Development of an Edge AI-Based Employee Attendance System to Improve the Reliability of IoT Attendance Systems

1st Ilham Wibawa Kusumah Dinata*
Mechatronics Engineering, Faculty of Engineering
Mayasari Bakti University
Tasikmalaya, West Java
ilhamwibawamt@gmail.com

2nd Yanti
Mechatronics Engineering, Faculty of Engineering
Mayasari Bakti University
Tasikmalaya, West Java
yanti.aiiasenja@gmail.com

3rd Indra Setiawan
Mechatronics Engineering, Faculty of Engineering
Mayasari Bakti University
Tasikmalaya, West Java
indra.stwn009@gmail.com

4th Fajar Firmansyah
Informatics Engineering, Faculty of Engineering
Mayasari Bakti University
Tasikmalaya, West Java
Ffajar1989@gmail.com

*Corresponding author: ilhamwibawamt@gmail.com
Received: 2026-09-19; **Accepted:** 2026-09-19

Abstract*—Employee attendance management is essential for improving organizational discipline and operational efficiency. Conventional attendance systems based on identity cards, fingerprint scanners, and manual recording may be affected by attendance fraud, physical contact, maintenance requirements, and communication or server interruptions. This study presents a reliability-oriented IoT employee attendance prototype integrating an ESP32-CAM, a Raspberry Pi 3 Model B+ as an edge gateway, HTTP and MQTT communication, offline buffering, Face Recognition, local data storage, and a web-based dashboard. The Raspberry Pi manages image forwarding, local buffering, MQTT communication, and synchronization, while a Flask-based server performs face detection, identity verification, attendance recording, and dashboard visualization. Face Recognition uses a Dlib-based Histogram of Oriented Gradients (HOG) detector and 128-dimensional facial embeddings with a fixed tolerance of 0.5. The initial closed-set evaluation used 212 facial samples from 11 registered employees, with all evaluated samples correctly classified under controlled conditions. The system also achieved successful HTTP transmission, database storage, dashboard synchronization, and offline-buffer recovery in the reported functional tests, with an average server response time of 1.069 s and an average MQTT delay of 2.47 ms. These results demonstrate prototype feasibility and service continuity under the tested conditions. They should not be interpreted as evidence of general biometric accuracy or large-scale deployment readiness. Additional robustness, impostor-rejection, security, privacy, and multi-user evaluations are required for broader validation.

Keywords—Edge Computing; Face Recognition; Internet of Things (IoT), MQTT Communication, Offline Buffering)

I. INTRODUCTION

The rapid advancement of Artificial Intelligence (AI), the Internet of Things (IoT), and Edge Computing has accelerated digital transformation across various sectors, including employee attendance management. Attendance records are essential for supporting employee discipline, payroll administration, productivity evaluation, and managerial decision-making. However, conventional attendance methods based on manual signatures, RFID cards, or fingerprint scanners remain vulnerable to attendance fraud, buddy punching, physical contact, hardware maintenance issues, and service interruptions caused by unstable network connectivity or server failures. These limitations can reduce operational efficiency and compromise attendance data integrity, particularly in organizations with limited information technology infrastructure or geographically distributed workplaces. Consequently, there is an increasing demand for contactless, automated, and reliable attendance systems capable of maintaining continuous operation under practical deployment conditions [1], [2], [3], [4], [5], [6], [7].

Face Recognition has emerged as a promising biometric technology for intelligent attendance systems because it enables individual identification using facial characteristics without requiring physical interaction. Compared with fingerprint- and RFID-based authentication, Face Recognition provides greater user convenience, minimizes physical contact, and reduces the possibility of identity sharing. Recent advances in deep learning and facial embedding techniques have improved recognition performance under variations in illumination, facial expression, and head pose, supporting applications in access control, surveillance, identity verification, healthcare, and attendance systems [8], [9], [10], [11], [12], [13], [14]. Nevertheless, practical deployment remains challenging because recognition performance may degrade under poor illumination, facial occlusion, motion blur, masks, and extreme head poses. Therefore, reliable attendance systems require not only accurate recognition algorithms but also robust system integration capable of maintaining stable operation under varying operational and communication conditions.

Among low-cost embedded vision platforms, the ESP32-CAM has attracted attention because it integrates an OV2640 camera module, Wi-Fi connectivity, compact dimensions, and low power consumption into a single microcontroller-based device. Previous studies have demonstrated the use of ESP32-CAM and related embedded camera platforms for IoT

Copyright © 2026 Universitas Sebelas Maret

monitoring and image-acquisition applications [3], [15]. Research on image processing at IoT edge nodes has further demonstrated the importance of considering computational and resource constraints when deploying vision-based applications on embedded platforms [16]. However, the limited computational capability and memory capacity of the ESP32-CAM make it unsuitable for executing computationally intensive Face Recognition processes efficiently. Modern Face Recognition pipelines require face detection, feature extraction, similarity computation, and identity classification, which demand greater computational resources than those typically available on low-cost microcontroller platforms [8], [10], [13].

To address these computational limitations, IoT-based attendance systems can employ distributed Edge AI architectures in which image acquisition and computationally intensive processing are assigned to different devices. In such an architecture, the ESP32-CAM can perform facial image acquisition, while a more capable computing platform manages communication, temporary storage, buffering, and forwarding. Edge computing is particularly relevant for resource-constrained applications because processing and supporting services can be positioned closer to the data source [5], [17]. The proposed study applies this principle by using the Raspberry Pi 3 Model B+ as an edge gateway while retaining Face Recognition inference on the Flask-based server.

