International Journal Software Engineering and Computer Science (IJSECS)
6 (2), 2026, 658-671
Published Online August 2026 in IJSECS (http://www.journal.lembagakita.com/index.php/ijsecs)
P-ISSN: 2776-4869, E-ISSN: 2776-3242. DOI: https://doi.org/10.35870/ijsecs.v6i2.7353.

RESEARCH ARTICLE **Open Access**

Implementation of One-to-One Face Matching for an Internal Face Verification Server in a Mobile Attendance System at the Secretariat General of DPD RI

Anang Lesmana [1*], **Agus Sulistyanto** [2], **Anton Zulkarnain Sianipar** [3]

[1*,2,3] STMIK Jayakarta, Central Jakarta City, Special Capital Region of Jakarta, Indonesia.

**Corresponding author*: 24576001@stmik.jayakarta.ac.id.

Received: June 1, 2026; Accepted: June 28, 2026; Published: August 1, 2026.

Abstract: The mobile attendance system at the Secretariat General of the Regional Representative Council of the Republic of Indonesia previously relied on Google ML Kit for face verification services. This approach introduced challenges related to reliance on third-party services, limited configuration flexibility, and institutional control over biometric data. This study aims to develop an internal Face Verification Server using the One-to-One Face Matching method as an alternative to third-party face verification services. The system was developed using the Waterfall software development methodology, covering requirements analysis, system design, implementation, testing, and maintenance. Face verification is performed by comparing a captured facial image with a stored facial embedding associated with an employee identification number. Quantitative evaluation was conducted using 40 verification samples, consisting of 20 genuine and 20 impostor samples, with a threshold value of 0.5. The evaluation achieved 100% accuracy, a False Acceptance Rate (FAR) of 0%, and a False Rejection Rate (FRR) of 0% on the tested samples. The results indicate that the proposed system successfully distinguished matching and non-matching facial images according to the predefined threshold. The internal deployment also reduces reliance on third-party services and enables the face verification service to operate within the institution's internal network infrastructure. This study contributes an independently managed Face Verification Server architecture for mobile attendance systems in a government institutional environment.

Keywords: Face Verification; One-to-One Face Matching; Mobile Attendance; Face Recognition; Python.

1. Introduction

The rapid development of information technology has transformed the way organizations and government institutions manage administrative and operational activities. Information technology infrastructure, which includes hardware, software, data storage, and communication technologies, plays an important role in supporting organizational performance and operational efficiency (Fauzi *et al.*, 2022). The use of information technology has also encouraged institutions to adopt digital services for various administrative activities, including employee attendance management (Segara & Nasution, 2025). Digital attendance systems can simplify the recording and management of attendance data while providing more practical access for employees and administrators. Attendance management is an important part of human resource administration in government institutions because attendance records are related to employee discipline, administrative processes, and performance monitoring. Conventional attendance methods may require manual recording and verification, which can result in recording errors and opportunities for attendance fraud. The use of facial recognition in attendance systems provides an alternative approach for verifying employee identity based on facial characteristics. Facial recognition technology has been applied in various attendance and authentication systems because it can support automated identity verification without requiring physical identification media such as identification cards or fingerprint scanners (Oshin *et al.*, 2025). The use of biometric technology in attendance systems, however, also introduces requirements related to the management and protection of biometric information. Facial data is associated with an individual's identity and therefore requires appropriate protection throughout the processes of collection, processing, storage, and verification. An internal biometric authentication architecture can provide greater organizational control over biometric data management and reduce reliance on external service providers (Sun *et al.*, 2022). Such an architecture can also allow an organization to determine how biometric data and facial embeddings are processed and stored within its own infrastructure. However, the effectiveness of this approach depends on the system architecture, implementation, and security mechanisms applied by the organization.

At the Secretariat General of the Regional Representative Council of the Republic of Indonesia (Sekretariat Jenderal DPD RI), a mobile-based attendance system has been implemented to support employee attendance recording using face verification technology. The existing verification process relies on Google ML Kit as a third-party service. This dependency presents several operational and administrative considerations, including limited control over the verification mechanism, dependence on external services, and the need to manage biometric data within an appropriate organizational environment. In addition, the use of external services may introduce dependencies on network connectivity when verification services require communication beyond the organization's internal infrastructure.

Network connectivity is particularly relevant to mobile attendance because employees may access the attendance system under different network conditions. Variations in network quality can affect communication between the attendance application and the verification service and may influence the overall attendance process. From the perspective of biometric data management, maintaining the processing and storage of facial embeddings within organizational infrastructure can provide greater control over where the data is handled and which systems are involved in the verification process. Therefore, an internal face verification service represents a potential alternative for organizations that require greater control over their biometric verification infrastructure. To address these requirements, this study proposes the development of an internal Face Verification Server deployed within the infrastructure of the Secretariat General of DPD RI. The proposed system uses the One-to-One Face Matching approach, in which a facial image captured during an attendance transaction is compared with a single facial reference associated with the employee's claimed identity. This approach differs from one-to-many face identification, which requires a submitted facial image to be compared with multiple facial templates in a database. Because the identity of the employee is already provided through the employee identification number, the verification process in the proposed system only needs to evaluate whether the submitted facial image corresponds to the facial embedding associated with that identity.

