

Design of an Edge Computing–Based Security System with Multimodal Face and Voice Biometric Authentication

Anggay Luri Pramana¹, Mochammad Fatih Ulumuddin¹, Daniel Achmad Farizki¹

¹Department of Informatics, Nahdlatul Ulama University of Sidoarjo, Indonesia.

Abstract

The increasing number of intrusion and unauthorized room access incidents necessitates the development of a security system that is more intelligent, responsive, and resistant to forgery. Conventional security systems or card-based identification methods are highly vulnerable to theft, duplication, and misuse, highlighting the need for biometric authentication that can operate independently without reliance on centralized servers. In this study, an edge computing approach is implemented by utilizing a Raspberry Pi as a local processing unit capable of performing real-time data processing without requiring an internet connection. A RaspiCam camera is employed to capture facial images, which are then detected using the YOLO (You Only Look Once) algorithm to achieve fast and efficient face detection. The detected faces are subsequently recognized using a Convolutional Neural Network (CNN). To further enhance security, a voice-based authentication stage is applied using the Mel-Frequency Cepstral Coefficients (MFCC) method to verify user identity. Experimental results demonstrate that the face recognition module achieves an accuracy of up to 96.5% under normal lighting conditions, while the voice recognition module reaches an accuracy of 94.7% in a quiet environment with a signal-to-noise ratio of 30 dB. Both authentication processes are executed sequentially with an average processing time of less than 500 ms, enabling real-time operation on an edge device. This approach provides a robust two-factor biometric authentication system that can be deployed without dependence on cloud-based services.

Keywords

CNN, Edge Computing, Face Recognition, Voice Recognition, YOLO.

*Correspondence
Anggay Luri Pramana
luri409.tif@unusida.ac.id

Introduction

Room security has become a critical concern due to the increasing risk of intrusion and unauthorized access by illegitimate parties. Conventional security mechanisms such as physical keys, identification cards, or PIN codes exhibit inherent weaknesses, as they can be shared, duplicated, or compromised, making them insufficient for ensuring a high level of security. To address these limitations, biometric authentication has been widely adopted because biometric traits are unique and inherently linked to each individual, making them difficult to forge. However, many existing biometric systems still rely on cloud-based services for data processing, which may introduce response latency and raise concerns regarding user privacy and data security.

Recent advancements in edge computing provide an alternative solution by shifting computational processes to local devices, enabling real-time authentication without requiring an internet connection. In this study, a Raspberry Pi is utilized as the primary processing unit to independently perform face detection, face recognition, and voice verification. Face detection is carried out using the YOLO (You Only Look Once) algorithm, which offers superior speed and accuracy compared to classical detection methods such as Haar Cascade. Once a face is detected, identity verification is performed using a Convolutional Neural Network (CNN), followed by voice authentication employing the Mel-Frequency Cepstral Coefficients (MFCC) method to ensure multi-layer identity validation. This integrated approach results in a room security system that is more reliable, responsive, and resistant to spoofing, while simultaneously reducing the risk of unauthorized access and data leakage.

Methodology

Face recognition is one of the earliest yet continuously evolving research topics, particularly due to its importance in security and biometric applications. Early face recognition systems primarily relied on handcrafted features and traditional classifiers, which often exhibited limited performance and adaptability under diverse conditions. Masud et al. (2020) In contemporary systems, facial identity recognition is commonly implemented using Convolutional Neural Networks (CNNs). CNNs are capable of automatically and adaptively learning high-level feature representations, thereby significantly improving face recognition accuracy across varying lighting conditions and facial poses. Suryadi et al (2010).

To enhance overall system security, authentication mechanisms are frequently integrated with voice recognition as an additional verification layer. According to studies reported in NeuroImage: Neural mechanisms for voice recognition, human voice recognition is supported by cognitive mechanisms that can identify unique vocal characteristics, including vocal patterns, throat resonance, and acoustic waveform properties. These findings indicate that voice recognition is highly individual-specific and relatively stable, making it an effective complementary biometric modality for authentication purposes. Pulli et al (2012).

In digital speech processing, one of the most widely used feature extraction techniques is Mel-Frequency Cepstral Coefficients (MFCC). MFCC transforms speech signals from the time domain into a spectral representation that better reflects the characteristics of human auditory perception. Based on the study Improved MFCC-based feature for robust speaker identification, enhanced MFCC methods have demonstrated improved robustness against noise and variations in acoustic environments, thereby increasing the reliability of speaker identification and voice authentication in real-world applications. Z. Wu et al (2005).

