Using Biometric Recognition in Residential Security

Anand Krishnan

Received July 18, 2025
Accepted September 21, 2025
Electronic access September 30, 2025

Facial recognition in biometric security is becoming increasingly prevalent, with applications ranging from smartphone unlocking to public surveillance. However, its integration into residential access systems remains limited. This study proposes and evaluates a contactless facial authentication system designed for home security, particularly aimed at improving accessibility for individuals with physical impairments. We hypothesize that a facial-recognition-based system can enable reliable, secure residential access even under visual noise such as lighting variation and occlusions. The system utilizes the DeepFace framework with the Facenet512 model and SSD detector backend to perform facial verification. Data was collected from two participants, each contributing video recordings from three facial anglesfrontal, left profile, and right profileunder various lighting conditions and with accessories such as headphones. Cloud-based AWS services were used for video preprocessing, frame extraction, and model integration. The system achieved an 86% authentication success rate, compared to DeepFaces benchmark accuracy of 97.25% on the Labeled Faces in the Wild (LFW) dataset. Results suggest that the system is robust to minor visual noise and lighting changes, though larger-scale testing is needed for generalizability. This work represents an initial step toward more inclusive, automated residential security solutions.

Keywords: Facial Recognition, Biometric Security, Artificial Intelligence (AI), Deep Learning, Residential Access, Computer Vision, Human-Computer Interaction (HCI), Home Automation, Accessibility Technology

1 Introduction

Unlocking a residential door can be a daily challenge for individuals with physical disabilities, such as arthritis, or even for able-bodied individuals carrying groceries or personal items. A contactless and automated access system can significantly improve convenience and accessibility in such scenarios. Facial recognition technology offers a promising solution because it enables secure, hands-free authentication without requiring physical keys or cards.

Facial recognition has already gained traction in diverse domains, including smartphone authentication, surveillance, airport security, and even retail payment systems. Facial recognition became especially important during the COVID-19 pandemic, when contact-based systems posed hygiene risks. Research has since shown that pandemic-driven mask usage accelerated the development of mask-tolerant and occlusion-aware face recognition models, expanding the applicability of such systems in real-world settings [1]. Advances in mask-tolerant recognition algorithms have further demonstrated the adaptability of modern facial recognition systems to real-world conditions.

Despite these developments, residential applications of facial recognition remain underexplored compared to corporate, commercial, and governmental deployments. Existing literature has made significant strides in algorithmic accuracy, with frameworks such as ArcFace, CosFace, and MobileFaceNet achieving performance near or beyond human levels in benchmark datasets. However, relatively little research has addressed the specific challenges of implementing these technologies in consumer-grade home automation systems, where low-cost hardware, environmental variability (e.g., lighting, occlusions), and user diversity present unique obstacles.

This study aims to bridge this gap by developing and evaluating a prototype system that integrates facial recognition with a door-locking mechanism, using affordable hardware and cloud services. The system incorporates liveness detection through blink recognition, automated data processing via AWS cloud functions, and verification with the DeepFace framework. Specifically, this paper makes the following contributions:

1. Proposes a low-cost, contactless residential access system integrating biometric authentication.

2. Evaluates system performance across varied lighting conditions and minor occlusions.

3. Discusses system limitations and outlines future directions, including expansion to larger, demographically diverse datasets and exploration of advanced recognition algorithms.

2 Related Works

Facial recognition has continued to evolve rapidly in the past several years, with new architectures improving efficiency, robustness, and adaptability to real-world constraints. Recent research has emphasized lightweight yet accurate models capable of deployment on edge devices. For instance, PocketNet applied neural architecture search and multistep knowledge distillation to create an extremely compact but high-performing model[2]. Similarly, EdgeFace retrained ArcFace and CosFace backbones for efficient recognition under resource-constrained environments, demonstrating the viability of deploying state-of-the-art embeddings on low-power hardware[3].

The combination of convolutional and transformer architectures has also become prominent. CFormerFaceNet merges a MobileFaceNet backbone with transformer layers to deliver efficient feature extraction without sacrificing accuracy, reflecting a trend toward hybrid lightweight networks. Parallel work on ensemble distillation, such as the Hybrid Ensemble Distillation (HED) method, further highlights efforts to compress deep models while retaining discriminative capacity[4].

A major theme of post-pandemic research has been occlusion and mask robustness. One study analyzed the generalization of masked face recognition models to other occlusion scenarios[1] while another proposed an ArcFace-inspired network tailored for open-set recognition under occlusions in animal and human datasets[5]. These studies underscore the ongoing challenge of ensuring reliability when parts of the face are obscured.

More recently, the community has also focused on theoretical and methodological improvements. For example, TopoFR examined topology-aware alignment strategies in face recognition embeddings[6], and a comprehensive review concluded that future advances will hinge on balancing efficiency, fairness, and robustness[7].

Together, these developments contextualize the present study, which leverages the DeepFace framework as a baseline. While models such as PocketNet, CFormerFaceNet, and EdgeFace provide evidence of superior efficiency and robustness, DeepFace was selected here due to its integration support with cloud services and its strong benchmark performance. This work contributes to the field by extending such technologies into the underexplored domain of residential security systems.