The proposed Face Verification Server provides two primary services: face registration and face verification. During registration, the submitted facial image is processed to generate a facial embedding, which is stored as reference data associated with the employee's identification number. During verification, the facial embedding generated from the submitted image is compared with the stored reference embedding using the One-to-One Face Matching method. The resulting comparison is used to determine whether the submitted face matches the registered facial data. The verification response is then returned to the attendance backend server and used to determine whether the attendance request can proceed. This architecture allows the face verification process to be separated from the mobile attendance application while remaining within the organization's internal infrastructure. Previous studies have examined facial recognition and face verification technologies for attendance and authentication applications. Existing research has addressed different recognition approaches, including Local Binary Pattern Histogram (LBPH), Eigenface, and convolutional neural network-based methods, as well as their implementation in attendance applications. The comparison of previous studies presented in this research shows that much of the existing work focuses on recognition algorithms, attendance applications, and recognition accuracy. However, comparatively limited attention is given to the development of an independently managed face verification service that is deployed within an organization's internal infrastructure and integrated with an existing government attendance system. This distinction forms the main research gap addressed by the present study.

Based on this gap, this research focuses on designing and implementing an internal Face Verification Server using the One-to-One Face Matching method to support the mobile attendance system at the Secretariat General of DPD RI. The proposed system is intended to provide an alternative to the existing third-party face verification service while giving the organization greater control over the verification process and the management of facial embeddings. The system is implemented as an internal service that supports face registration and verification and communicates with the attendance backend through the organization's local network infrastructure. Through this implementation, the study evaluates the feasibility of deploying an internally managed face verification service for a government mobile attendance system.

2. Related Work

2.1 Face Verification

Face verification is a biometric process used to determine whether two facial representations belong to the same individual. The decision is generally determined using a predefined threshold applied to a similarity or distance measure. In a distance-based system, a value that satisfies the predefined threshold is classified as a match, whereas a value beyond the threshold is classified as a non-match (Solomon *et al.*, 2022). A face verification system generally consists of several processing stages. The process begins with facial image acquisition, followed by face detection to locate the facial region in the captured image. The detected face is then processed to extract facial features and represent them as a numerical feature vector or facial embedding. The resulting representation is compared with a stored reference template to produce a similarity or distance value, which is subsequently used to determine the verification result. The performance of face verification can be influenced by several factors, including facial pose, image resolution, illumination, occlusion, and overall image quality. Variations in these factors may affect the quality of the extracted facial representation and consequently influence the verification decision, particularly when images are captured under unconstrained conditions (Perez-Montes *et al.*, 2023). Face verification has been applied in various domains, including workforce attendance, access control, and electronic authentication systems. These applications use facial characteristics as an alternative to physical credentials and other authentication mechanisms. In attendance systems, face verification can be used to confirm whether the person submitting an attendance request corresponds to the claimed identity.

2.2 Face Recognition

Face recognition is a biometric technology that analyzes facial characteristics to determine or verify an individual's identity. Facial information can be obtained from static images or video streams and processed to identify distinctive characteristics and generate a representation that can be compared with registered facial data. The technology has been applied in areas such as security, access control, attendance management, and user authentication (Sulaiman *et al.*, 2024). Traditional face recognition systems generally consist of face detection, feature extraction, and identity matching stages. Earlier approaches relied on manually designed features and conventional recognition algorithms, whereas recent approaches increasingly use machine learning and deep learning techniques. Convolutional Neural Networks (CNNs), for example, can learn hierarchical facial representations from image data and have been widely used in modern face recognition systems (Alobaidy *et al.*, 2024). Face recognition can be implemented through different operational tasks. Face identification determines which registered individual corresponds to a submitted facial image, generally by comparing the image against multiple registered identities. In contrast, face verification determines whether a submitted facial image corresponds to a specific claimed identity. This distinction is important because the proposed system does not perform identification across the entire employee database. Instead, it performs verification against the facial representation associated with a specific employee identification number. The application of face recognition has expanded across authentication, access control, surveillance, and attendance systems. Its use provides a contactless approach to identity verification and can support automated authentication processes (Gumanof *et al.*, 2025; Yemima & Girsang, 2023). However, the effectiveness of a face recognition system depends on the recognition method, image conditions, reference data, and verification criteria used in the implementation.

2.3 One-to-One Face Matching

One-to-One Face Matching, also referred to as 1:1 face verification, is a biometric verification approach that compares a captured facial image with a single facial reference associated with a claimed identity. The objective is to determine whether both facial representations belong to the same individual. The verification decision is based on the similarity or distance between the two representations and is affected by factors such as image quality, environmental conditions, and the selected decision threshold (Fysh & Bindemann, 2023). In a 1:1 verification process, the identity claim is provided before the facial comparison is performed. For example, an employee identification number can be used to retrieve the corresponding registered facial embedding. The facial embedding generated from the submitted image is then compared only with the reference embedding associated with that employee. This process differs from 1:N face identification, in which a submitted facial image must be compared with multiple registered facial templates to determine the most likely identity. The distinction between 1:1 verification and 1:N identification is particularly relevant to the proposed attendance system. Because the employee identification number is available when an attendance request is submitted, the system does not need to search the entire facial database. Instead, it only needs to determine whether the submitted face corresponds to the reference representation associated with the claimed employee identity. This reduces the number of comparisons required during each verification transaction. Automated facial recognition systems can support both identification and verification functions. In identification, a captured facial image is compared with a collection of registered facial images, whereas verification compares the captured image with a designated reference image to validate a claimed identity (Carragher *et al.*, 2024). The present study focuses specifically on the verification task and implements it through a One-to-One Face Matching approach.