By integrating YOLO-based face detection, CNN-based face recognition, and MFCC-based voice authentication, security systems can provide stronger, more accurate, and spoof-resistant identity

validation. The effectiveness of this approach is further enhanced through the adoption of edge computing, where all biometric processing is performed locally on the device, resulting in faster response times and improved protection of sensitive biometric data.

YOLO

YOLO (You Only Look Once) is a deep learning-based object detection algorithm first introduced by Redmon et al. in 2016. The primary advantage of YOLO lies in its high detection speed compared to earlier object detection approaches such as R-CNN, Fast R-CNN, and Faster R-CNN. While region-based detection methods typically operate through a two-stage process involving region proposal generation followed by classification, YOLO reformulates object detection as a single regression problem. This approach enables the direct prediction of object classes and bounding boxes from the input image in an end-to-end manner, resulting in faster and more efficient detection performance. Fauziyah et al. (2025).

Comparison YOLO Version

Table 1. Comparison YOLO Version

Method	Description
YOLOv5	Implemented using PyTorch and supports multiple model scales (small, medium, large, and extra-large).
YOLOv6	Optimized for industrial applications with improved inference speed on servers and edge devices.
YOLOv7	Provides more efficient multi-task detection compared to YOLOv5 and YOLOv6.
YOLOv8	Supports classification, detection, and segmentation; offers plug-and-play functionality via the Ultralytics API and is optimized for CPU, GPU, and edge deployment.

Table 1 presents a comparison of the advantages of each YOLO version, including framework support, model scalability, detection efficiency, and optimization for server and edge device deployment.

Comparison YOLO and Haarcascade

Table 2. Comparison YOLO and Haarcascade

Aspect	Haarcascade	YOLO
Method	Haar features with cascade classifier	Deep learning-based single-shot CNN detector
Speed	Fast on simple images	Very fast and stable for real-time applications
Accuracy	Moderate, sensitive to environmental conditions	High, robust to object appearance variations
Environmental Conditions	Performs poorly under varying lighting and pose	Performs well in complex environments

Table 2 describes the comparison between Haar Cascade and YOLO in terms of detection method, speed, accuracy, robustness to environmental variations, and computational requirements.

Raspberry Pi

Raspberry Pi is a compact single-board computer (SBC) developed by the Raspberry Pi Foundation in the United Kingdom since 2012. Initially designed as an affordable educational platform for teaching

programming and basic computing concepts, Raspberry Pi has evolved into a widely adopted platform for research, prototyping, and hobbyist applications. Its low cost, small form factor, and sufficient computational capabilities make it particularly suitable for embedded systems and edge computing-based implementations.

OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source software library designed for image processing and computer vision applications. Initially developed by Intel in 1999, OpenCV has evolved into one of the most widely used libraries in the fields of computer vision, machine learning, and robotics. The library provides extensive functionalities for real-time image and video processing, making it a standard tool for research and practical implementations in vision-based systems. Elnozahy et al (2025).

Normalization

Before further processing, the input facial images are normalized to improve consistency and robustness of the recognition results. Image normalization aims to reduce variations caused by lighting conditions, contrast differences, and sensor noise, thereby ensuring more stable feature extraction. One commonly applied technique is intensity normalization, where pixel values are scaled into a predefined range, typically 0 to 255. Intensity normalization can be expressed as:

$$I_{norm} = \frac{I - I_{min}}{I_{max} - I_{min}}$$

Where I represents the original pixel intensity, while I_{min} and I_{max} denote the minimum and maximum intensity values in the image, respectively. This normalization process helps standardize facial image inputs prior to feature extraction and classification, leading to improved recognition performance under varying illumination conditions.

Greyscale

Color images in RGB format are converted into grayscale prior to further processing. This conversion aims to reduce computational complexity while preserving essential intensity information required for feature extraction. The grayscale conversion is performed using the luminance method, which accounts for the different sensitivities of the human visual system to red, green, and blue components.