In addition to computational limitations, communication reliability represents an important challenge in IoT-based attendance systems. Temporary network interruptions or server unavailability may interrupt attendance transactions and potentially result in missing or delayed records. MQTT is widely used in IoT applications because its publish–subscribe architecture supports lightweight asynchronous communication; however, its performance and security can vary with broker implementation, payload characteristics, and network conditions [18], [19]. Communication protocols alone do not guarantee service continuity when the destination server becomes unavailable. A local buffering and recovery mechanism is therefore required to retain data during interruptions and synchronize buffered information after connectivity is restored.

Although numerous studies have investigated Face Recognition-based attendance systems, many reported implementations primarily emphasize recognition performance and attendance functionality. Comparatively less attention is given to the continuity of attendance services when communication or server availability is temporarily disrupted. In particular, local buffering, store-and-forward operation, automatic resynchronization after reconnection, database consistency, and operational continuity are not always evaluated together as reliability requirements. This gap is relevant for low-cost IoT deployments in which the image-acquisition device has limited computational and storage resources and depends on a remote recognition service.

Accordingly, the research gap addressed in this study is the integration and experimental evaluation of a reliability-oriented attendance architecture in which an edge gateway provides a local recovery layer between the image-acquisition device and the Face Recognition server. The study does not propose a new Face Recognition algorithm. Instead, it investigates whether the combination of an ESP32-CAM, Raspberry Pi 3 Model B+ gateway, local buffering, MQTT communication, and automatic synchronization can maintain attendance-data continuity during temporary server or communication interruptions.

To address this gap, this study develops an IoT-based employee attendance prototype integrating an ESP32-CAM for facial image acquisition, a Raspberry Pi 3 Model B+ as an edge gateway, and a Flask-based Face Recognition server. The ESP32-CAM captures facial images and transmits them through HTTP. The Raspberry Pi manages image forwarding, MQTT communication, local buffering, synchronization, LCD notification, and local data services, while the Flask server performs face detection, identity verification, attendance recording, and dashboard visualization. When the recognition server becomes temporarily unavailable, the edge gateway retains captured data locally and forwards the buffered data after communication is restored.

The novelty of this study is therefore positioned at the system-architecture and reliability level rather than at the algorithmic level. The Raspberry Pi 3 Model B+ is used not only as a communication gateway but also as a local recovery layer that implements offline buffering and store-and-forward operation. The architecture integrates HTTP-based image transmission, MQTT-based result synchronization, local storage, evidence management, and automatic recovery in a single low-cost attendance prototype. The contribution is the integration and experimental validation of these mechanisms for service continuity under temporary communication or server failures.

The main contributions of this study are as follows:

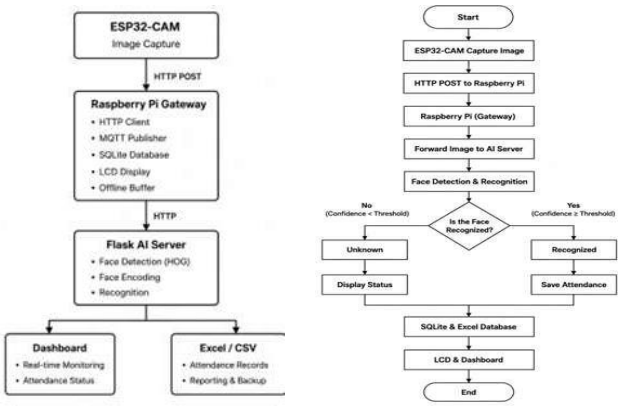
1. An integrated reliability-oriented Edge AI attendance architecture is developed by combining an ESP32-CAM for facial image acquisition, a Raspberry Pi 3 Model B+ as an edge gateway, MQTT communication, and a Flask-based Face Recognition server.
2. An offline buffering and automatic synchronization mechanism is implemented at the edge gateway to temporarily retain captured attendance data during temporary server or communication interruptions and automatically forward the buffered data after connectivity is restored.
3. The proposed architecture is experimentally evaluated not only in terms of Face Recognition accuracy but also in terms of HTTP communication success, MQTT communication delay, server response time, database consistency, dashboard synchronization, and offline buffer recovery.
4. The study demonstrates the feasibility of using a low-cost edge gateway to support service continuity in an IoT-based attendance system without requiring computationally intensive Face Recognition processing to be executed directly on the ESP32-CAM.

II. METHODS

This study develops an Internet of Things (IoT)-based employee attendance prototype using a distributed Edge AI architecture consisting of an ESP32-CAM, a Raspberry Pi 3 Model B+, and a Flask-based Face Recognition server. The methodology separates image acquisition, edge communication and buffering, and computationally intensive recognition. The ESP32-CAM acquires facial images, the

Copyright © 2026 Universitas Sebelas Maret

Raspberry Pi provides gateway and recovery functions, and the server performs Face Recognition and attendance processing. The research methodology and operational workflow are illustrated in Fig. 1, and The hardware specifications and operational configurations deployed in this prototype are detailed in Table I.



(a) System Architecture (b) Workflow

Fig. 1. (a) System Architecture (b) Workflow

A. System Architecture

The ESP32-CAM periodically captures facial images and transmits JPEG frames to the Raspberry Pi using HTTP POST over a Wi-Fi network. The Raspberry Pi validates incoming images, forwards them to the Flask-based Face Recognition server, manages MQTT communication, local SQLite services, LCD notification, and offline buffering. Recognition results are stored in SQLite and Excel databases, published through MQTT, and displayed on a web-based dashboard. When the server becomes unavailable, captured images are temporarily stored in the local buffer and automatically synchronized after communication recovery.

B. Facial Dataset Collection and Preparation

The facial reference dataset was collected from 11 registered employees who participated in the prototype evaluation. Approximately 15–20 facial images were available for each employee, resulting in 212 facial samples. The images were obtained from the employees' official photo-pass records and processed by the Face Recognition server. The samples were used to generate the reference facial representations for the registered identities. Because the dataset contains a limited number of identities and was collected under controlled conditions, the resulting performance is interpreted as a prototype-level closed-set evaluation rather than a population-level biometric benchmark.