2.4 Face Liveness Detection

Face liveness detection is an additional mechanism used to distinguish a live person from artificial or reproduced facial inputs. Presentation attacks may involve photographs, video playback, or other forms of artificial facial representations. Liveness detection can therefore provide an additional layer of protection by assessing whether the facial input originates from a live subject during the verification process (Khairnar *et al.*, 2023). Face verification and liveness detection address different aspects of biometric authentication. Face verification determines whether the submitted facial representation corresponds to the claimed identity, whereas liveness detection evaluates whether the submitted biometric sample originates from a live subject rather than a presentation artifact. Therefore, liveness detection can complement a face verification system when protection against presentation attacks is required. The current Face Verification Server focuses on One-to-One Face Matching and does not include an active liveness detection mechanism. Therefore, liveness detection is considered a potential improvement for future development rather than a component evaluated in the present study. [cek referensi] The reference used to support this discussion should be verified to ensure that it directly addresses facial liveness detection and presentation attacks.

2.4 Face Liveness Detection

Several studies have investigated facial recognition and face verification for authentication and attendance applications. Previous research has examined different recognition approaches, including Local Binary Pattern Histogram (LBPH), Eigenface, convolutional neural networks, and facial vector representations. These studies demonstrate the applicability of facial biometrics for automated attendance recording and identity verification.

Table 1. Comparison of Previous Studies

Study	Method	Main Problem	Main Result
Likario *et al.* (2024)	LBPH Face Recognition	Manual attendance processes were susceptible to human errors and data inconsistencies.	Developed a facial recognition-based attendance approach for automated attendance monitoring.
Fauzi *et al.* (2022)	Eigenface (PCA)	Conventional attendance systems were inefficient and vulnerable to attendance fraud.	Applied Eigenface-based facial recognition to support attendance management.
Nicholas & Al Rivan (2025)	CNN with InceptionV3 architecture	Conventional attendance systems lacked flexibility and were vulnerable to fraud, while face recognition systems faced challenges related to accuracy, scalability, and real-time performance.	Developed a web-based attendance system using facial recognition.
Jamalapuram *et al.* (2024)	128-dimensional facial vectors stored in SQLite	Manual attendance systems were inefficient, while recognition performance could be affected by lighting, pose, and image quality.	Developed an automated attendance system using facial feature vectors and SQLite-based attendance storage.
Hernowo & Laksana (2024)	Eigenface-based face recognition on Android	The existing fingerprint attendance system required manual data processing and did not support real-time attendance monitoring.	Implemented an Android-based attendance system using face recognition.
Proposed System	One-to-One Face Matching	Dependence on Google ML Kit, the need for greater organizational control over biometric data, and reliance on external verification services.	Developed an internal Face Verification Server using Python and Flask to support 1:1 biometric verification within the organization's network infrastructure.

Based on the reviewed studies, previous research has primarily focused on face recognition algorithms and their application to attendance systems. The studies reviewed in this research provide evidence of the use of different facial recognition methods for automated attendance and identity verification. However, comparatively less attention is given to the development of an independently managed face verification service that operates within an organization's internal infrastructure and is integrated with an existing government attendance system. The proposed study addresses this gap by developing an internal Face Verification Server using the One-to-One Face Matching method. The server is designed to provide face registration and verification services for the mobile attendance system at the Secretariat General of DPD RI. Unlike approaches that depend on external face verification services, the proposed architecture places the verification service within the organization's internal network infrastructure.

3. Methodology

3.1 Research Framework

This research employs the Waterfall model as the software development framework for developing the proposed Face Verification Server. The Waterfall model was selected because it provides a systematic and sequential development process consisting of requirements analysis, system design, implementation, testing, and maintenance. These stages were used to develop the face registration and face verification services required by the mobile attendance system at the Secretariat General of DPD RI.

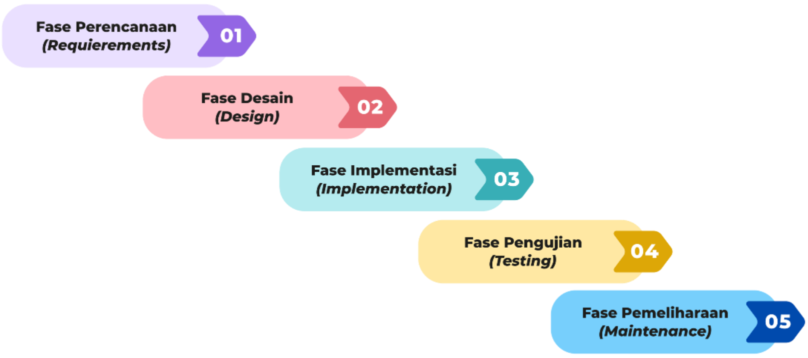


Figure 1. Waterfall Method

1) Requirements
The requirements phase was conducted to identify system requirements based on the limitations of the face verification service used by the existing mobile attendance system. The main requirement was the development of an internal Face Verification Server that provides face registration and face verification services using the One-to-One Face Matching method. A web-based monitoring dashboard was also included as a system requirement to support monitoring of face registration and verification activities performed by users.
2) Design
The design phase focused on the face registration workflow, face verification workflow, and monitoring dashboard interface. The registration workflow was designed to define the process of generating and storing employee facial embeddings, while the verification workflow was designed to define the comparison process between the submitted facial embedding and the stored reference embedding. The web-based monitoring dashboard was designed to display face registration and verification activities processed by the server.
3) Implementation
The implementation phase involved developing the Face Verification Server using Python and the Flask framework. The system was implemented as a web API that provides two primary services: face registration and face verification. Facial feature extraction was implemented using the face_recognition library, while OpenCV was used for image processing. The resulting facial embeddings were stored as reference data associated with employee identification numbers. A web-based monitoring dashboard was also developed to display registration and verification activities processed by the server.
4) Testing
The testing phase was conducted to verify that the developed functions operated according to the defined system requirements. Black Box Testing was applied to the face registration feature, face verification feature, and monitoring dashboard. Different input scenarios were provided, and the resulting outputs were observed to determine whether each function produced the expected response. In addition to

functional testing, face verification samples were evaluated quantitatively to assess the system's ability to distinguish matching and non-matching facial images.