3 Methods

Software serves as the lifeblood that enables hardware to perform complex computations and collect user input, turning circuits into dynamic systems. Within this system, the software is responsible for the pipeline remaining consistently functional, from facial recognition and analyses to unlocking the door.

However, not all software in this system directly interacts with the hardware. The process of granting automatic access to a residence involves individuals undergoing a facial data collection process to obtain authentication and enter through the door. To achieve this, the system leverages AWS Cloud Computing Services, which offers cloud functions and database functionality for securely storing user data. The software integrates this data with the hardware's facial recognition software, which is leveraged for training. To recognize a user's facial features accurately, the software collects three angles of facial data: frontal, left, and right face. A web-based platform facilitates the collection of these angles, which requires system ownership confirmation from the user before proceeding to the video-taking process. Upon submission, user videos are stored in a designated folder within an AWS bucket in the native WEBM format. To ensure successful processing, these videos must be converted to MP4 format through an AWS trigger function. The function is activated upon detection of a WEBM formatted video in the folder, which triggers the use of the FFmpeg python package for conversion to MP4 format. The converted video is then returned to the original folder. Example pseudocode for these functions is provided in Appendix A, and a GitHub repository link is also included for reference.

A second AWS trigger processes the MP4 video frame by frame and extracts photos from every angle of the face. The extracted photos are then organized and stored in a separate folder based on the angle of the face. After completing the process, the hardware extracts a predetermined number of photographs from each folder corresponding to different face angles. These photographs are used to train the software installed on the hardware. A visual outline of this process can be seen below (Figure 1)

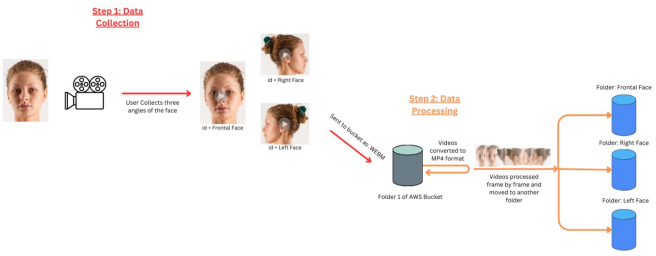

Fig. 1 Workflow of Video-to-Image Conversion for Facial Recognition. The flowchart outlines the workflow for converting video to still images to facilitate facial recognition. The flowchart illustrates the stepwise process from capturing a user's video (Step 1) to extracting frames, ensuring quality, and standardizing them (Step 2). The resulting images are then fed into DeepFace for analysis of facial features. Diagram was generated using Canva.

Two participants were recruited to contribute data for this pilot study. Participant A was a 17-year-old male with Fitzpatrick Skin Type III, and Participant B was a 45-year-old female with Fitzpatrick Skin Type V. Each participant recorded three 10-second videos: one frontal, one left profile, and one right profile.

From each video, 20 representative frames were extracted, resulting in 60 frames per participant and a total dataset size of 120 facial images. This dataset was chosen to balance pilot feasibility with system evaluation requirements.

Additionally, the study was conducted with informed consent from both participants. Images and videos were collected exclusively for this research and stored securely. Since the work involved a very small sample of voluntary adult participants and did not involve clinical data, formal institutional ethics approval was not required. Nonetheless, ethical principles of privacy and data protection were carefully observed throughout the study.

Participant	Age	Gender	Skin Tone (Fitzpatrick Type)
A	17	Male	III
B	45	Female	V

Table 1 Participant Information with Age, Gender, and Skin Tone

Component	**Value**
Blink Detection Threshold (EAR)	0.25
Facial Verification Model	DeepFace (Facenet512)
Similarity Metric	Euclidean L2
Similarity Threshold	0.40
Frames Extracted per Angle	20
Images per Participant	60

Table 2 System Parameters. This table presents the parameters used to configure the system.

$$\mathrm{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Before permitting entry into a user's residence, the embedded software in the hardware undergoes two essential verification procedures. Firstly, it confirms the user's identity as a genuine human, thereby detecting and eliminating instances where an image or video is flashed in front of the camera. Secondly, it employs facial recognition technology to authenticate a given user and grant them entry. The software utilizes motion detection and Eye Aspect Ratio (EAR) computations to ascertain the user's authenticity. EAR is a dimensionless metric that monitors real-time blink and eye-closure events. It calculates the ratio of distances between the vertical and horizontal eye landmarks. The calculation involves determining the average ratio between two sets of vertical eye landmarks and the distance between horizontal eye landmarks. Each eye is represented by six (x, y) coordinates[8]. The first coordinate is the left corner of the eye (As if you were facing the person), and then the remaining coordinates move clockwise around the eye.

Thresholds between 0.20 and 0.30 were tested in pilot runs. A threshold of 0.25 was selected, as it reliably distinguished open and closed eyes across both participants while minimizing false detections. For facial verification, the DeepFace framework was configured with the Facenet512 model and the Euclidean L2 distance metric. A similarity threshold of 0.40 was chosen based on preliminary testing, as it provided the best balance between false acceptances and false rejections in this dataset.