Each facial sample was processed for face detection and representation extraction using the Dlib-based Face Recognition framework. A 128-dimensional facial embedding was generated for each detected face and associated with the corresponding employee identity. The resulting embeddings formed the reference database used during recognition. No retraining of the underlying Dlib model was performed in this study.

This dataset represents the initial registration conditions of the proposed attendance system. Because the dataset was collected from a small number of employees and under controlled conditions, the resulting recognition performance should be interpreted as a prototype-level evaluation, rather than a comprehensive assessment of biometric performance in diverse work environments.

C. Face Enrollment and Encoding Procedure

The proposed system does not retrain the underlying Dlib face recognition model. Instead, the employee enrollment process consists of extracting facial embeddings from the collected images and storing the resulting representations as reference data. The face detection process uses the Histogram of Oriented Gradients (HOG) method, while the facial representation is generated using a 128-dimensional embedding model provided by the Dlib-based Face Recognition framework.

During enrollment, each facial image is processed to identify the facial region and generate its corresponding embedding. The embeddings are then associated with the employee identity and stored in the facial encoding database. During the recognition stage, a newly acquired facial image is processed using the same procedure, and its embedding is compared with the stored reference embeddings using Euclidean distance.

A recognition result is accepted when the calculated similarity satisfies the predefined tolerance value of 0.5. This threshold was maintained consistently throughout the experiments to ensure that the recognition results were obtained under the same decision criterion.

D. Face Recognition Testing Procedure

The face recognition evaluation was conducted after the facial enrollment process had been completed. Each registered employee was presented to the ESP32-CAM, and facial images were acquired and transmitted to the Raspberry Pi 3 Model B+ edge gateway. The gateway forwarded the captured images to the Flask-based Face Recognition server for face detection and identity verification.

For each recognition attempt, the system compared the facial embedding extracted from the captured image with the reference embeddings stored during the enrollment process. The recognition result was considered correct when the system assigned the image to the actual registered employee identity. A recognition failure was recorded when the system failed to identify the registered employee or assigned the image to an incorrect identity.

The recognition threshold was fixed at a tolerance value of 0.5 for all experiments. The recognition performance was calculated from the number of correctly recognized samples relative to the total number of evaluated samples:

$$Recognition\ Rate = \frac{Correct\ Recognition}{Total\ Test\ Samples} \text{ x } 100\% \qquad (1)$$

The initial recognition evaluation was performed under controlled indoor workplace conditions. The ESP32-CAM was used for image acquisition, the Raspberry Pi 3 Model B+ operated as the edge gateway, and the Flask-based AI server executed Face Recognition. During the initial test, the camera position and network configuration were maintained, and participants were positioned within the normal operating range of the prototype. These conditions were intentionally controlled to establish baseline system behavior before robustness testing. The baseline result therefore represents performance under the tested conditions and should not be interpreted as comprehensive biometric performance across workplace environments.

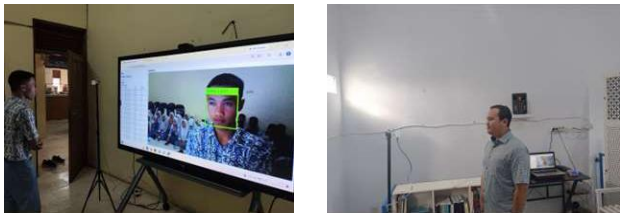
Vol. 08 No. 2, October-2026, Pages 102-110 DOI: https://dx.doi.org/10.20961/jeeict.8.2.125587
Copyright © 2026 Universitas Sebelas Maret

TABLE I. HARDWARE AND EXPERIMENTAL SETUP

Component	Specification / Configuration	Function
ESP32-CAM	AI Thinker ESP32-CAM, OV2640	Image acquisition and transmission
Edge Gateway	Raspberry Pi 3 Model B+ with 1 GB RAM operated using Debian GNU/Linux 13 (Trixie), Debian version 13.5, with a 64-bit aarch64 architecture and Linux kernel 6.18.34+rpt-rpi-v8.	Local buffering, MQTT communication, and synchronization
AI Server	Laptop, Intel® Core™ i5-4200U CPU @ 1.60 GHz, 12 GB RAM, Windows 11 Pro 24H2	Face recognition, database, and Flask application
Software Environment	Python 3.13.5 (virtual environment), Flask 3.1.3, OpenCV 4.13.0.92, NumPy 2.4.6, Pandas 3.0.3, scikit-learn 1.9.0, paho-mqtt 2.1.0	Face recognition application
Camera Setup	Height: 1.50 m and orientation: approximately horizontal.	Facial image acquisition
Robustness Setup	Distance: 0.5 and 1.0m, facial angles: 0° and 15°, occlusion: no occlusion, glasses and face mask	Robustness evaluation
Network	Common Wi-Fi network	Communication between system components
Communication	HTTP and MQTT	Image transmission and attendance synchronization
MQTT	Broker: 10.151.198.36, Port: 1883, Topic: attendance/result	Attendance-result communication
Power Supply	ESP32-CAM: ESP32-CAM-MB via USB, 5 V / 2 A adapter; Raspberry Pi: Micro USB, 5 V / 5 A adapter; AI server: standard AC power supply	Power supply for system components

E. Experimental Environment

The initial evaluation was conducted in an indoor environment representing the workplace. The ESP32-CAM was used as an image acquisition device, while the Raspberry Pi 3 Model B+ operated as an edge gateway for image forwarding and local communication management. The Face Recognition process was executed on a Flask-based AI server as depicted in Fig. 2.