The grayscale intensity value is calculated as follows:

$$I_{gray} = 0.299R + 0.587G + 0.114B$$

where R , G , and B represent the red, green, and blue channel intensities, respectively. The weighting coefficients emphasize the green channel, which has the greatest influence on human brightness perception. This luminance-based grayscale representation produces a more perceptually accurate intensity image, making it suitable for subsequent normalization and feature extraction stages in face recognition systems.

Similarity Matching

During the recognition stage, facial identity is determined by measuring the similarity between the extracted facial feature vector of the input image and the stored facial templates in the database. Similarity measurement plays a crucial role in deciding whether the detected face corresponds to a registered user. One of the most commonly used similarity metrics is Euclidean Distance, which computes the geometric distance between two feature vectors. The Euclidean Distance is defined as:

$$D(x, y) = \sum_{i=1}^n (x_i - y_i)^2$$

where $x = (x_1, x_2, \dots, x_n)$ represents the feature vector extracted from the input face, and $y = (y_1, y_2, \dots, y_n)$ denotes the corresponding facial template stored in the database. A smaller distance value indicates a higher degree of similarity between the two facial representations, thereby increasing the likelihood that both vectors belong to the same individual.

Voice Recognition

Artificial intelligence-based voice recognition systems utilize neural network models to emulate the mechanisms of human auditory processing. These systems operate by processing audio signals through multiple layers of artificial neural networks to extract discriminative voice features and perform predictions, such as recognizing spoken words or commands. Pulli et al (2012). To implement voice recognition computationally, an effective feature extraction method is required to accurately capture speaker-specific characteristics. One of the most widely adopted techniques is Mel-Frequency Cepstral Coefficients (MFCC).

This method maps audio signals onto the Mel scale, which reflects human auditory perception of sound frequency, thereby producing feature representations that are more consistent with the characteristics of human hearing. According to the study Improved MFCC-based Feature for Robust Speaker Identification, MFCC demonstrates strong robustness against variations in intonation and environmental noise, making it suitable for speaker identification systems that require high reliability under real-world conditions. Z. Wu et al (2015).

The MFCC extraction process consists of several stages, including pre-emphasis, framing, windowing, Fast Fourier Transform (FFT), Mel filterbank processing, and Discrete Cosine Transform (DCT), resulting in a compact yet representative voice feature vector. When combined with deep learning techniques, MFCC-based features enable high identification accuracy and stable performance in practical voice recognition applications.

Feature Extraction and Representation Voice Signal

Audio signals captured by microphones on IoT devices are transformed into numerical representations using feature extraction techniques such as Mel-Frequency Cepstral Coefficients (MFCC), spectrograms, or log-Mel filterbanks. These representations are then used as input to neural network models, enabling the system to learn both spectral and temporal patterns inherent in human speech. Through this process, the neural network can effectively model distinctive voice characteristics for tasks such as recognition, classification, or authentication. Pulli et al (2012).

Figures 1–5 illustrate the motivation, transformation behavior, feature robustness, and performance impact of logarithmic and MFCC-based processing in speech recognition under noisy conditions. Figure 1 shows the logarithmic compression function, where small variations in high-energy regions are reduced while preserving discriminative details in low-energy regions, improving dynamic range handling. Figure 2 presents the log filter-bank energies of clean and noisy speech, demonstrating that logarithmic scaling significantly reduces the gap between clean and noisy signals, thereby enhancing robustness. In contrast, Figure 3 depicts raw filter-bank energies without logarithmic compression, where noise causes larger distortions, especially in low-energy frequency bands.

Figure 4 compares standard logarithmic functions with generalized log functions using different parameters, highlighting how parameter selection controls the degree of compression and noise suppression. Finally, Figure 5 reports recognition accuracy versus SNR, showing that MFCC-based methods—particularly the proposed SS_MF approach—consistently outperform the baseline under low SNR conditions, with performance converging as the signal becomes cleaner, confirming the effectiveness of logarithmic spectral processing for noise-robust speech recognition.

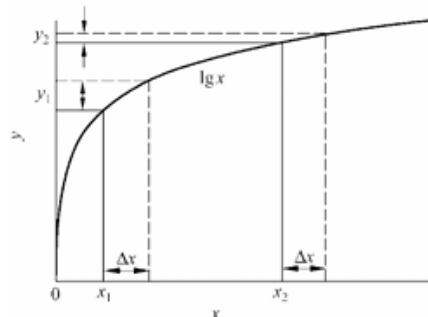


Figure 2. Mismatches log transformation.