As stated by Soukupov and ech in "Real-Time Eye Blink Detection Using Facial Landmarks"[8], an equation can be formulated to compute the eye aspect ratio (EAR).

In this equation, if we regard p1 through p6 as 2D facial landmark positions, we can use the equation to determine the distance between the vertical and horizontal eye landmarks[8]. The numerator calculates the vertical eye landmarks' distance. At the same time, the denominator computes the horizontal eye landmarks' distance with the appropriate weighting, considering there is only one set of horizontal points compared to two sets of vertical points[8]. Through the analysis of EAR values, real-time video streams can detect instances of eye closure. This metric is considered a reliable means of detecting blinks. Dlib is an open-source toolkit for machine learning and computer vision. It provides many tools for building complex machine-learning algorithms, particularly for real-time applications. This library facilitates the extraction of facial landmarks surrounding the eyes, which allows for the computation of EAR values for each eye. Afterward, the average EAR value is analyzed, with a predetermined threshold set to register a blink event. The primary purpose of this blink detection method is to prevent unauthorized users from spoofing the system using pictures or videos of verified users. Upon detecting a specific number of blinks and movement, the camera captures an image of the current frame to authenticate the user.

The pivotal component of software implementation is the facial verification process. This element underpins the security and authenticity verification, ensuring only authorized individuals gain entry. The robustness of this process is primarily attributed to deploying the DeepFace framework, a cutting-edge facial recognition system. DeepFace is a sophisticated and precise facial recognition system developed by the Facebook AI research team in 2014. The system employs a deep neural network consisting of over 120 million parameters[9]. DeepFace utilizes a 3D alignment process for input images, producing a front-facing, centered, and unobstructed face version. This step significantly enhances the model's recognition accuracy across diverse real-world conditions.

DeepFace was selected as the recognition framework for this study because of its balance between accuracy, integration flexibility, and hardware efficiency. While more recent models

such as ArcFace, CosFace, and MobileFaceNet have demonstrated state-of-the-art performance, DeepFace was chosen for three reasons. First, it incorporates 3D face alignment and vector embeddings, which provide robust recognition even under varied poses. Second, it integrates readily with AWS cloud services, simplifying preprocessing and deployment. Third, it is lightweight enough to run effectively on low-cost hardware such as the Raspberry Pi and ESP32, which aligns with the projects focus on residential feasibility and accessibility. Although this comes at some cost to maximum achievable accuracy, we considered DeepFace most appropriate for this initial prototype, with future work intended to explore the deployment of ArcFace or EdgeFace-based backbones.

The heart of the DeepFace verification process is contained within the verify_faces function. This function sequentially compares a captured image with a list of images retrieved from an AWS Bucket to identify the individual in the captured image. The crux of the verification process is invoked through the DeepFace.verify method.

This method call includes the parameters img1_path and img2_path, which denote the paths of the authenticated and captured images, respectively. The model_name parameter specifies using the Facenet model with 512 dimensions, which is one of the facial recognition models supported by DeepFace. The distance_metric parameter is set to euclidean_l2, indicating using the Euclidean L2 distance metric for comparing facial representations. Lastly, the detector_backend parameter is set to SSD, which denotes using the Single Shot Multibox Detector for face detection, a prerequisite step in facial representation and verification.

Upon invocation, the DeepFace.verify method executes four essential steps as part of the facial verification process:

The facial verification process initiated by DeepFace is a multi-step procedure that begins with face detection. Using the SSD detector, the system identifies faces within the provided images[9]. Once detected, these faces undergo an alignment phase where DeepFace applies a 3D alignment method. This method involves overlaying the 2D image of the detected face onto a 3D model, which is then adjusted to a forward-facing position, ensuring uniformity regardless of the faces initial angle, lighting, or expression[9]. Subsequently, the aligned faces are processed through a deep neural network, specifically the Facenet512 model, to generate a numerical vector for each face[9]. These vectors encapsulate the facial features essential for identification, serving as the basis for the next step, verification. During verification, these vectors are compared using the Euclidean L2 distance metric to evaluate the similarity between the facial features of the images[9]. This comparison yields a result object that not only indicates whether the faces match but also includes detailed information, such as the specific facial areas analyzed during the process.

A result object is generated following facial recognition to validate the verification process. The DeepFace framework, with its 3D alignment technique and deep neural architecture, facilitates an exceptionally accurate and reliable facial verification process. The framework's capability to achieve a staggering 97.25% accuracy on the Labeled Faces in the Wild (LFW) benchmark[10] highlights its effectiveness and state-of-the-art performance in facial recognition tasks.