(a) baseline attendance test with live dashboard (b) facial-orientation test configuration

Fig. 2. Experimental setup

During the initial evaluation, the participants were placed in front of the camera within the normal operating range of the prototype. The experiments were mainly conducted under controlled indoor lighting conditions, and the camera position was maintained during the evaluation. The network connection between the ESP32-CAM, the Raspberry Pi gateway, and the AI server was also maintained during the normal recognition tests.

These controlled conditions were selected to evaluate the basic recognition function of the proposed system before conducting more challenging environmental testing. Therefore, the reported recognition performance represents the behavior of the prototype under controlled conditions and should not be interpreted as a comprehensive measurement of biometric performance in all workplace conditions.

F. Robustness Testing

To address the reviewer's request for broader validation, a robustness-testing protocol was defined using independent test images that were not part of the enrollment reference set.

The protocol examines illumination period, facial orientation, subject-to-camera distance, facial occlusion, and unregistered users. The same recognition pipeline and decision threshold are maintained across conditions. For each trial, actual identity, predicted identity, confidence score, face distance, face count, and response time are recorded, as summarized in Table II. Quantitative results should be reported only from completed trials; no robustness or biometric error values are inferred from the baseline closed-set dataset.

1) Lighting Conditions

The system was evaluated under two lighting periods: daytime and nighttime. Daytime testing represented typical working-hour conditions with available ambient daylight and indoor illumination. Nighttime testing represented conditions after daylight was no longer available and the system relied primarily on artificial indoor lighting. For both periods, newly captured images not included in the enrollment reference set were used. The objective was to determine whether changes in ambient illumination affected face detection, confidence, distance, and response time.

2) Face Angle

The influence of facial orientation on recognition performance was evaluated by varying the horizontal face angle relative to the camera. Four face orientations were tested: 0° and 45°. The 0° condition represented a frontal face position and was used as the reference condition.

The subject was positioned at a fixed distance from the ESP32-CAM while gradually changing the facial orientation according to the predefined angles. This experiment was intended to determine the extent to which changes in facial pose affect face detection, embedding generation, recognition confidence, and recognition accuracy.

3) Subject-to-Camera Distance

The effect of the distance between the subject and the ESP32-CAM was evaluated at four distances: 0.5 m and 1.0 m. The camera position and orientation were maintained throughout the experiment, while the subject position was adjusted according to the specified distance.

For each distance, the system recorded the predicted identity, confidence score, face distance, face detection result, and response time. The results were then compared to determine the operating distance at which the proposed system provides reliable face recognition performance.

Copyright © 2026 Universitas Sebelas Maret

4) Occlusion Conditions

Robustness against partial obstruction of facial features was evaluated using four conditions: no occlusion, eyeglasses and face mask. The no-occlusion condition served as the baseline. The eyeglasses condition was used to evaluate the effect of obstruction around the eye region, while the face-mask condition evaluated recognition when the lower part of the face was covered.

These experiments were performed using registered subjects, and the recognition results were compared with the baseline condition. The evaluation focused on recognition rate, confidence score, face distance, and response time.

5) Unknown Users

To evaluate the system's ability to reject individuals who are not registered in the database, additional tests were conducted using unknown users who were not included in the enrollment dataset. These subjects were processed using the same recognition pipeline and experimental settings as the registered users.

An unknown-user attempt was considered a correct rejection when the system classified the subject as *Unknown*. Conversely, when an unknown user was incorrectly assigned to one of the registered identities, the result was categorized as a false acceptance.

TABLE II. SUMMARY OF ROBUSTNESS TESTING

Test Category	Experimental Conditions	Evaluation Objective
Lighting	Daytime Conditions, Nighttime Conditions.	Evaluate robustness to illumination changes
Face Angle	0°, 45°	Evaluate the effect of facial orientation
Distance	0.5 m, 1.0 m	Determine the effect of subject-to-camera distance
Occlusion	No occlusion, Glasses, Face mask	Evaluate robustness to partial obstruction
User Type	Registered users, Unknown users	Evaluate genuine recognition and unknown-user rejection
Test image separation	Robustness images newly captured and excluded from enrollment reference samples	

G. System Workflow

The operational workflow consists of six sequential stages:

1. Facial image acquisition using ESP32-CAM.
2. HTTP image transmission to the Raspberry Pi gateway.
3. Face detection and recognition on the Flask server.
4. Attendance verification using a 10-second cooldown mechanism.
5. Storage of attendance records in SQLite and Excel databases.
6. Dashboard synchronization through MQTT or local buffering if the server is unavailable.

Whenever communication fails, the Raspberry Pi automatically stores captured images locally and periodically retransmits them after the server becomes available while preserving chronological order.

H. Face Recognition Process

Face Recognition is implemented using the Dlib Histogram of Oriented Gradients (HOG) detector and 128-dimensional facial embeddings. Identity matching uses Euclidean distance through the Face Recognition library functions compare_faces() and face_distance(), with a fixed tolerance value of 0.5. The 10-second cooldown mechanism is used to reduce duplicate attendance records during continuous image acquisition. The recognition threshold and cooldown are system parameters rather than newly trained model parameters.The baseline evaluation dataset contains 212 facial samples from 11 registered employees.

I. System Evaluation

The proposed prototype was evaluated through functional and performance testing covering Face Recognition accuracy, communication reliability, database consistency, dashboard synchronization, and offline buffer recovery. During offline buffer evaluation, the Face Recognition server was intentionally disconnected for five minutes to verify automatic buffering and sequential synchronization after reconnection. The experimental configuration is summarized in Table III.