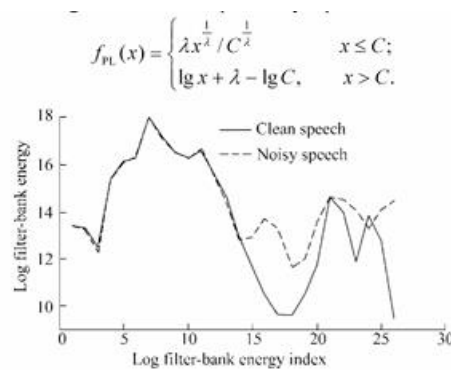


Figure 3. Filter-bank energy index.

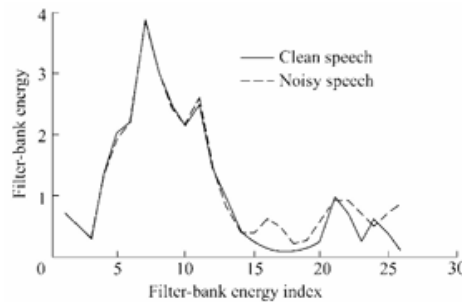


Figure 4. Filter-bank energies using the combined Transformation.

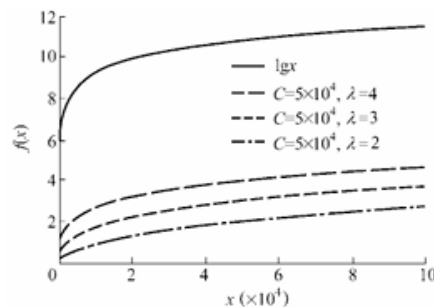


Figure 5. Transformation functions corresponding to different parameters C and λ .

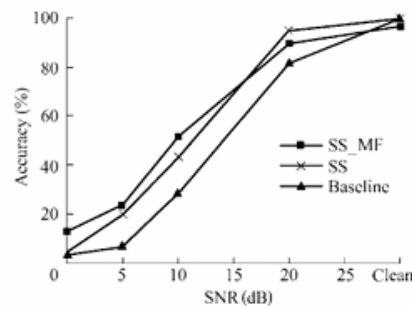


Figure 6. Performance Baseline, SS, and SS_MF Methods.

In speaker identification systems, three feature extraction methods are commonly employed, namely the Baseline method, Spectral Subtraction (SS) method, and SS_MF method, which was enhanced by Z. Wu et al (2005). The Baseline method utilizes standard MFCC features without any signal enhancement process, causing its performance to degrade significantly in the presence of noise or environmental interference. The SS method incorporates a Spectral Subtraction process to suppress noise prior to MFCC extraction. This approach improves the system's robustness against background noise; however, it remains limited in maintaining feature stability under complex and highly variable noise conditions.

In contrast, the SS_MF method proposed by Wu and Cao integrates Spectral Subtraction with enhancements at the Mel filterbank stage. This combined strategy produces MFCC features that are more robust and consistent, resulting in improved speaker identification performance, particularly in noisy and real-world environments.

Step Extraction Mel-Frequency Cepstral Coefficients (MFCC)

Tabel 3. Explanation of Step MFCC Extraction

Stage	Function
Pre-Emphasis	Enhances high-frequency components that are weakened during audio recording.
Framing	Divides the audio signal into short segments to ensure quasi-stationary characteristics.
Windowing	Reduces distortion at frame boundaries and minimizes spectral leakage.
Fast Fourier Transform (FFT)	Converts the signal from the time domain to the frequency domain.
Mel Filterbank	Maps frequency components onto the Mel scale according to human auditory perception.
Log Compression	Normalizes the energy range and simulates human loudness perception.

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are one of the core methods in modern face recognition systems due to their ability to perform automatic and hierarchical feature extraction. Unlike traditional face recognition approaches that rely on manually designed features, CNNs are capable of learning visual patterns such as textures, shapes, and facial structures directly from image data.

This capability has also been widely adopted in other computer vision tasks, including object and vehicle type classification, where deep learning models demonstrate superior performance compared to conventional methods. A comparative analysis of deep learning evaluation methods in vehicle type

classification further confirms the effectiveness of CNN-based approaches in learning discriminative visual features from real-world images. Pramana (2024).