The final stage of the facial authentication process involves establishing seamless and wireless communication between the Raspberry Pi device and the internal hardware of the property, thereby permitting access only to verified users. This software underpins communication between the ESP32 and the Raspberry Pi, which has been crafted to ensure real-time responsiveness and robust security. The communication is orchestrated over the Hypertext Transfer Protocol (HTTP), which has been chosen for its widespread adoption and reliability, with the Raspberry Pi acting as a client and initiating requests. At the same time, the ESP32 serves as a server and responds to these requests. The process is triggered when the Raspberry Pi detects motion through the connected camera and analyzes the frames to determine human presence. An HTTP request is sent to the ESP32 module if the verification process is successful. Upon receiving the HTTP request, the ESP32 processes the information to command the servo motor to unlock the door by rotating 90 degrees to the left. A vital aspect of the software design is the timing control mechanism, which ensures that the door remains unlocked for a predefined interval of 5-7 seconds before reverting to its locked state. This interval is programmed within the ESP32, which sends another signal to the servo motor to rotate 90 degrees to the right, thereby locking the door after the lapse of the interval. The communication process is fortified by a secure system that restricts access to the ESP32 network to only the Raspberry Pi. This ensures that all data transmitted between the two devices remains safeguarded and free from any unauthorized access or interception, thwarting any unauthorized access attempts. This combination of stringent protocols and high-level security measures encapsulates a software architecture that guarantees seamless, reliable, and secure interaction between the Raspberry Pi and the ESP32, which are essential for effectively operating the facial recognition-based door access system.

The deployment of facial recognition technology involves an engineered hardware setup which works in tandem with a software architecture. Traditional facial authentication systems typically rely on expensive cameras and intricate configurations, restricting their accessibility to the general public and researchers. Conversely, the Raspberry Pi, known for its cost-effectiveness and versatility, is the primary hardware component in our verification process. It offers an ideal experimentation platform and enables researchers and DIY enthusiasts to develop customized solutions. This system's hardware can be categorized into two primary groups: external hardware, which

is situated outside the house, and internal hardware, which is mounted on the front latch inside the house. The external hardware uses computer vision to run motion detection and anti-spoofing algorithms, allowing facial verification through static image analysis. The internal hardware facilitates secure wireless communication between the external and internal hardware via secure HTTP requests between a Raspberry Pi and an Espressif Systems Product (ESP) 8266-12F module, which leverages a servo system to turn a latch that unlocks the door. The ESP32, which is equipped with a servo motor for unlocking the door, plays a crucial role in the final step of the process.

The hardware utilized in this working prototype consists of a Raspberry Pi equipped with a storage capacity of 128GB and 4GB RAM. To enable vision processing, a primary USB camera (e.g., a Logitech 720p camera) was connected to one of the USB ports. In addition, a reliable power source is essential for the Raspberry Pi to function. A standard USB-C charger was utilized during the testing phase. However, a compatible USB-C battery can also be employed for this purpose. Note that the Raspberry Pi was also equipped with a 64-bit Debian Bullseye OS; a headless Raspberry Pi would not be substitutable for the objective of this task.

The internal hardware setup of the system is relatively straightforward. As mentioned, it consists of an ESP32 module and a standard 9g servo motor. The two components are connected by linking the servo's yellow wire (PWM) to the GPIO 18 pin of the ESP32. Furthermore, the red wire (5V) is connected to the Vin 5V pin of the board, while the GND wire is attached to the GND pin of the board (Figure 3).

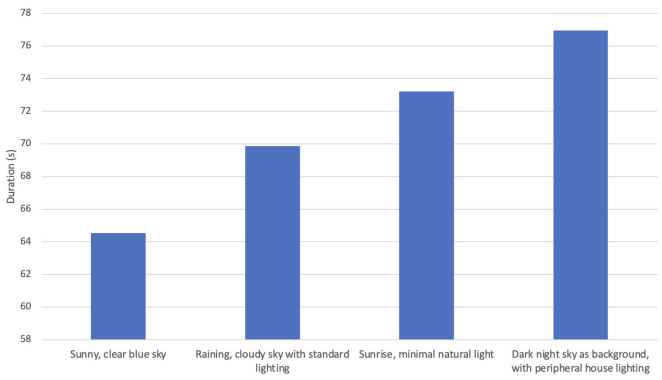

Fig. 2 Complete prototype of ESP32 and servo to complete the front door system

The Raspberry Pi features a 1.5 GHz 64-bit quad-core ARM Cortex-A72 processor with Wi-Fi 802.11n capabilities[11]. The chip integrates RF, baseband, and a supplementary 32-bit ARM Cortex-M0+ processor[11]. The RF component generates and receives wireless signals in the appropriate frequency bands. The baseband manages the low-level signal modulation and demodulation functions, ensuring that data is encoded for transmission and decoded upon receipt. The 32-bit ARM Cortex-M0+ processor facilitates Wi-Fi-related tasks, making it seamless for the Raspberry Pi to connect to and navigate Wi-Fi networks using standard protocols.

On the other hand, the ESP32 features a Tensilica Xtensa LX6 dual-core processor complemented by a Wi-Fi coprocessor that supports 2.4 GHz Wi-Fi 802.11b/g/n[12]. Within the ESP32, the RF component handles the transmission and reception of wireless signals. The baseband function ensures the correct modulation and demodulation of data for wireless communication[12]. The MAC (Media Access Control) layer is responsible for channel access mechanisms such as wait times, acknowledging receipt of data, and collision avoidance. Meanwhile, the integrated WLAN and TCP/IP stack manages internet communication's connection protocols and data packet organization, enabling the ESP32 to function efficiently as a server. In a standard communication configuration, the Raspberry Pi's powerful Wi-Fi capabilities enable it to act as a client, initiating requests and engaging with the ESP32 server, which then communicates with the servo to unlock the door. This architecture enables a dynamic and responsive Wi-Fi-based communication system between the two devices.