TABLE III. SYSTEM EVALUATION CONFIGURATION

Parameter	Configuration
Registered employees	11
Face encodings	212
Images per employee	15–20
Image resolution	640 × 480 JPEG
Face detector	Dlib HOG
Face embedding	128-D Dlib ResNet
Recognition threshold	50%
Duplicate prevention	10 s cooldown
Network	Wi-Fi 2.4 GHz
MQTT	Eclipse Mosquitto (QoS 1)
Buffer mechanism	Raspberry Pi Offline Buffer
Evaluation	Black-box, Face Recognition, HTTP, MQTT, Dashboard, SQLite, Offline Buffer

J. Security, Privacy, and Ethical Considerations

The proposed system processes facial images and attendance records; therefore, security, privacy, and ethical considerations are important. The current prototype focused on functional operation, face-recognition performance, and communication reliability, while TLS encryption, biometric-template protection, role-based access control, secure credential management, data-retention policies, and presentation-attack detection were not evaluated. The prototype used HTTP for image transmission and MQTT for attendance-result synchronization, with the MQTT broker configured at 10.151.198.36, port 1883, using the attendance/result topic. Thus, this configuration represents an experimental prototype rather than a production-level security implementation. Future deployment should incorporate HTTPS/MQTT-TLS, client authentication, access control, protected biometric storage, restricted dashboard access, data-retention policies, and liveness detection. Facial data should be accessed only by authorized personnel and used for the intended research and attendance purposes. Facial images from 11 participants were collected for system development and evaluation; the manuscript should state the actual informed-consent or ethical-approval procedure applied during data collection.

K. Additional Reliability Validation Requirements

Although the prototype demonstrated service continuity during the reported communication-failure tests, further

Copyright © 2026 Universitas Sebelas Maret

evaluation is required under less controlled conditions. Unstable networks, server outages exceeding five minutes, concurrent requests, and higher attendance loads were not experimentally evaluated and are therefore considered future validation scenarios.

Future tests should assess packet loss, outage duration, concurrent requests, and workload, using buffer occupancy, recovery time, duplicate records, failed transactions, synchronization success, and data-loss rate as evaluation metrics. The current results demonstrate the feasibility of the proposed buffering and synchronization mechanism under the tested conditions, while broader reliability and scalability remain to be validated.

III. RESULTS

This section presents the implementation and evaluation of the proposed Smart Attendance System integrating the ESP32-CAM, Raspberry Pi 3 Model B+, and the Face Recognition server. The evaluation covers functional correctness, Face Recognition performance, communication reliability, response time, dashboard synchronization, and offline buffer recovery. All experiments were conducted under the same hardware platform, communication settings, recognition threshold (50%), and duplicate prevention interval (10s) to ensure consistent and reproducible performance throughout the evaluation.

This section presents the experimental evaluation of the proposed Smart Attendance prototype. To provide a structured assessment, the system was evaluated through functional validation, Face Recognition performance, communication reliability, response time, dashboard synchronization, and offline buffer recovery. Table IV summarizes the overall evaluation matrix, including the number of experimental trials, success and failure cases, and the principal performance indicators evaluated throughout this study. The detailed results corresponding to each evaluation category are presented in the following subsections

TABLE IV. SUMMARY OF SYSTEM EVALUATION RESULTS

Evaluation Parameter	Trials	Success	Failure	Evaluation Result
Black-box Testing	8	8	0	Passed (100%)
Face Recognition (closed-set baseline)	11	11	0	Recognition rate = 100% (controlled closed-set test)
Response Time	30	30	0	Average response time = 1.069 s
HTTP Communication	30	30	0	Success rate = 100%
MQTT Communication	34	34	0	Average delay = 2.47 ms
SQLite Database	30	30	0	Data storage consistent in 30/30 trials
Dashboard Synchronization	30	30	0	Real-time synchronization successful in 30/30 trials
Offline Buffer Recovery	30	30	0	Successful recovery in 30/30 trials (100%)
Robustness Testing Protocol	Not completed quantitatively	-	-	Protocol defined; quantitative results are not reported because the independent robustness trials were not completed

A. Black-box Testing

Functional correctness was first evaluated to verify that every subsystem operated according to its design specification before quantitative performance measurements were conducted.

TABLE V. BLACK-BOX TESTING RESULTS

No.	Function Tested	Expected Result	Test Result	Status
1	ESP32-CAM image acquisition	Facial image successfully captured	Successful	Valid
2	HTTP POST communication	Image successfully transmitted to Raspberry Pi	Successful	Valid
3	Raspberry Pi gateway	Image successfully forwarded to AI server	Successful	Valid
4	Face Recognition	Registered identity correctly recognized	Successful	Valid
5	SQLite database	Attendance data successfully stored	Successful	Valid
6	Excel attendance report	Attendance record successfully exported	Successful	Valid
7	LCD notification	User identity displayed correctly	Successful	Valid
8	Web dashboard	Attendance data synchronized in real time	Successful	Valid

Table V summarizes the functional evaluation of the proposed Smart Attendance System. All functional modules operated according to their design specifications without observed failures, confirming successful integration among the ESP32-CAM, Raspberry Pi gateway, Face Recognition server, SQLite database, LCD interface, and web dashboard.

B. Face Recognition Performance

Following functional validation, the Face Recognition subsystem was evaluated using the experimental configuration summarized in Table VI.

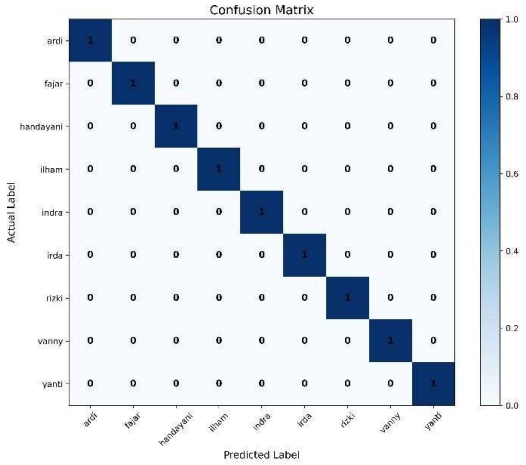


Fig. 3. Confusion Matrix of Face Recognition Performance

Copyright © 2026 Universitas Sebelas Maret

Table VI summarizes the Face Recognition configuration used during the experiments. Recognition performance was assessed using the confusion matrix shown in Fig. 3.

