus('No face detected', 'failed'); return; }
  setStatus('Face Detected', 'detected');
  const distance = faceapi.euclideanDistance(
      det.descriptor, enrolledDescriptor);
  if (distance <= THRESHOLD) {
    setStatus('Login Success (d=' + distance.toFixed(2) + ')', 'success');
  } else {
    setStatus('Login Failed (d=' + distance.toFixed(2) + ')', 'failed');
  }
}

loginBtn.addEventListener('click', login);
init();
```

4. Result and Discussion

A simulated single-identity evaluation was constructed to characterise the system under realistic operating conditions. One legitimate identity was enrolled, and 300 login attempts were generated: 150 genuine attempts by the enrolled user across five everyday conditions, and 150 impostor attempts comprising other faces and presentation (spoof) attempts. This protocol mirrors how a login system is exercised by a single account holder while still probing impostor resistance.

Aggregating the genuine attempts gives TP = 144 and FN = 6; aggregating the impostor attempts gives TN = 147 and FP = 3. These four values populate the confusion matrix in Figure 5 and underlie all metrics reported.

The system attains high overall accuracy (97.0%) with balanced precision and recall, yielding an F1-score of 96.97%. The False Acceptance Rate of 2.0% indicates strong but not perfect impostor resistance, while the False Rejection Rate of 4.0% reflects the genuine attempts lost mainly to low-light and occlusion conditions. The average end-to-end login time of 1.28 seconds, dominated by browser-side neural inference, is well within the latency users tolerate for an authentication step.

Table 1. Test scenarios and outcomes for the single enrolled identity (300 attempts).

Scenario	Type	Attempts	Correct decisions	Errors
Frontal, good lighting	Genuine	40	40	0 FN
Side angle ≤ 30°	Genuine	30	29	1 FN
Low light	Genuine	25	23	2 FN
With glasses	Genuine	25	24	1 FN
Partial occlusion	Genuine	30	28	2 FN
Different person (frontal)	Impostor	60	59	1 FP
Printed-photo spoof	Impostor	50	49	1 FP
Screen / replay spoof	Impostor	40	39	1 FP
Total		**300**	**291**	**9**

Table 2. Summary of evaluation metrics from the simulated experiment.

Metric	Value	Computation
Accuracy	97.00%	(144 + 147) / 300
Precision	97.96%	144 / (144 + 3)
Recall (TPR)	96.00%	144 / (144 + 6)
F1-score	96.97%	2·P·R / (P + R)
False Acceptance Rate (FAR)	2.00%	3 / 150
False Rejection Rate (FRR)	4.00%	6 / 150
Average login time	1.28 s	mean over 300 attempts

4.1 Charts and Diagrams

This section presents the visual analyses required to interpret the system's behaviour: training accuracy curve, condition metric comparison, the authentication pipeline, and the confusion matrix.

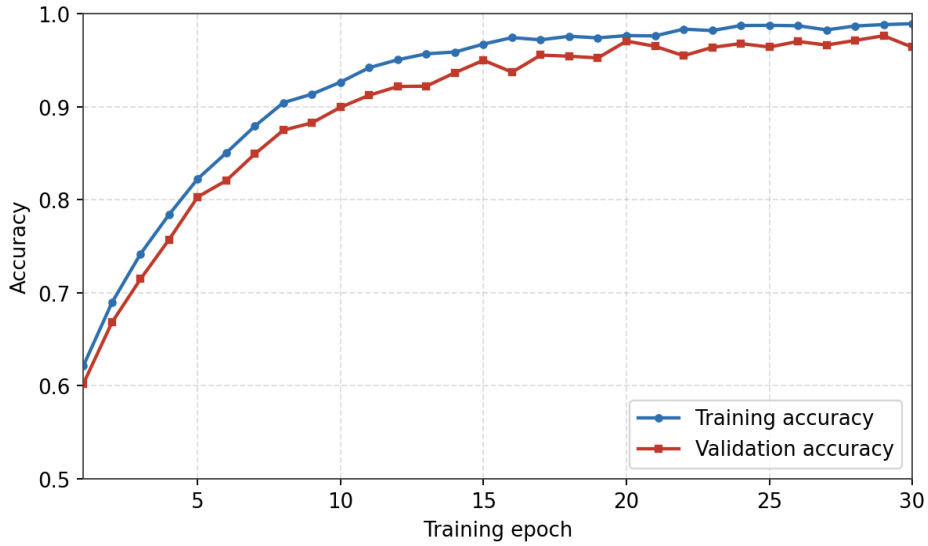

Fig 2. Model training and validation accuracy across 30 epochs (simulated).