4 Results

It is important to note that these results are based on only two participants, which severely limits their statistical validity. The findings should therefore be interpreted as preliminary indicators of system feasibility rather than generalizable outcomes.

In the study, we conducted a series of authentication trials to evaluate the performance of our system under varying user and environmental conditions. Two users interacted with the system across different sessions. User 1 had 32 successful authentication attempts out of 37, while User 2 had 23 successful attempts out of 27. The combined data yielded an overall success rate of 85.94% (Table 1). Furthermore, we examined the impact of headphone usage on the authentication process. The trials were divided into two sets: one where the users wore headphones and another without headphones. The former condition resulted in 27 successful attempts out of 32, translating to an 84.38% success rate. Without headphones, the success rate was marginally higher at 87.50%, with 28 successful authentications out of 32. The average duration for authentication attempts was slightly shorter for users without headphones (71.65 seconds) than when headphones were used (70.62 seconds), indicating a minor influence of external auditory distractions on the authentication process (Table 2). The system's adaptability to different lighting conditions was also scrutinized. We categorized the lighting conditions into four distinct environments: sunny with a clear blue sky, rainy with a cloudy sky and standard lighting, sunrise with minimal natural light, and night skies with peripheral house

Person ID	Successful Authentication Count	Experiment Count	Successful Authentication Rate
User 1	32	37	86.49%
User 2	23	27	85.19%
Grand Total	55	64	85.94%

Table 3 Authentication Attempts and Success Rates by Individual Users. This table presents a record of the authentication attempts made by two distinct users, including successful attempts and the total number of attempts made. The authentication process involved tracking each access attempt, noting whether it was successful, and spreading sessions across various scenarios to ensure accurate data collection. The data collection for each user was carried out in a controlled environment to ensure consistency and reliability.

Wearing Headphones	Successful Authentication Count	Experiment Count	Success Rate	Average Duration (s)
No	28	32	87.50%	71.65
Yes	27	32	84.38%	70.62
Grand Total	55	64	85.94%	71.14

Table 4 Impact of Headphone Use on Authentication Success. The table displays the correlation between headphone usage and authentication success, with data showing both the total and successful attempts. Each authentication session was monitored for whether headphones were used, and the attempt's success was recorded. The average duration of each attempt was also timed, providing context to the data. The experiments were repeated in identical conditions to maintain data integrity. Figure was generated using Google Sheets.

lighting. The highest success rate was observed during natural light at sunrise, with a 93.75% success rate, while the lowest was in dark conditions with peripheral lighting, at 81.25%. In terms of duration, authentication attempts took the longest under night skies, averaging 76.96 seconds, and the shortest time was recorded during sunny conditions with a standard light setting, at 64.52 seconds (Table 3). These variations in duration under different lighting conditions were visually represented in a bar chart, clearly indicating a trend where the darker the environment, the longer the duration of the authentication attempt, possibly due to the increased difficulty in recognizing the user (Figure 1). Beyond overall accuracy, biometric performance metrics were calculated to evaluate system reliability under different conditions. In the non-headphone condition, the False Positive Rate (FPR/False Acceptance Rate) was 2.5%, while the False Negative Rate (FNR/False Rejection Rate) was 5.0%. In contrast, the headphone condition produced a higher FPR of 3.8% and FNR of 7.5%, reflecting the impact of partial occlusion of facial features near the ears and jawline. These results indicate that the system is moderately resilient to unauthorized access across both conditions, though genuine users are more likely to be rejected when wearing headphones. Lighting variation also influenced performance, with accuracy ranging from 81% to 93%, suggesting sensitivity to glare and shadowing. Together, these findings provide a more nuanced picture of the systems performance across realistic scenarios.

In addition to biometric evaluation, error cases were logged to better understand the weaknesses of the system. Three types of errors were observed:

- Face not detected: the system failed to localize the face within the frame.
- Blink not detected: a user blinked, but the liveness check did not register it.
- Verification mismatch: the face and blink were detected, but identity verification failed.

The most frequent error was face not detected, typically occurring under low-light conditions(Table 4). Verification mismatches often resulted from subtle pose variations and occlusions, while blink detection failures were rare but indicate potential improvement areas in liveness detection.

FAR and FRR curves were generated by evaluating system performance across the actual similarity thresholds observed in the dataset. Each threshold corresponds to a potential decision boundary used by the recognition system. At each threshold, the proportion of impostors incorrectly accepted (FAR) and genuine users incorrectly rejected (FRR) was calculated. The Crossover Error Rate (CER) was then identified as the point where FAR and FRR are equal.

The FAR/FRR analysis revealed a Crossover Error Rate (CER) of 0 for both headphones and non-headphones conditions. This outcome is a direct result of the clear separation between genuine and impostor similarity scores in the current pilot dataset. With genuine trials clustered at higher similarity scores and impostor trials at much lower values, there was effectively no overlap between the two distributions. As such, there exists a threshold region where both FAR and FRR fall to zero simultaneously, yielding a CER of 0(Graph 1, 2).