5) Maintenance
The maintenance phase was conducted to support the continued operation of the Face Verification Server after deployment. Maintenance activities included backing up the application files and stored facial embeddings, correcting errors identified during operation, and updating supporting libraries and components when necessary to maintain system stability and compatibility.

3.2 System Architecture

The proposed system architecture consists of three main components: a mobile attendance application, an attendance backend server, and a Face Verification Server. The mobile attendance application is used by employees to submit attendance requests containing a facial image and employee identification number (NIP). The attendance backend server forwards the submitted information to the Face Verification Server, where identity verification is performed using the One-to-One Face Matching method. The Face Verification Server handles face registration and face verification requests received from the attendance backend server. During verification, the system retrieves the facial embedding associated with the submitted employee identification number and compares it with the facial embedding generated from the submitted image. The verification result is then returned to the backend server and used as a basis for processing the attendance request. The services are deployed within the organization's internal infrastructure, allowing communication between the servers to be performed through the internal network.

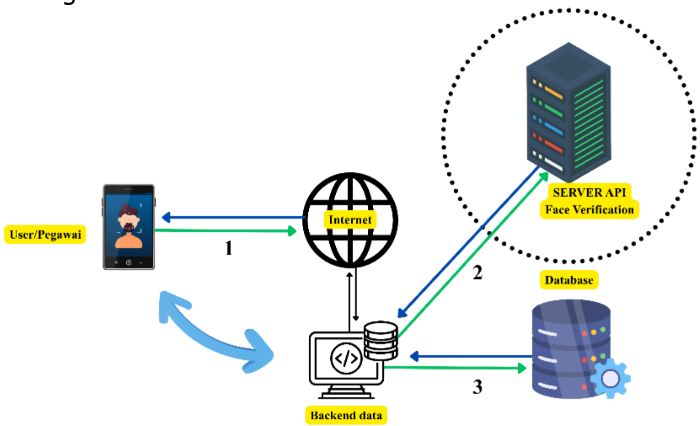


Figure 2. System Integration

The Face Verification Server was deployed within the internal infrastructure of the Secretariat General of DPD RI using Ubuntu Server 24.04.2 as the operating system. The server was equipped with an 8-core processor, 8 GB of memory, and 100 GB of storage. These resources were used to support the face registration process, facial embedding storage, and face verification services accessed through the Face Verification API. Communication between the attendance backend server and the Face Verification Server was performed through an internal local area network (LAN). This configuration allows data exchange between th systems without requiring the face verification process to rely on an external verification service.

Figure 3. Local Network Topology

3.3 Tools and Technologies

Table 2. Tools and Technologies Used in the System

Tool/Technology	Version	Function
Python	3.10.18	Server-side application development
Flask	3.1.1	Web API development
face_recognition	1.3.0	Facial feature extraction and face matching

OpenCV	4.11.0	Image processing
NumPy	2.2.6	Numerical array processing and facial embedding storage
HTML, CSS, and JavaScript	HTML5, CSS3, ES6	Web-based monitoring dashboard interface
Ubuntu Server	24.04.2	Operating system for system deployment
Visual Studio Code	1.123.0	Source code development and system implementation

3.4 Face Registration Process

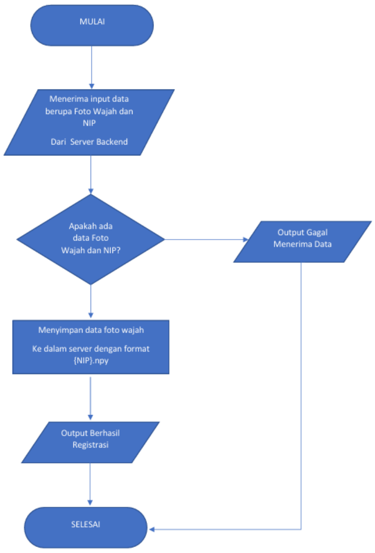


Figure 4. Registration Flow

The verification process begins by checking whether a facial embedding associated with the submitted employee NIP is available on the server. If no corresponding embedding is found, the system returns a response indicating that face registration has not been completed. If the reference embedding is available, the system extracts facial features from the submitted image and generates a facial embedding. The Euclidean distance between the stored reference embedding and the generated verification embedding is then calculated. The resulting distance is compared with a predefined threshold of 0.5. A distance value of $\leqslant 0.5$ is classified as a match, whereas a distance value of > 0.5 is classified as a non-match.