CNN-based face recognition methods have demonstrated that convolutional networks can generate feature representations that are more discriminative, stable, and robust to variations in lighting conditions and facial poses. Suryadi et al (2010). CNN architectures typically employ convolutional layers to extract local and high-level features, pooling layers to reduce dimensionality and minimize sensitivity to noise, and fully connected layers to perform identity classification. This layered learning mechanism enables CNN-based systems to achieve superior performance in facial identity recognition compared to conventional feature-based approaches. This layered learning mechanism enables CNN-based systems to achieve superior performance in facial identity recognition compared to conventional feature-based approaches.

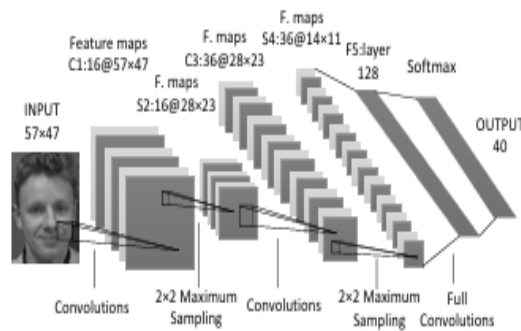


Figure 7. Face recognition extraction

Table 5. Stage of Face Recognition Extraction

Stage of each Layer	Explanation
Input Image	The facial image is provided to the system as the initial input before further processing.
Convolution Layer C1	Extracts low-level features such as edges, corners, and basic facial patterns.
Pooling Layer S2 (2x2 Max Pooling)	Reduces feature map dimensions to accelerate computation and stabilize feature representations.
Convolution Layer C3	Learns more complex facial features, including the shapes of eyes, nose, and mouth.
Pooling Layer S4 (2x2 Max Pooling)	Further reduces feature map size while preserving essential discriminative information.
Fully Connected Layer (F5)	Transforms the extracted feature maps into a numerical identity vector representing the face.
Softmax Output Layer	Determines the most probable facial identity based on class probability distribution.

Table 5 describes the stages of feature extraction in a CNN-based face recognition system, from image input to identity classification. The primary advantage of Convolutional Neural Networks (CNNs) in face recognition lies in their ability to construct feature representations progressively from simple to complex levels. In the early layers, CNNs extract basic visual features such as edges and contours, while deeper layers learn more abstract facial representations, including the structural characteristics of the eyes, nose, and mouth. This hierarchical feature learning significantly facilitates accurate identity classification.

Numerous studies have demonstrated that CNN-based face recognition approaches achieve substantially higher identification accuracy compared to classical methods such as Eigenfaces or Local

Binary Patterns (LBP), particularly in environments with varying lighting conditions and facial orientations. Consequently, CNNs have become a highly relevant and effective solution for biometric security systems that demand high performance, robustness, and real-time operational capability.

This study adopts a quantitative experimental approach aimed at designing, developing, and evaluating the performance of a room security system based on multimodal biometric authentication. The proposed system integrates face recognition using Convolutional Neural Networks (CNN) and voice recognition based on Mel-Frequency Cepstral Coefficients (MFCC), implemented on a Raspberry Pi platform. The experimental process involves real-time facial image acquisition using a RaspiCam camera, followed by face detection using the Haar Cascade method to extract facial regions of interest. The detected facial images are then classified using a CNN model to verify the authenticity of the user's identity.