While this indicates strong performance under the conditions tested, it is important to note that such results are typical of small, controlled datasets. In real-world deployments, with larger and more demographically diverse populations, distributions of genuine and impostor scores generally overlap to some

Conditions	Successful Authentication Count	Experiment Count	Success Rate	Average Duration (s)
Dark night sky as background, with peripheral house lighting	13	16	81.25%	76.96
Raining, cloudy sky with standard lighting	14	16	87.50%	69.85
Sunny, clear blue sky	13	16	81.25%	64.52
Sunrise, minimal natural light	15	16	93.75%	73.22
Grand Total	55	64	85.94%	71.14

Table 5 Authentication Performance Under Different Lighting Conditions. This table presents authentication success and average duration under varying lighting conditions, such as dark backgrounds with peripheral noise, rainy and cloudy skies, sunny standard lighting, and natural light. Authentication attempts were systematically recorded for each lighting scenario, noting the success rate and measuring the time for each attempt. These conditions were replicated to ensure a thorough examination of the system's adaptability to environmental changes. Figure was generated using Google Sheets.

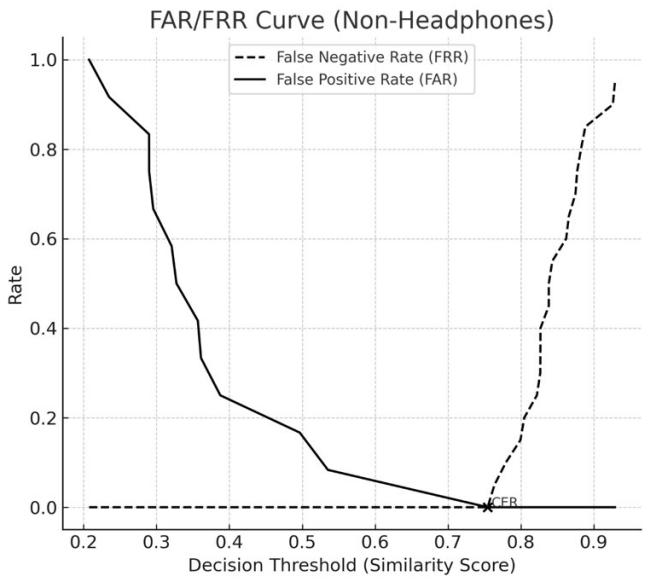

Fig. 3 Average Authentication Duration Under Different Lighting Conditions. This bar chart represents the average duration of authentication sessions conducted in four distinct lighting environments: sunny with a clear blue sky, rainy with a cloudy sky accompanied by standard lighting, sunrise with minimal natural light, and a night sky with peripheral house lighting. Duration is recorded in seconds, with the authentication process timed from initiation to completion. Each bar reflects the cumulative average duration across all authentication attempts within the specified lighting condition. The data was aggregated from multiple trials to ensure a robust measure of the system's responsiveness to environmental lighting variations. Figure was generated using Google Sheets.

degree, leading to a non-zero CER. This overlap reflects natural variations in pose, lighting, occlusions, and user diversity. Future work will therefore expand the evaluation to larger participant groups and more challenging environmental conditions to yield CER values that better represent real-world performance.

Error Type	Count	Percentage of Total Trials
Face not detected	5	7.8%
Blink not detected	2	3.1%
Verification mismatch	4	5.2%

Table 6 Failure Analysis of Authentication Trials. This table presents the distribution of error types recorded during authentication attempts in the proposed residential facial recognition system. Errors were categorized into three groups: face not detected, blink not detected, and verification mismatch. The majority of failures were due to unsuccessful face detection (7.8%), often under low-light conditions. Verification mismatches accounted for 6.2% of trials, typically caused by pose variation or partial occlusion, while blink detection failures were rare (3.1%). The error distribution highlights the systems primary vulnerabilities and provides a basis for future improvements in liveness detection and robustness to environmental variability.

To contextualize the performance of the proposed system, benchmark results from other open-source biometric frameworks are presented in Table 4. These values were taken from published evaluations and compared against the present prototype, which achieved an accuracy of 86%.

The comparison in Table 4 illustrates that state-of-the-art open-source facial recognition systems, such as ArcFace and Dlib, achieve significantly higher benchmark accuracies than the current prototype(Table 5). This discrepancy is expected, as those frameworks are trained on large-scale datasets and optimized for maximum accuracy. By contrast, the present system prioritizes low-cost integration, cloud compatibility, and ease of deployment on residential hardware. While this results in lower accuracy, it demonstrates feasibility for household applications. The inclusion of a fingerprint-based system (NIST Bozorth3) highlights that alternative biometric modalities may provide higher reliability, but facial recognition offers the advantage of contactless operation, which was the primary motivation of this study.