3.5 Face Verification Process

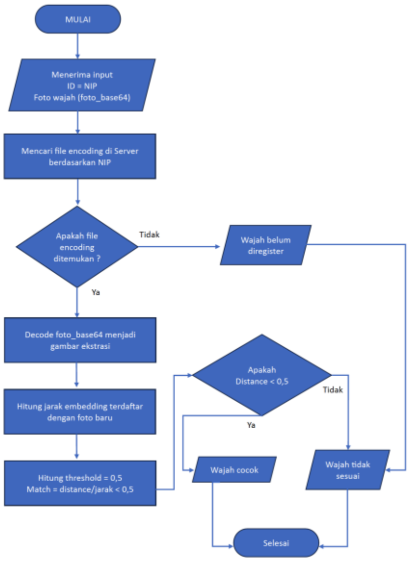


Figure 5. Verification Flow

The verification process begins by checking whether a facial embedding associated with the submitted employee NIP is available on the server. If no corresponding embedding is found, the system returns a response indicating that face registration has not been completed. If the reference embedding is available, the system extracts facial features from the submitted image and generates a facial embedding. The Euclidean distance between the stored reference embedding and the generated verification embedding is then calculated. The resulting distance is compared with a predefined threshold of 0.5. A distance value of ⩽ 0.5 is classified as a match, whereas a distance value of > 0.5 is classified as a non-match. Face verification systems commonly use a threshold to distinguish matching and non-matching facial representations. The selected threshold determines the decision boundary between genuine and imposter pairs and can therefore affect verification performance (Diez-Tomillo *et al.*, 2023). In the proposed system, the Euclidean distance between the registered facial embedding and the verification facial embedding is used as the decision measure. Several candidate threshold values, namely 0.4, 0.5, and 0.6, were considered during preliminary testing. A threshold of 0.5 was selected for implementation based on the preliminary evaluation. If quantitative evaluation results are available, the selection of the threshold should be supported by the corresponding performance measures for each candidate value. A threshold of 0.4 is more restrictive and may increase the possibility of rejecting genuine users, whereas a threshold of 0.6 is more permissive and may increase the possibility of accepting non-matching users. The threshold of 0.5 was selected as the basis for the verification process and was subsequently applied throughout the system.

Table 3. Threshold Selection Criteria

Threshold	Observation
0.4	More restrictive and may increase the possibility of rejecting genuine users
0.5	Selected for implementation based on preliminary evaluation
0.6	More permissive and may increase the possibility of accepting non-matching users

4. Result and Discussion

4.1 Results

A quantitative performance evaluation was conducted to assess the ability of the proposed Face Verification Server to distinguish between genuine and impostor verification attempts. The evaluation used verification data from 20 employees registered in the system. A threshold of 0.5 was applied, where distance values of ⩽ 0.5 were classified as matches and distance values of > 0.5 were classified as non-matches. A total of 40 verification samples were evaluated, consisting of 20 genuine samples and 20 impostor samples. The evaluation results were analyzed using Accuracy, False Acceptance Rate (FAR), and False Rejection Rate (FRR).

Table 4. Verification Test Dataset

Parameter	Value
Registered Employees	20
Genuine Samples	20
Impostor Samples	20
Total Samples	40
Threshold	0.5

The verification evaluation showed that all 20 genuine samples were correctly classified as matches, while all 20 impostor samples were correctly classified as non-matches. Based on these results, the system produced 20 true positives (TP), 20 true negatives (TN), 0 false positives (FP), and 0 false negatives (FN).

Table 5. Verification Performance Results

Metric	Value
Accuracy	100%
FAR	0%
FRR	0%

The results indicate that the proposed system correctly classified all 40 verification samples evaluated in this test. The resulting FAR and FRR values were both 0%, indicating that no impostor sample was incorrectly accepted and no genuine sample was incorrectly rejected within the evaluated dataset. [cek data] These results apply only to the verification samples used in this study and should not be interpreted as evidence of general accuracy under broader operating conditions. The face registration endpoint was integrated with the attendance backend server to transmit employee identification numbers (NIP) and facial images to the Face Verification Server. The received data were processed to generate facial embeddings, which were subsequently stored on the server as reference data for future verification.

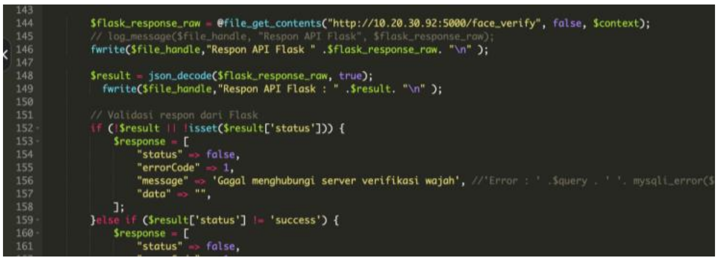
```
$flask_response_raw = @file_get_contents("http://10.20.30.92:5000/register_face", false, $context);
log_to_file($fh, "Respon API Flask => " . $flask_response_raw);

$flask_response = json_decode($flask_response_raw, true);

if (!$flask_response || !isset($flask_response['status'])) {
    json_response($fh, [
        "status" => false,
        "errorCode" => 1,
        "message" => 'Tidak dapat menghubungi server Flask',
        "data" => null,
    ]);
}
```

Figure 6. Face Registration Endpoint Implementation in the Backend Server

The face verification endpoint was integrated with the attendance backend server to transmit employee identification numbers (NIP) and facial images to the Face Verification Server. The verification result returned by the Face Verification Server was then used as the basis for identity validation during the attendance process.



```
        $flask_response_raw = @file_get_contents("http://10.20.30.92:5000/face_verify", false, $context);
        // log_message($file_handle, "Respon API Flask", $flask_response_raw);
        fwrite($file_handle,"Respon API Flask " .$flask_response_raw. "\n" );

        $result = json_decode($flask_response_raw, true);
          fwrite($file_handle,"Respon API Flask : " .$result. "\n" );

        // Validasi respon dari Flask
        if (!$result || !isset($result['status'])) {
            $response = [
                "status" => false,
                "errorCode" => 1,
                "message" => 'Gagal menghubungi server verifikasi wajah', //'Error : ' .$query . ' '. mysqli_error($
                "data" => "",
            ];
        }else if ($result['status'] != 'success') {
            $response = [
                "status" => false,
```