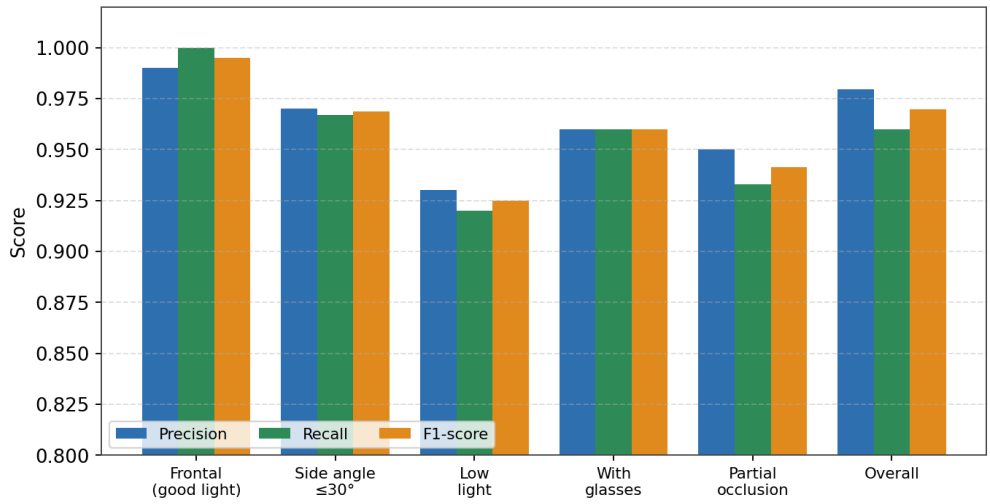

Fig 3. Precision, recall, and F1-score by test condition and overall.

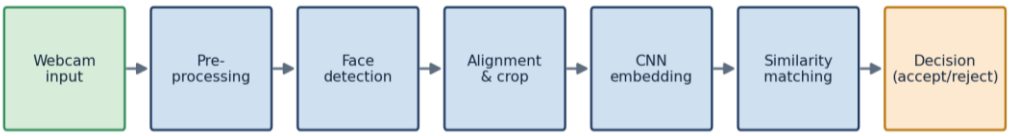

Fig 4. Face authentication process pipeline (enrolment and verification share the pipeline).

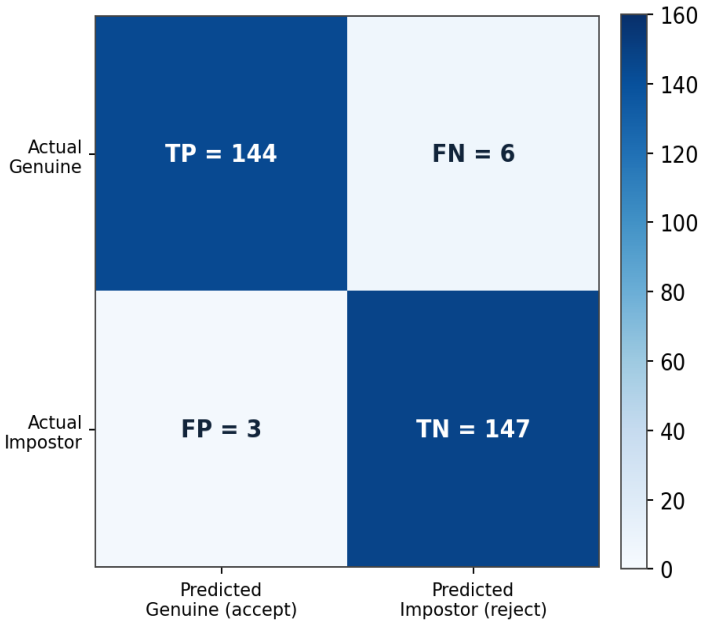

Fig 5. Confusion matrix over 300 login attempts (TP=144, FN=6, FP=3, TN=147).

The per-condition results in Figure 3 show that performance is highest for frontal, well-lit captures and degrades under low light and partial occlusion, which together account for four of the six false rejections. This pattern is consistent with the known sensitivity of face embeddings to illumination and missing facial regions. Crucially, the impostor errors (false acceptances) were concentrated in the spoof scenarios rather than in genuine-different-person comparisons, signalling that presentation attacks not legitimate look-alike are the dominant residual risk.

The evaluation is based on simulated data for a single enrolled identity and should be confirmed on larger, real, multi-subject datasets before any operational claim. The decision threshold was fixed; in practice it must be calibrated per deployment to reach a desired FAR/FRR operating point. The prototype also assumes a single, adequately sized, mostly frontal face and does not yet incorporate liveness detection.