TABLE VI. FACE RECOGNITION TEST CONFIGURATION

Parameter	Result
Registered identities	11 employees
Face encodings	212
Images per identity	15–20 images
Face detection	Dlib Histogram of Oriented Gradients (HOG)
Feature extraction	128-dimensional Dlib ResNet Face Embedding
Similarity measurement	compare_faces() and face_distance()
Recognition threshold	50% (selected through preliminary experiments)
Duplicate prevention	10-second cooldown
Attendance storage	SQLite, CSV, Microsoft Excel
Dashboard synchronization	MQTT Publish–Subscribe
Offline mechanism	Raspberry Pi Offline Buffer
Evaluation metric	Accuracy Score
Test dataset	*test_dataset*
Recognition output	Recognized / Unknown

Fig. 3 presents the confusion matrix obtained during testing. All registered identities were correctly classified without off-diagonal misclassification, resulting in 100% recognition accuracy (212/212 correct classifications) under the experimental conditions. The selected recognition threshold and the 10-second cooldown mechanism also prevented duplicate attendance records during continuous image acquisition.

C. *Response Time Evaluation*

Communication latency was subsequently evaluated by measuring the server response time during repeated attendance transactions.

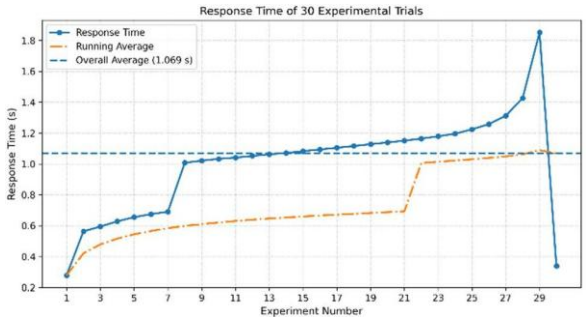


Fig. 4. Response Time of 30 Experimental Trials

Fig. 4 illustrates the response time measured during 30 experimental trials. The proposed system achieved an average server response time of 1.069 s, with only minor fluctuations caused by image processing and communication latency. Although one trial experienced a higher response time, all attendance transactions were completed successfully without communication failure or data loss, indicating stable system performance throughout the evaluation.

D. *HTTP Communication Performance*

Reliable image transmission between the ESP32-CAM, Raspberry Pi gateway, and Face Recognition server was evaluated through HTTP communication testing.

TABLE VII. HTTP COMMUNICATION PERFORMANCE

Parameter	Result
Number of transmission trials	30
Successful image transmissions	30
Failed transmissions	0
Success rate	100%
Communication protocol	HTTP POST
Image format	JPEG (640 × 480)
Network	Wi-Fi 2.4 GHz

Table VII summarizes the HTTP communication performance. All captured images were successfully transmitted from the ESP32-CAM through the Raspberry Pi gateway to the Face Recognition server, achieving a 100% communication success rate without packet loss during the experimental trials.

E. *MQTT Communication Performance*

MQTT performance was evaluated to verify reliable dashboard synchronization after attendance processing.

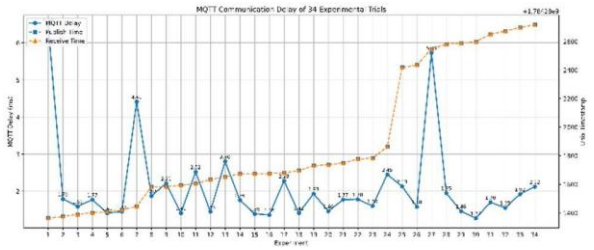


Fig. 5. MQTT Communication Delay of 34 Experimental Trials.

Fig. 5 presents the MQTT communication delay measured over 34 experimental trials. The average communication delay was 2.47 ms, with only a few temporary peaks caused by network conditions and operating system scheduling. No message loss or communication failure was observed, demonstrating reliable MQTT-based synchronization between the Raspberry Pi gateway and the dashboard.

F. *Offline Buffer Performance*

The robustness of the proposed Edge AI architecture was further evaluated by intentionally interrupting communication with the Face Recognition server.

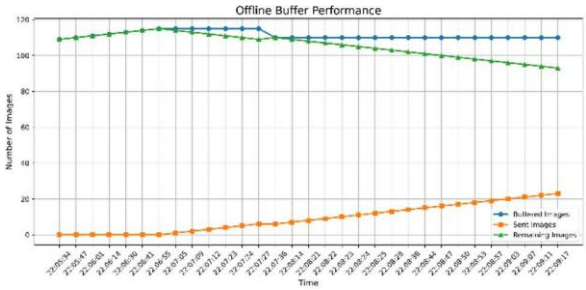


Fig. 6. Offline Buffer Performance during Server Downtime and Automatic Synchronization

Fig. 6 illustrates the offline buffering mechanism during temporary server downtime. While the Face Recognition server was unavailable, the Raspberry Pi successfully stored captured images in the local buffer. After the server resumed operation, all buffered data were automatically retransmitted in chronological order without observed data loss or duplicate attendance records. These results confirm the effectiveness of

Copyright © 2026 Universitas Sebelas Maret

the proposed store-and-forward mechanism for maintaining service continuity during temporary communication failures.

G. Dashboard Synchronization

Finally, the implementation output was verified through the web-based dashboard to demonstrate successful synchronization of attendance information after all technical evaluations had been completed.