Figure 7. Face Verification Endpoint Implementation in the Backend Server

The face registration functionality was tested through the mobile attendance application. During registration, an employee submitted an employee identification number (NIP) and a facial image through the application. The submitted data were transmitted to the attendance backend server and subsequently forwarded to the Face Verification Server through the registration endpoint. The system processed the facial image, generated a facial embedding, and stored the resulting embedding on the server as reference data for subsequent verification.

Table 6. Registration Test Scenario

Test Module	Face Registration Endpoint (/register_face)
Scenario	An employee performs face registration through the mobile attendance application
Input	Employee identification number (NIP) and facial image
Expected Output	The system receives the NIP and facial image, generates a facial embedding, and stores the embedding data on the server
Result	The facial embedding data were successfully stored on the server
Status	Success

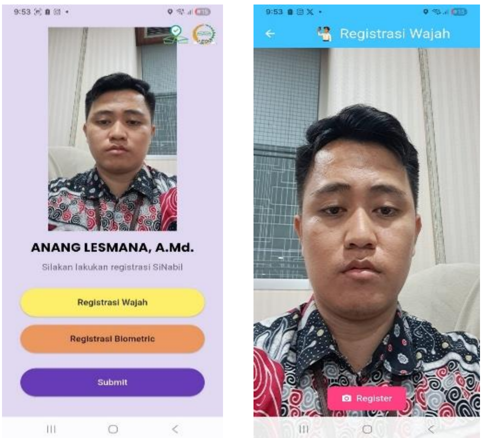


Figure 8. Registration Process in Mobile Application

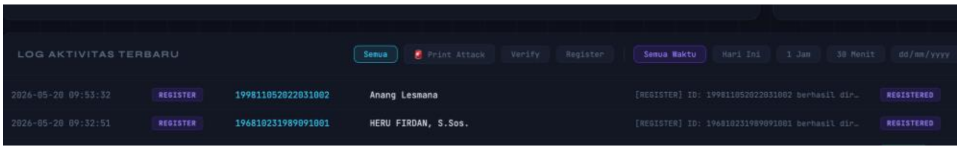


Figure 9. Successful Registration Log

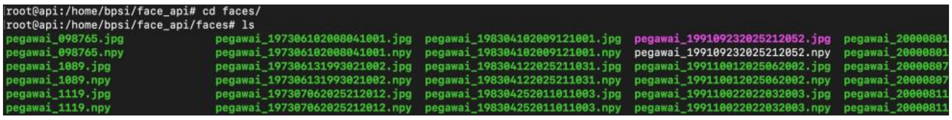


Figure 10. Face Embedding List

The face verification functionality was also tested through the mobile attendance application. During the verification process, an employee submitted an employee identification number (NIP) and a facial image. The data were transmitted to the attendance backend server and subsequently forwarded to the Face Verification Server through the verification endpoint. The system checked whether a facial embedding associated with the submitted NIP was available. When the reference embedding was found, the system generated a facial embedding from the submitted image and calculated the Euclidean distance between the two embeddings. The resulting distance was compared with the predefined threshold of 0.5. A distance value of $\leqslant 0.5$ was classified as a match, while a value of > 0.5 was classified as a non-match. The verification result was then returned to the attendance backend server and displayed through the mobile application.

Table 7. Verification Test Scenario

Test Module	Face Verification Endpoint (/face_verify)
Scenario	An employee performs face verification when recording check-in or check-out attendance through the mobile attendance application
Input	Employee identification number (NIP) and facial image
Expected Output	The system receives the NIP and facial image, calculates the Euclidean distance, and determines whether the submitted face matches the registered facial data
Result	The system successfully verified the facial image and returned the expected verification result
Status	Success

Figure 11. Verification Process in Mobile Application

Figure 12. Successful Verification Log

The monitoring dashboard was also tested to verify its ability to display face registration and verification activities processed by the Face Verification Server. The dashboard successfully displayed the recorded activities and their corresponding verification results for administrative monitoring.

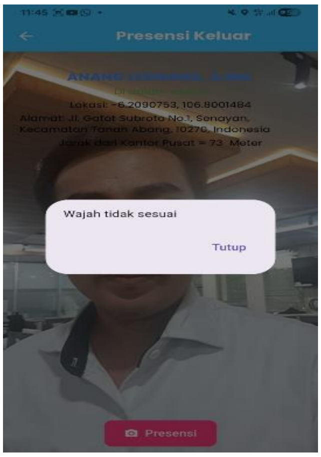


Figure 13. Verification Test Using a Different Face

Figure 14. Failed Verification Log

The monitoring dashboard was also tested to verify its ability to display face registration and verification activities processed by the Face Verification Server. The dashboard successfully displayed the recorded activities and their corresponding verification results for administrative monitoring.