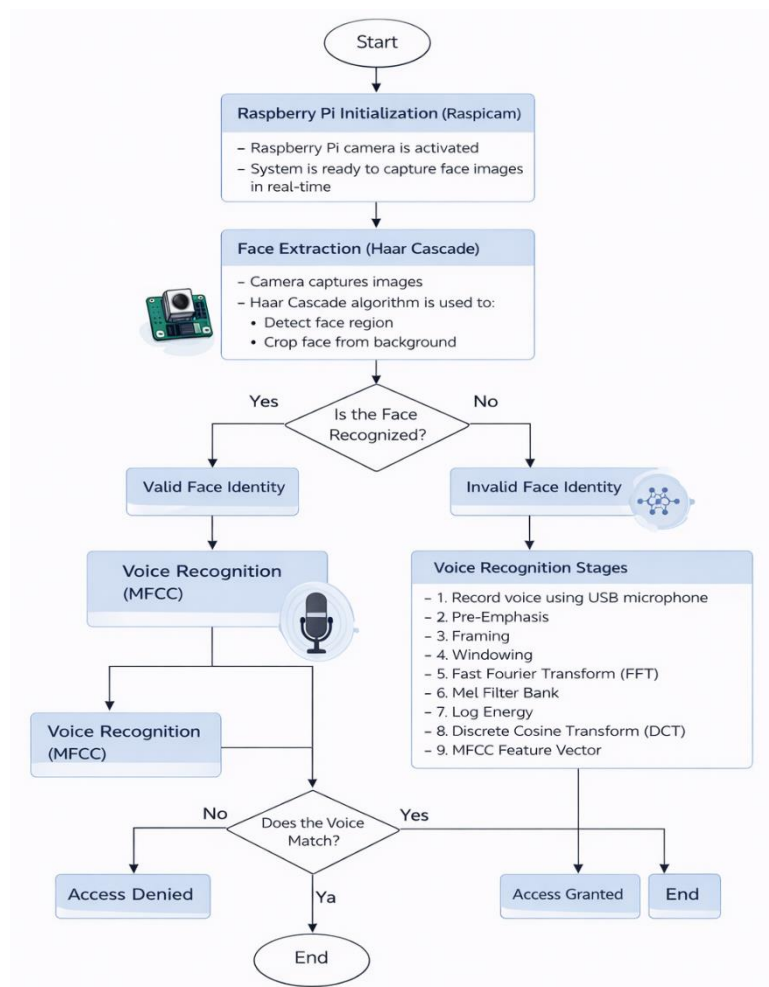


Image 8. Explanation system flow using flowchart

Once facial identity verification is successfully completed, the system proceeds to the second authentication stage through voice recognition using a USB microphone. The captured audio signals are processed using the MFCC algorithm to extract voice feature vectors, which are subsequently compared with previously stored voice templates in the user database.

System performance is evaluated in terms of accuracy, recognition precision, and response time through a series of experiments conducted under various conditions, including different lighting environments, face-to-camera distances, and variations in voice intonation. The experimental results are analyzed to assess the reliability and effectiveness of the proposed system as an efficient IoT-based solution for room security applications.

The system workflow is described as follows:

1. Start
2. Camera Initialization (RaspiCam)
3. Face Detection Using YOLO
4. Face Feature Extraction and Classification Using CNN
5. Facial Identity Validation
6. Voice Verification Request
7. Voice Recognition Process Using MFCC
8. Voice Feature Vector Matching with Database
9. Voice Authentication Decision
10. Access Granted (Door Unlock)
11. End and Ready for the Next Process

Overall, the proposed room security system implements a two-factor biometric authentication mechanism that combines face recognition and voice verification to enhance access security. The process begins with real-time face detection using a camera, followed by identity recognition through a CNN model to verify the facial data against the stored database.

Once the facial identity is successfully validated, the system proceeds to a second authentication stage by requesting voice verification. Voice recognition is performed using MFCC-based feature extraction to ensure that the speaker's voice matches the registered user's voice profile. This dual-layer verification approach effectively minimizes the risk of unauthorized access, such as spoofing attempts using facial images or recorded voices.

After both biometric modalities are successfully authenticated, the system generates a control signal to automatically unlock the door. This workflow demonstrates that the integration of multimodal biometric authentication can provide a more robust, responsive, and reliable security solution, supporting the effective implementation of intelligent IoT-based room security systems.

Results and Discussion

Results

Table 6 presents the experimental of CNN

Kondisi	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	Mean detection latency (ms)
Normal lighting	96.5	96.8	96.2	96.5	220
Low lighting	88.1	89	86.7	87.8	310
Backlight	82.3	83.5	80.8	82.1	345
Normal lighting	90.7	91.2	90	90.6	260
Off-angle, normal light	89.2	90	88	89	235

Table 6 presents the experimental results obtained to evaluate the performance of the CNN-based face recognition system under various lighting conditions, distances, and viewing angles. The evaluation parameters include accuracy, precision, recall, F1-score, and mean detection latency, which represents the average time required by the system to detect and recognize a face.