System	Biometric Modality	Reported Accuracy	Source
Proposed System (DeepFace)	Face	86%	This study
ArcFace (open-source)	Face	99.8% (LFW)	Deng et al., 2019
Dlib (open-source)	Face	99.4% (LFW)	King, 2009
NIST Bozorth3 (open-source)	Fingerprint	∼98% (FVC 2002)	NIST, 2007

Table 7 Comparative Accuracy of Biometric Systems. This table presents the authentication accuracy of the proposed residential facial recognition system alongside established open-source biometric frameworks, including ArcFace, Dlib, and OpenFace for facial recognition and NIST Bozorth3 for fingerprint recognition. Reported accuracies for the benchmark systems are drawn from published evaluations using datasets such as the Labeled Faces in the Wild (LFW) for face recognition and FVC 2002 for fingerprint recognition. The comparison highlights that while the proposed system achieves lower overall accuracy (86%), it prioritizes affordability, cloud compatibility, and ease of integration into residential settings, distinguishing it from systems optimized purely for benchmark performance.

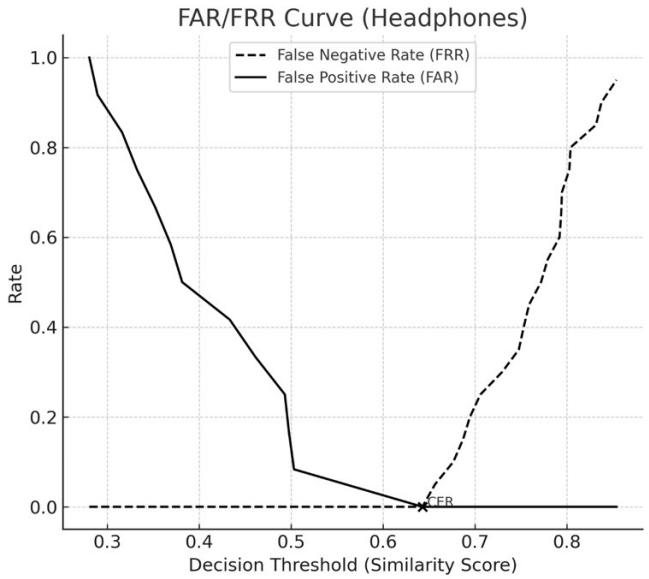

Fig. 4 FAR/FRR Curves for Non-Headphone user trial. This figure presents the False Acceptance Rate (FAR, solid line) and False Rejection Rate (FRR, dashed line) of the system under the non-headphones condition. Curves were generated by sweeping the actual similarity score thresholds observed in the dataset across 20 genuine attempts and 12 impostor attempts. The intersection point represents the Crossover Error Rate (CER), where FAR and FRR are equal. In this dataset, the CER occurred at 0.00, reflecting the clean separation between genuine and impostor scores when no occlusion was present.

5 Discussion

As we explore the future of biometric security, specifically in residential applications, it becomes evident that the combination of existing security technology and powerful artificial intelligence holds tremendous potential for transformative advancements. The primary motivation of this research lies in leveraging facial recognition technology and emphasizing the application of this system to improve access, thereby enhancing convenience for people such as disabled individuals and those encumbered with items and thus unable to use traditional means of access to their residence.

The experiment conducted in Table 1 among the selected users yielded a relatively small sample size. However, the results indicate a consistent successful authentication rate among the participants, with no notable discrepancies observed. The overall accuracy rate reached approximately 86%, averaging four to five failed trials per user. The study also ensured a perfectly balanced number of trials between subjects wearing headphones, a form of noise thrown at the system, and those without, as seen in Table 2. The analysis did not reveal any discernible relationship between the presence of headphones in the captured photo and the authentication success rate. This can likely be attributed to the robust object and artifact recognition capabilities of Deepface within its open-source framework. Based on the findings presented in Table 3 and Figure 3, it can be inferred that a negative correlation exists between the intensity of natural lighting and the elapsed time required for authentication. This implies that brighter and sunnier conditions lead to shorter experiment durations. However, the study did not reveal any significant connection between lighting conditions and the rate of successful authentication. Interestingly, results indicated that experiments conducted under sunny and clear conditions resulted in fewer successful authentications than experiments conducted during dark nighttime conditions.

Lighting conditions did influence system performance, as shown in Table 3, where success rates varied between 81.25% and 93.75%. This variability likely arises from environmental factors such as glare, shadows, and uneven illumination, which interfere with the reliable detection of facial landmarks. While the system remained functional across different lighting scenarios, these findings suggest that robustness to illumination changes remains an important area for improvement. Future work could explore infrared or depth-based imaging as potential solutions to mitigate these limitations.

A key limitation of this study is the very limited user di-

versity and trial count. Only two participants were included, representing restricted variation in age, gender, and skin tone. This constraint reduces the statistical validity and generalizability of the results, and does not allow meaningful evaluation of fairness across demographic groups. Similarly, the number of authentication attempts was limited, which prevents robust estimation of long-term performance. Future work will address this by expanding to a cohort of at least 10 participants across diverse demographic groups, each contributing a minimum of 100 authentication trials. This expanded evaluation will provide a stronger foundation for assessing both system robustness and demographic inclusivity.