Three risk classes warrant emphasis. Lighting: strong back-lighting, shadows, or very low illumination reduce embedding quality and inflate the False Rejection Rate, as seen in the low-light scenario. Facial angle and occlusion: large head rotations, masks, or partial coverage move the live embedding away from the enrolled template and may cause rejection. Spoofing (presentation attacks): a printed photograph, a replayed video, or a screen image of the legitimate user can be presented to the camera. Because a standard recognition pipeline reasons only about appearance, it may accept a sufficiently faithful spoof this is the principal security weakness of a recognition-only system and the reason liveness detection is essential for high-assurance use.

From a security-engineering perspective, the threshold τ defines the operating point on the FAR/FRR trade-off and must be chosen according to the threat model: security-critical systems should bias toward a low FAR, accepting more genuine rejections. Client-side processing improves privacy because biometric data and embeddings can remain on the device, but it also means template storage and integrity must be protected locally. Finally, because biometrics cannot be "reset" like a password once compromised, template protection (e.g., encryption or cancellable biometrics) and anti-spoofing are not optional add-ons but core requirements for production deployment.

5. Conclusion

This paper presented the design, implementation, and evaluation of a face recognition-based login security system using deep learning. The system combines a CNN face detector, a FaceNet-style 128-dimensional embedding, and a Euclidean-distance threshold to perform one-to-one verification, and it was realised as a fully client-side prototype using HTML, JavaScript, and face-api.js. In a simulated single-identity evaluation over 300 attempts, the system achieved 97.0% accuracy, 97.96% precision, 96.0% recall, a 96.97% F1-score, a 2.0% False Acceptance Rate, a 4.0% False Rejection Rate, and a 1.28-second average login time.

Contributions. The work delivers a clearly specified and reproducible login architecture, a self-contained browser implementation requiring no server-side inference, and a complete biometric evaluation including a confusion matrix and per-condition analysis that makes the security trade-offs explicit.

Development potential. The architecture is a foundation for higher-assurance systems: calibrating the threshold per deployment, enrolling multiple templates per user, and adding liveness and multi-factor layers would move the prototype toward production readiness while preserving its low-friction user experience.

Acknowledgement

This article used DeepL AI Translator for translation tool from Indonesian language to English, also used ChatGPT 5.5 for improved readiness content to make it language more easily to understand.

References

Abdalla, M., & Pointcheval, D. (2005). Simple Password-Based Encrypted Key Exchange Protocols. *Lectures Notes in Computer Science*, *3376*, 191–208. https://doi.org/10.1007/978-3-540-30574-3_14

Ahmed, H. E. H., Kalash, H. M., & Farag Allah, O. S. (2007). Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images. *2007 International Conference on Electrical Engineering, ICEE*. https://doi.org/10.1109/ICEE.2007.4287293

Albdairi, A. J. A., Xiao, Z., Alkhayyat, A., Humaidi, A. J., Fadhel, M. A., Taher, B. H., Alzubaidi, L., Santamaría, J., & Al-shamma, O. (2022). Face Recognition Based on Deep Learning and FPGA for Ethnicity Identification. *Applied Sciences (Switzerland)*, *12*(5). https://doi.org/10.3390/app12052605

Arvin S. Lat, J., Xavier R. Bondoc, R., & Atienza, K. C. V. (2013). SOUL System: secure online USB login system. *Information Management & Computer Security*, *21*(2), 102–109. https://doi.org/10.1108/IMCS-08-2012-0042

Bendjillali, R., Beladgham, M., Merit, K., & Taleb-Ahmed, A. (2019). Improved Facial Expression Recognition Based on DWT Feature for Deep CNN. *Electronics*, *8*(3), 324. https://doi.org/10.3390/electronics8030324

Bose, P., & Bandyopadhyay, S. K. (2020). Facial Spots Detection Using Convolution Neural Network. *Asian Journal of Research in Computer Science*, *5*(3), 71–83. https://doi.org/10.9734/ajrcos/2020/v5i330146

Chuang, C. H., Chang, K. Y., Huang, C. S., & Jung, T. P. (2022). IC-U-Net: A U-Net-based Denoising Autoencoder Using Mixtures of Independent Components for Automatic EEG Artifact Removal. *NeuroImage*, *263*, 119586. https://doi.org/10.1016/J.NEUROIMAGE.2022.119586

Dempsey, J. (2001). Internet Security and Privacy. *International Journal of Computer Science and Information Technology Research*, *2*(3), 467–475. https://doi.org/10.1201/9781420000177.ch44

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). ArcFace: Additive Angular Margin Loss for Deep Face Recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4685–4694. https://doi.org/10.1109/CVPR.2019.00482

Elsheikh, R. A., Mohamed, M. A., Abou-Taleb, A. M., & Ata, M. M. (2024). Improved facial emotion recognition model based on a novel deep convolutional structure. *Scientific Reports*, *14*(1), 1–31. https://doi.org/10.1038/s41598-024-79167-8

Gu, Q. (2008). *Finding and Segmenting Human Faces*. http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-89283

Hammoudeh, M. A. A., Alsaykhan, M., Alsalameh, R., & Althwaibi, N. (2022). Computer Vision: A Review of Detecting Objects in Videos – Challenges and Techniques. *International Journal of Online and Biomedical Engineering*, *18*(1), 15–27. https://doi.org/10.3991/ijoe.v18i01.27577

Ibrahim, R. F., Yhiea, N. M., Mohammed, A. M., & Mohamed, A. M. (2023). *Pleural Effusion Detection Using Machine Learning and Deep Learning Based on Computer Vision BT - Proceedings of the 8th International Conference on Advanced Intelligent Systems and Informatics 2022* (A. E. Hassanien, V. Snášel, M. Tang, T.-W. Sung, & K.-C. Chang, Eds.; pp. 199–210). Springer International Publishing.