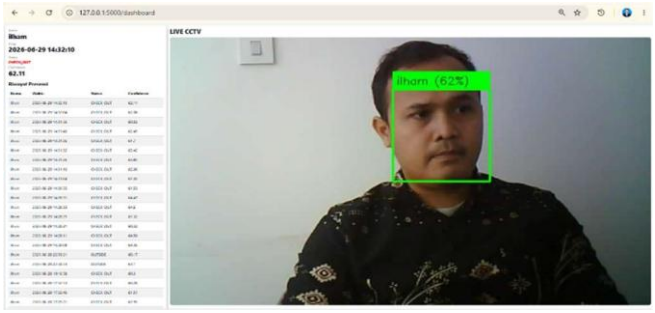


Fig. 7. Bootstrap-Based Web Dashboard

Fig. 7 presents the Bootstrap-based web dashboard developed for real-time monitoring of the proposed Smart Attendance System. The interface integrates live CCTV streaming with Face Recognition outputs, displaying the recognized identity, confidence score, attendance status, timestamp, and attendance history. Following successful recognition, attendance records were automatically updated on the dashboard, providing continuous visualization of system operation and supporting real-time monitoring through a unified web interface.

H. Daytime and Nighttime Operational Observation

In addition to the controlled technical evaluations, operational observations were conducted to examine system operation during daytime and nighttime attendance periods. The daytime period was defined as 06:00–16:00 for CHECK-IN, while the nighttime period was defined as 16:00–24:00 for CHECK-OUT. The observations included the number of attendance records, recognized identity, confidence range, and the occurrence of Unknown classifications as outlined below in Table VIII.

TABLE VIII. DAYTIME AND NIGHTTIME OPERATIONAL OBSERVATION RESULTS

Evaluation Parameter	Daytime (06:00–16:00)	Nighttime (16:00–24:00)	Remarks
Observation period	06:00–16:00	16:00–24:00	Time-based operational evaluation
Attendance status	CHECK-IN	CHECK-OUT	Defined according to the attendance schedule
Number of records	510	161	Based on the supplied attendance records
Recognized identity	Ilham	Ilham	All available records were identified as the registered user *Ilham*
Unknown user	0	0	No Unknown classification was recorded in the supplied dataset
Recognition result	Ilham recognized	Ilham recognized	Successful operational recognition was observed in the supplied records
Recognition confidence range	50.01–70.74%	50.01–62.95%	Observed confidence range
Lowest confidence	50.01%	50.01%	Lowest observed confidence
Highest confidence	70.74%	62.95%	Highest observed confidence
Data availability	Available	Available	Both daytime CHECK-IN and nighttime CHECK-OUT records are available
Evaluation status	Operationally observed	Operationally observed	Results represent operational observations
Evaluation interpretation	Registered-user recognition observed	Registered-user recognition observed	Not interpreted as biometric accuracy, FAR, or FRR

I. Overall Evaluation Summary

The experimental results collectively indicate that the proposed Smart Attendance prototype functioned as intended under the defined experimental conditions. All functional modules operated successfully, while the Face Recognition subsystem correctly classified all evaluation samples. Communication tests demonstrated successful HTTP image transmission and low MQTT communication delay, supporting stable synchronization between the ESP32-CAM, Raspberry Pi gateway, AI server, and web dashboard. In addition, the offline buffering mechanism successfully preserved attendance data during temporary server interruptions and automatically synchronized buffered records after communication was restored without observed data loss or duplicate attendance records. These findings demonstrate the feasibility of integrating Face Recognition, Edge AI, and IoT communication into a unified employee attendance prototype for controlled experimental environments.

IV. CONCLUSIONS

This study presents the design and implementation of an Internet of Things (IoT)-based employee attendance prototype integrating an ESP32-CAM for facial image acquisition, a Raspberry Pi 3 Model B+ as an intelligent edge gateway, and a Python-based Face Recognition server. The proposed architecture combines HTTP image transmission, MQTT publish–subscribe communication, SQLite and CSV/Excel attendance storage, offline buffering, automatic synchronization, LCD notification, and web-based dashboard monitoring within a unified Edge AI framework. Under the experimental conditions, all functional modules successfully passed black-box testing, while the Face Recognition subsystem correctly classified all 212 evaluation samples from 11 registered employees. The system also achieved a 100% HTTP image transmission success rate, an average server response time of 1.069 s, an average MQTT communication delay of 2.47 ms, successful offline buffer recovery, and effective duplicate-attendance prevention through the implemented 10-second cooldown mechanism. These findings demonstrate the feasibility of integrating Face Recognition, edge gateway processing, and IoT communication into a functional attendance prototype capable of maintaining service continuity during temporary server interruptions. However, the results should be interpreted as evidence obtained from a controlled prototype

Copyright © 2026 Universitas Sebelas Maret

evaluation rather than confirmation of long-term operational reliability or readiness for practical workplace deployment.

The present study has several limitations that should be addressed before the proposed architecture can be adopted in real organizational environments. The evaluation involved a limited identity dataset, controlled laboratory conditions, a single-camera configuration, and did not include comprehensive biometric performance analysis using False Acceptance Rate (FAR), False Rejection Rate (FRR), Equal Error Rate (EER), Receiver Operating Characteristic (ROC), or large-scale unknown-user validation. In addition, security and privacy aspects, including encrypted communication, role-based access control, biometric data protection, and presentation attack (spoofing) detection, were not evaluated. Therefore, future research should investigate larger, more diverse biometric datasets; real-world workplace deployment; multi-user and multi-device scalability; network robustness; comprehensive biometric accuracy evaluation; privacy and cybersecurity mechanisms; cloud or Enterprise Resource Planning (ERP) integration; and comparative studies with conventional fingerprint- and card-based attendance systems. These improvements will provide a more comprehensive validation of the proposed architecture and better assess its applicability for practical intelligent attendance management systems.