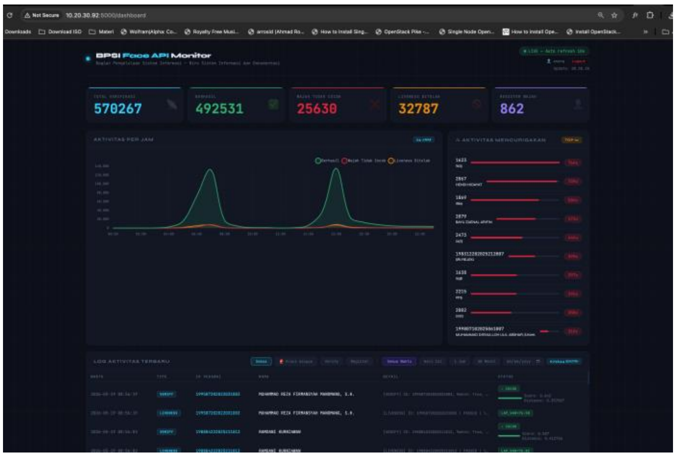


Figure 15. Dashboard Monitoring

4.2 Discussion

The results demonstrate that the proposed Face Verification Server was successfully implemented and integrated with the mobile attendance system at the Secretariat General of DPD RI. The developed system successfully performed face registration and face verification using the One-to-One Face Matching approach. In the quantitative evaluation, all 40 verification samples were correctly classified at the selected threshold of 0.5, resulting in an accuracy of 100%, FAR of 0%, and FRR of 0%. These results indicate that the system performed correctly for the genuine and impostor samples included in the evaluation dataset. The proposed internal Face Verification Server was developed to reduce the system's dependence on the face verification service previously used in the mobile attendance application. The proposed architecture provides direct control over the verification threshold and face matching process because these functions are implemented on an internally managed server. In the evaluation conducted in this study, a threshold of 0.5 correctly classified all 40 verification samples, consisting of 20 genuine and 20 impostor samples. This result demonstrates the performance of the selected threshold on the evaluated dataset but does not establish its general performance under different datasets or operating conditions.

Another difference between the proposed system and the previous implementation is the location of biometric data processing and storage. In the proposed architecture, facial embeddings are generated and stored on the organization's internal Face Verification Server, while communication between the attendance backend server and the Face Verification Server is performed through the internal network. This configuration reduces dependence on an external face verification service and provides the organization with greater control over the processing and storage of facial embeddings. The proposed system also separates the face verification service from the mobile attendance application and provides face registration and verification functions through an internal API. The reviewed studies primarily focus on the application of face recognition methods in attendance systems, including Eigenface-based approaches, convolutional neural networks, and facial vector representations (Hernowo & Laksana, 2024; Nicholas & Al Rivan, 2025). In contrast, this research focuses on the implementation and deployment of a dedicated Face Verification Server within an existing government mobile attendance infrastructure. This approach allows the face verification process to be managed independently from the attendance application while maintaining communication through the organization's internal network.

The internal deployment also provides practical benefits for system management. Facial embeddings can be processed and stored within the organization's infrastructure, while communication between the attendance backend server and the Face Verification Server can be performed through the internal LAN. The architecture also allows the verification threshold and matching process to be configured directly within the organization's server environment. Therefore, the proposed system provides a potential alternative for government environments that require greater organizational control over biometric data processing and face verification services. However, the present study does not include a dedicated security assessment or a direct performance comparison with the previous Google ML Kit implementation. Therefore, the results should not be interpreted as quantitative evidence that the proposed architecture is more secure, faster, or more accurate than the previous implementation.

5. Conclusion and Recommendations

This study demonstrates the successful design and implementation of an internal Face Verification Server to support the mobile attendance system at the Secretariat General of DPD RI. The developed system provides face registration and verification functionalities using the One-to-One Face Matching approach and has been integrated with the attendance backend server. The system performs verification by comparing a facial embedding generated from a submitted facial image with the corresponding reference embedding stored on the server. In the quantitative evaluation, the system correctly classified all 40 verification samples, consisting of 20 genuine samples and 20 impostor samples, using a threshold of 0.5. The evaluation resulted in an accuracy of 100%, a False Acceptance Rate (FAR) of 0%, and a False Rejection Rate (FRR) of 0%. These results indicate that the proposed system performed correctly for the verification samples evaluated in this study. The main contribution of this research is the development of an internally managed Face Verification service as an alternative to the previous verification mechanism based on Google ML Kit. The proposed system processes and stores facial embeddings within the organization's internal infrastructure and performs identity verification using the One-to-One Face Matching approach. Communication between the attendance backend server and the Face Verification Server is also conducted through the internal local network. This architecture reduces dependence on external face verification services and provides greater organizational control over biometric data processing, storage, and verification parameters.

Several limitations should be considered when interpreting the results. First, the system was evaluated within a single organizational environment, namely the Secretariat General of DPD RI, using a relatively limited verification dataset consisting of 40 samples. Therefore, the results cannot be generalized to other operational environments or larger and more diverse datasets without further evaluation. Second, the current implementation focuses on facial embedding comparison and does not incorporate dedicated liveness detection mechanisms to address presentation attacks such as the use of printed images or video replays. Third, the system is currently deployed on a single server instance, which may limit scalability and availability when the number of verification requests increases significantly.Future research may address these limitations by evaluating the system using larger and more diverse datasets under different environmental conditions. Liveness detection techniques may also be incorporated to provide an additional layer of protection against presentation attacks. Additional verification security mechanisms, such as temporary account locking after multiple consecutive failed attempts, may be considered to reduce repeated unauthorized verification attempts. Furthermore, the server architecture may be extended to support distributed or microservices-based deployment to improve scalability and availability under higher workloads. Future development may also enhance the monitoring dashboard with reporting and data export functions to support administrative and operational requirements.