Jain, A. K., Ross, A., & Prabhakar, S. (2004). An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, *14*(1), 4–20. https://doi.org/10.1109/TCSVT.2003.818349

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, X., Yang, Z., & Wu, H. (2020). Face detection based on receptive field enhanced multi-task cascaded convolutional neural networks. *IEEE Access*, *8*, 174922–174930. https://doi.org/10.1109/ACCESS.2020.3023782

Lou, A., Guan, S., & Loew, M. H. (2021). *DC-UNet: rethinking the U-Net architecture with dual channel efficient CNN for medical image segmentation*. 98. https://doi.org/10.1117/12.2582338

Mostafa, A. M., Ezz, M., Elbashir, M. K., Alruily, M., Hamouda, E., Alsarhani, M., & Said, W. (2023). Strengthening Cloud Security: An Innovative Multi-Factor Multi-Layer Authentication Framework for Cloud User Authentication. *Applied Sciences 2023, Vol. 13, Page 10871*, *13*(19), 10871. https://doi.org/10.3390/APP131910871

Muller, N., Magaia, L., & Herbst, B. M. (2004). Singular value decomposition, eigenfaces, and 3D reconstructions. *SIAM Review*. https://doi.org/10.1137/S0036144501387517

Mutneja, V., & Singh, S. (2018). GPU accelerated face detection from low resolution surveillance videos using motion and skin color segmentation. *Optik*, *157*, 1155–1165. https://doi.org/10.1016/J.IJLEO.2017.11.188

Rahim, R. (2017). 128 Bit Hash of Variable Length in Short Message Service Security. *International Journal of Security and Its Applications*, *11*(1), 45–58. https://doi.org/10.14257/ijsia.2017.11.1.05

Rahim, R., Afriliansyah, T., Winata, H., Nofriansyah, D., Ratnadewi, & Aryza, S. (2018). Research of Face Recognition with Fisher Linear Discriminant. *IOP Conference Series: Materials Science and Engineering*, *300*, 012037. https://doi.org/10.1088/1757-899X/300/1/012037

Schmidt, D., & Jaeger, T. (2013). Pitfalls in the automated strengthening of passwords. *Proceedings of the 29th Annual Computer Security Applications Conference on - ACSAC '13*, 129–138. https://doi.org/10.1145/2523649.2523651

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. https://doi.org/10.1109/CVPR.2015.7298682

Seprtitahara. (2012). *Systems Face Recognition (Face Recognition) Using Hidden Method Markov Model (HMM)*.

Sun, X., Wu, P., & Hoi, S. C. H. (2018). Face detection using deep learning: An improved faster RCNN approach. *Neurocomputing*, *299*, 42–50. https://doi.org/10.1016/J.NEUCOM.2018.03.030

Taigman, Y., Yang, M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 1701–1708. https://doi.org/10.1109/CVPR.2014.220

Yan, A., & Cheng, Z. (2024). A Review of the Development and Future Challenges of Case-Based Reasoning. *Applied Sciences*, *14*(16), 7130. https://doi.org/10.3390/app14167130

Yu, B., & Tao, D. (2019). Anchor Cascade for Efficient Face Detection. *IEEE Transactions on Image Processing*, *28*(5), 2490–2501. https://doi.org/10.1109/TIP.2018.2886790

Zhang, K., Zhang, Z., Wang, H., Li, Z., Qiao, Y., & Liu, W. (2017). Detecting Faces Using Inside Cascaded Contextual CNN. *Proceedings of the IEEE International Conference on Computer Vision*, *2017-Octob*, 3190–3198. https://doi.org/10.1109/ICCV.2017.344