Under normal lighting conditions at a distance of 0.5–1.0 m, the system achieved an accuracy of 96.5%, a precision of 96.8%, and a recall of 96.2%, with an average detection latency of 220 ms. This scenario produced the best performance, as the lighting and distance were within the optimal range for facial image acquisition.

When tested under low-light conditions, the accuracy decreased to 88.1%, while the average latency increased to 310 ms. This performance degradation is primarily attributed to reduced illumination intensity, which negatively affects the clarity of facial features. The backlight condition resulted in the lowest performance, with an accuracy of 82.3% and a latency of 345 ms, as illumination from behind the subject caused partial loss of facial feature information.

At a distance of 1.5 m under normal lighting, the system achieved an accuracy of 90.7% with an average latency of 260 ms, indicating that reliable face recognition is still feasible at moderately increased distances. Meanwhile, under off-angle conditions of $\pm 30^\circ$, the recorded accuracy was 89.2% with a latency of 235 ms, reflecting a performance decline due to changes in facial orientation relative to the camera.

Overall, the experimental results indicate that lighting conditions are the most influential factor affecting system performance, followed by distance and viewing angle. The average detection latency, ranging from 220 ms to 345 ms, satisfies the requirements for real-time face recognition on edge computing devices, demonstrating the feasibility of deploying the proposed system in practical IoT-based security applications.

Table 7 presents the experimental of Voice Recognition

Noise (SNR)	Accuracy (%)	EER (%)	FAR (%)	FRR (%)	Mean processing latency (ms)
Quiet (30 dB)	94.7	4.2	2.8	5.1	180
Moderate (15 dB)	86.9	8.6	5.5	10.7	200
Noisy (5 dB)	72.4	18.9	14.2	23.6	230

Table 7 presents the experimental results used to evaluate the performance of the MFCC-based voice recognition system under different levels of environmental noise. The evaluation metrics include accuracy, Equal Error Rate (EER), False Accept Rate (FAR), False Reject Rate (FRR), and mean processing latency. Under quiet room conditions with a signal-to-noise ratio (SNR) of 30 dB, the system achieved an accuracy of 94.7%, with an EER of 4.2%, FAR of 2.8%, and FRR of 5.1%. These results indicate stable system performance when background noise is minimal, with an average processing latency of 180 ms, which is suitable for real-time authentication. When the noise level increased to an SNR of 15 dB (moderate noise), the system accuracy decreased to 86.9%, accompanied by an EER of 8.6%, FAR of 5.5%, and FRR of 10.7%. This condition suggests that moderate environmental noise begins to degrade the quality of voice features extracted using the MFCC method.

Under highly noisy conditions with an SNR of 5 dB, system performance declined significantly, with accuracy dropping to 72.4%, EER increasing to 18.9%, FAR reaching 14.2%, and FRR rising to 23.6%. This degradation is primarily caused by strong background noise that interferes with the speech signal, making it difficult to extract reliable speaker-specific features. The average processing latency also increased to 230 ms, reflecting the additional computational effort required under noisy conditions. Overall, the results demonstrate that environmental noise has a substantial impact on MFCC-based voice recognition performance. While the system performs reliably in low-noise environments, its accuracy and error rates deteriorate as noise levels increase, highlighting the importance of noise-robust feature extraction and preprocessing techniques for real-world voice authentication systems.

Overall, the results demonstrate that environmental noise has a substantial impact on MFCC-based voice recognition performance. While the system performs reliably in low-noise environments, its

accuracy and error rates deteriorate as noise levels increase, highlighting the importance of noise-robust feature extraction and preprocessing techniques for real-world voice authentication systems.

Table 7 presents the experimental of Processing Time

Task	Avg latency (ms)	95%-tile latency (ms)	Avg CPU usage (%)	Avg RAM usage (MB)
YOLO face detection (edge variant)	120	210	40	150
CNN face classification	100	180	30	80
MFCC extraction + voice classifier	180	260	35	110
Full multimodal authentication	400	650	75 (peak)	320 (peak)

Table 8 presents the experimental results used to evaluate the computational load and processing time of each component in the multimodal authentication system deployed on an edge device (Raspberry Pi). The observed parameters include average latency, 95th-percentile latency, average CPU usage, and average RAM usage.