References

Alobaidy, M. A. A., Yosif, Z. M., Alsoufi, M. S., & Al-Ashqar, S. (2024). Attendance system based on face recognition dependent on deep intelligent techniques. *Revue d'Intelligence Artificielle, 38*(3), 1009–1016. https://doi.org/10.18280/ria.380326

Carragher, D. J., Sturman, D., & Hancock, P. J. B. (2024). Trust in automation and the accuracy of human–algorithm teams performing one-to-one face matching tasks. *Cognitive Research: Principles and Implications, 9*(1). https://doi.org/10.1186/s41235-024-00564-8

Diez-Tomillo, J., Alcaraz-Calero, J. M., & Wang, Q. (2023). Dynamic-distance-based thresholding for UAV-based face verification algorithms. *Sensors, 23*(24). https://doi.org/10.3390/s23249909

Fauzi, E., Sinatrya, M. V., Ramdhani, N. D., Muhammad, Z., & Safari, R. (2022). Pengaruh kemajuan teknologi informasi terhadap perkembangan akuntansi Ruhuphy Ramadhan. *Jurnal Riset Pendidikan Ekonomi, 7*.

Fysh, M. C., & Bindemann, M. (2023). Understanding face matching. *Quarterly Journal of Experimental Psychology, 76*(4), 862–880. https://doi.org/10.1177/17470218221104476

Gumanof, I. M., Teknologi Informasi, J., Negeri Padang, P., & Barat, S. (2025). Sistem verifikasi E-KTP dengan menggunakan face recognition. *Jurnal Bitwise, 1*(2). https://jurnal-bitwise.org/

Hernowo, A., & Laksana, T. G. (2024). Pengenalan pola citra wajah untuk presensi karyawan menggunakan algoritma Eigenface berbasis Android (Studi analisis: Presensi DPP Partai Demokrat). *Jurnal Ilmiah Multidisiplin, 1*(2), 199–211. https://doi.org/10.62282/juilmu.v1i2.199-211

Jamalapuram, P., Sragvi, P. S., & Renuka, S. (2024). *Face recognition: Based attendance system*. IJFMR. https://www.ijfmr.com/

Jiang, T., Chen, X., Song, J., & Hilliges, O. (2022). *InstantAvatar: Learning avatars from monocular video in 60 seconds*. arXiv. http://arxiv.org/abs/2212.10550

Khairnar, S., Gite, S., Kotecha, K., & Thepade, S. D. (2023). Face liveness detection using artificial intelligence techniques: A systematic literature review and future directions. *Big Data and Cognitive Computing, 7*(1), Article 37. https://doi.org/10.3390/bdcc7010037

Likario, M., Hasna Nabila, H., Khoirunnisa, S., Fauziah, S., & Rosyani, P. (2024). Implementasi deteksi wajah pada sistem peresensi dengan menerapkan teknik face recognition. *JRIIN: Jurnal Riset Informatika dan Inovasi, 2*(2). https://jurnalmahasiswa.com/index.php/jriin

Nicholas, N., & Al Rivan, M. E. (2025). Facial recognition software for employee presence using convolutional neural network with InceptionV3 architecture. *Brilliance: Research of Artificial Intelligence, 5*(2), 761–772. https://doi.org/10.47709/brilliance.v5i2.6769

Oshin, O., Amenaghawon, J., Moninuola, F., & Idowu-Bismark, O. (2025). Class attendance system using facial recognition. *Ingénierie des Systèmes d'Information, 30*(6), 1589–1595. https://doi.org/10.18280/isi.300617

Perez-Montes, F., Olivares-Mercado, J., Sanchez-Perez, G., Benitez-Garcia, G., Prudente-Tixteco, L., & Lopez-Garcia, O. (2023). Analysis of real-time face-verification methods for surveillance applications. *Journal of Imaging, 9*(2), Article 21. https://doi.org/10.3390/jimaging9020021

Segara, G. K., & Nasution, P. N. (2025). Perkembangan teknologi informasi di Indonesia: Tantangan dan peluang. *Jurnal Sains Student Research, 3*(1), 21–33. https://doi.org/10.61722/jssr.v3i1.3128

Solomon, E., Woubie, A., & Cios, K. J. (2022). UFace: An unsupervised deep learning face verification system. *Electronics, 11*(23). https://doi.org/10.3390/electronics11233909

Sulaiman, K. J., Sari, I., & Ramadhanu, A. (2024). Implementasi pengolahan citra digital dalam pengenalan wajah menggunakan contrast stretching dan algoritma Viola Jones. *Indonesian Journal of Computer Science*.

Sun, Q., Wu, J., & Yu, W. (2022). BioShare: An open framework for trusted biometric authentication under user control. *Applied Sciences, 12*(21). https://doi.org/10.3390/app122110782

Sutedi, A., Indrakusumah, M. R., Deddy Supriatna, A., & Mulyani, A. (2025). Rancang bangun sistem pengelolaan inventaris barang dengan verifikasi pengguna berbasis face recognition. *Jurnal Algoritma, 22*(2). https://doi.org/10.33364/algoritma/v.22-2.2672

Yemima, S., & Girsang, B. (2023). Pentingnya regulasi khusus tentang pemanfaatan sistem Face Recognition Technology dalam peningkatan keamanan dan penegakan hukum di Indonesia. *Jurnal Hukum dan HAM Wara Sains, 2*(10).