ACKNOWLEDGMENT

The research reported in this article was supported by Mayasari Bakti University through the Research Grant for the 2025/2026 Academic Year under Contract No. 025/Contract-Research/LPPM/UMB/VI/2026. The authors gratefully acknowledge the valuable support and contributions of all members of the research team throughout the completion of this study.

REFERENCES

[1] M. A. Azkiya Aziz, S. Ismail, and N. Allias, "Deep Learning in Face Recognition for Attendance System: An Exploratory Study," J. Comput. Res. Innov., vol. 7, no. 2, pp. 74–81, 2022, doi: 10.24191/jcrinn.v7i2.288.

[2] J. Patel, S. Gandhi, V. Katheriya, P. Pataliya, and A. Majumdar, "Enhancing Classroom Attendance Systems with Face Recognition through CCTV using Deep Learning," Procedia Comput. Sci., vol. 258, pp. 3031–3041, 2025, doi: 10.1016/j.procs.2025.04.561.

[3] G. M. G. Makigod, J. J. B. Sebastian, E. R. Arboleda, and L. A. T. Ongquit, "Design and Development of IoT Based Automated Attendance System Using Facial Recognition," J. Eng. Technol. Ind. Appl., vol. 10, no. 48, pp. 136–144, Jul. 2024, doi: 10.5935/jetia.v10i48.976.

[4] Ubong David Essien and Godwin Okon Ansa, "A deep learning-based face recognition attendance system," Glob. J. Eng. Technol. Adv., vol. 17, no. 1, pp. 009–022, Oct. 2023, doi: 10.30574/gjeta.2023.17.1.0165.

[5] R. Singh and S. S. Gill, "Edge AI: A survey," Internet Things Cyber-Physical Syst., vol. 3, no. March, pp. 71–92, Jan. 2023, doi: 10.1016/j.iotcps.2023.02.004.

[6] L. Kong and S. Jiao, "Edge-computing-driven Internet of Things : A Survey Edge-computing-driven Internet of Things : A Survey," vol. 55, no. 8, 2026. doi : 10.1145/355530

[7] M. Whaiduzzaman et al., "A Review of Emerging Technologies for IoT-Based Smart Cities," Sensors, vol. 22, no. 23, pp. 1–28, Dec. 2022, doi: 10.3390/s22239271.

[8] M. Wang and W. Deng, "Deep face recognition: A survey," Neurocomputing, vol. 429, pp. 215–244, Mar. 2021, doi: 10.1016/j.neucom.2020.10.081.

[9] Y. Jing, X. Lu, and S. Gao, "3D face recognition: A comprehensive survey in 2022," Comput. Vis. Media, vol. 9, no. 4, pp. 657–685, Dec. 2023, doi: 10.1007/s41095-022-0317-1.

[10] H. L. Gururaj, B. C. Soundarya, S. Priya, J. Shreyas, and F. Flammini, "A Comprehensive Review of Face Recognition Techniques, Trends, and Challenges," IEEE Access, vol. 12, no. June, pp. 107903–107926, 2024, doi: 10.1109/ACCESS.2024.3424933.

[11] A. Nemavhola, C. Chibaya, and S. Viriri, "A Systematic Review of CNN Architectures, Databases, Performance Metrics, and Applications in Face Recognition," Inf., vol. 16, no. 2, pp. 1–41, Feb. 2025, doi: 10.3390/info16020107.

[12] F. Barreto, J. Sarvaiya, and S. Patnaik, "Learning Representations for Face Recognition: A Review from Holistic to Deep Learning," Adv. Technol. Innov., vol. 7, no. 4, pp. 279–294, Oct. 2022, doi: 10.46604/aiti.2022.8308.

[13] D. Wanyonyi and T. Celik, "Open-Source Face Recognition Frameworks: A Review of the Landscape," IEEE Access, vol. 10, pp. 50601–50623, 2022, doi: 10.1109/ACCESS.2022.3170037.

[14] Y. Feng, S. Yu, H. Peng, Y.-R. R. Li, and J. Zhang, "Detect Faces Efficiently: A Survey and Evaluations," IEEE Trans. Biometrics, Behav. Identity Sci., vol. 4, no. 1, pp. 1–18, Dec. 2022, doi: 10.1109/TBIOM.2021.3120412.

[15] [15] J. Nannung, A. Sa'ban Miru, Muliadi, and Gunawan, "Development of an IoT-Based Home Security System Prototype Using Multisensors and ESP32-CAM," J. Embed. Syst. Secur. Intell. Syst., vol. 6, no. 2, pp. 168–177, Jun. 2025, doi: 10.59562/jessi.v6i2.8456.

[16] L. Parsons, G. Deng, and R. Ross, "Detecting image edges in IoT nodes without FPU," Multimed. Tools Appl., vol. 83, no. 10, pp. 31161–31175, 2024, doi: 10.1007/s11042-023-16672-4.

[17] C. Contoli, L. Calisti, G. Di Fabrizio, N. Kania, A. Bogliolo, and E. Lattanzi, "A power-aware vision-based virtual sensor for real-time edge computing," J. Real-Time Image Process., vol. 21, no. 4, pp. 1–12, Aug. 2024, doi: 10.1007/s11554-024-01482-0.

[18] A. Mileva, A. Velinov, L. Hartmann, S. Wendzel, and W. Mazurczyk, "Comprehensive analysis of MQTT 5.0 susceptibility to network covert channels," Comput. Secur., vol. 104, p. 102207, May 2021, doi: 10.1016/j.cose.2021.102207.

[19] J. Dizdarevic, M. Michalke, and A. Jukan, "Engineering and Experimentally Benchmarking Open Source MQTT Broker Implementations," May 2023, [Online]. Available: https://arxiv.org/abs/2305.13893.

Copyright © 2026 Universitas Sebelas Maret