The face detection process using the edge-optimized YOLO model achieved an average latency of 120 ms, with a 95th-percentile latency of 210 ms. Resource utilization during this process was recorded at approximately 40% CPU usage and 150 MB of RAM. These results indicate that the object detection stage operates with a moderate computational load on the edge device, while still maintaining acceptable real-time performance.

Overall, the findings demonstrate that YOLO-based face detection can be efficiently executed on resource-constrained edge platforms, providing a balance between detection speed and computational resource consumption. This performance characteristic is essential for ensuring the feasibility and scalability of multimodal biometric authentication systems in real-world IoT-based security applications.

The CNN-based face classification stage requires an average latency of 100 ms with a 95th-percentile latency of 180 ms, while consuming approximately 30% CPU usage and 80 MB of RAM. This stage is relatively lightweight, as it processes only the facial regions obtained from the preceding detection stage rather than the entire image. The voice feature extraction and classification process using Mel-Frequency Cepstral Coefficients (MFCC) exhibits an average latency of 180 ms with a 95th-percentile latency of 260 ms, along with an average CPU usage of 35% and RAM usage of 110 MB. These values indicate that audio signal processing imposes a slightly higher computational load compared to facial classification, due to the additional signal processing steps involved.

Discussion

The CNN-based face classification stage requires an average latency of 100 ms with a 95th-percentile latency of 180 ms, while consuming approximately 30% CPU usage and 80 MB of RAM. This stage is relatively lightweight, as it processes only the facial regions obtained from the preceding detection stage rather than the entire image. The voice feature extraction and classification process using Mel-Frequency Cepstral Coefficients (MFCC) exhibits an average latency of 180 ms with a 95th-percentile latency of 260 ms, along with an average CPU usage of 35% and RAM usage of 110 MB. These values indicate that audio signal processing imposes a slightly higher computational load compared to facial classification, due to the additional signal processing steps involved.

Meanwhile, the complete multimodal authentication process (face and voice) records an average latency of 400 ms with a 95th-percentile latency of 650 ms, while peak resource utilization reaches 75% CPU usage and 320 MB of RAM. The increased computational load is expected, as the system sequentially executes two biometric models. Despite this overhead, the overall processing time remains within acceptable limits for real-time operation on edge devices.

References

- Elnozahy, Seifeldin Sherif Fathy Ali, Senthill C. Pari, and Lee Chu Liang, 'Raspberry Pi-Based Face Recognition Door Lock System', *Internet of Things*, 6 (2025) <https://doi.org/10.3390/iot6020031>
- Fauziyah, Y., & Komalasari, R. (2025). Perancangan sistem informasi to-do list berbasis web untuk mendukung aktivitas akademik mahasiswa. *Politeknik LP3I*.
- Masud, Mehedi, Ghulam Muhammad, Hesham Alhumyani, Sultan S. Alshamrani, Omar Cheikhrouhou, Saleh Ibrahim, and others, 'Deep Learning-Based Intelligent Face Recognition in IoT-Cloud Environment', *Computer Communications*, 152 (2020), 215–22 <https://doi.org/10.1016/j.comcom.2020.01.050>
- Pramana, A. L., "Comparative Analysis of Deep Learning Method Evaluation on Vehicle Type Classification," *Jurnal Ilmu Komputer dan Desain Komunikasi Visual*, vol. 9, no. 1, pp. 451–465, 2024, <https://doi.org/10.55732/jikdiskomvis.v9i1.1252>
- Pulli, Kari, Anatoly Baksheev, Kirill Korniyakov, and Victor Eruhimov, 'Mobile Computer-Vision Technology Will Soon Become as Ubiquitous as Touch Interfaces', *Communications of the ACM*, 55 (2012), 61–69 <https://doi.org/10.1145/2184319.2184337>
- Suryadi, Dikky, Cut Susan Otiva, T. Irfan Fajri, Uli Wildan Nuryanto, and Muhammad Lukman Hakim, 'Optimasi Kinerja Sistem IoT Menggunakan Teknik Edge Computing', *Jurnal Minfo Polgan*, 13 (2024), 1456–61 <https://doi.org/10.33395/jmp.v13i2.14102>
- Wu, Z., and Z. Cao, "Improved MFCC-based feature for robust speaker identification," *Tsinghua Science and Technology*, vol. 10, no. 2, pp. 158–161, 2005, doi: 10.1016/S1007-0214(05)70048-1