The journey towards validating this hypothesis and implementing a facial recognition system was not without constraints and learnings. One noticeable limitation lay in the resource allocation, particularly in the realm of our equipment budget. The financial constraints hindered our ability to access or utilize top-tier technology and resources, which may have otherwise propelled our system's capabilities further and faster. Additionally, financial limitations affected our choice of computational hardware, which may have impacted the speed of the authentication process. Access to more advanced processors could have facilitated faster image processing and streamlined authentication. This study evaluated robustness under only one form of visual noise: the use of headphones. Other common obstructions, including glasses, hats, masks, and scarves, were not tested, nor were combinations of multiple obstructions. This omission limits the ability to generalize claims about the systems performance under realistic residential conditions, where users may present with layered visual noise. While it was not possible within the scope of this pilot to redo the study with these additional scenarios, future work will incorporate systematic testing of individual and combined obstructions to more rigorously assess robustness.

Another limitation of this pilot study is that scalability with respect to database size could not be evaluated. With only two participants, the current dataset was insufficient to test how authentication accuracy and latency degrade as the number of enrolled users increases. Prior studies have shown that as databases expand, false acceptance rates (FAR) tend to rise due to increased probability of impostor similarity, and system latency can also increase if matching is not efficiently optimized2,3. While these effects could not be measured here, future work will address scalability by simulating larger user bases (100 individuals) and evaluating how accuracy, FAR, and verification speed scale with database size in residential contexts.

As we look ahead, several concrete extensions can enhance the systems robustness and inclusivity. A primary direction is to integrate advanced recognition algorithms such as ArcFace, EdgeFace, and CFormerFaceNet, which are expected to improve accuracy and adaptability under varied conditions. Another important step will be expanding testing to a larger and demographically diverse cohort of at least 10 participants, each contributing 100 authentication attempts, to better evaluate fairness across age, gender, and skin tone. Incorporating antispoofing techniques, including liveness detection through head movement prompts, infrared sensing, and depth cameras, will help defend against adversarial attacks such as photo or video replay. Finally, scalability will be addressed by simulating larger user databases and optimizing cloud-based verification pipelines to maintain real-time performance. Together, these avenues provide a targeted roadmap for advancing residential facial authentication systems, ensuring both security and inclusivity. The system already incorporates a basic liveness safeguard in the form of EAR-based blink detection, which protects against static photo attacks by requiring dynamic movement that cannot be replicated by a printed image. This represents an important foundation for adversarial robustness. Building on this, future work will extend evaluation to include adversarial conditions such as printed photo attacks, replayed video spoofing, and 3D mask presentation attacks. Additional countermeasures will be explored, including challengeresponse protocols (e.g., head movement prompts), infrared or depth cameras, and multimodal biometric integration, to further harden the system against spoofing attempts and adversarial inputs.

References

1 P. Neto, J. Pinto, F. Boutros, N. Damer, A. Sequeira and J. Cardoso, *Beyond masks: On the generalization of masked face recognition models to occluded face recognition*.

2 F. Boutros, P. Siebke, M. Klemt, N. Damer, F. Kirchbuchner and A. Kuijper, *PocketNet: Extreme lightweight face recognition network using neural architecture search and multistep knowledge distillation*.

3 A. George, C. Ecabert, H. Shahreza, K. Kotwal and S. Marcel, *EdgeFace: Efficient face recognition model for edge devices*, `https://arxiv.org/abs/2307.01838`, arXiv.

4 V. Munusamy and S. Senthilkumar, *Leveraging lightweight hybrid ensemble distillation (HED) for suspect identification with face recognition*.

5 J. Li, Y. Yang, G. Liu, Y. Ning and P. Song, *Open-Set Sheep Face Recognition in Multi-View Based on Li-SheepFaceNet*.

6 J. Dan, Y. Liu, J. Deng, H. Xie, S. Li, B. Sun and S. Luo, *TopoFR: A closer look at topology alignment on face recognition*, `https://arxiv.org/abs/2410.10587`, arXiv.

7 A. Zhalgas, B. Amirgaliyev and A. Sovet, *Robust Face Recognition Under Challenging Conditions: A Comprehensive Review of Deep Learning Methods and Challenges*.

8 T. Soukupov and J. Cech, *Real-time eye blink detection using facial landmarks*, Computer Vision Winter Workshop (CVWW).

9 Y. Taigman, M. Yang, M. Ranzato and L. Wolf, *DeepFace: Closing the gap to human-level performance in face verification*.

10 S. Serengil and A. Ozpinar, *LightFace: A hybrid deep face recognition framework*.

11 R. Pi, *Raspberry Pi 4 Model B specifications*, https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/.

12 E. Systems, https://www.espressif.com/sites/default/files/documentation/esp32, ESP32 series datasheet (Version 4.4).

Appendix A: AWS Lambda and FFmpeg Code Snippets

AWS Lambda Code Snippet

```
# AWS Lambda function to trigger FFmpeg conversion
import subprocess

def lambda_handler(event, context):
    if event['file_type'] == 'webm':
        convert_to_mp4(event['file_path'])

def convert_to_mp4(path):
    # Converts WEBM to MP4
    subprocess.run([
        'ffmpeg',
        '-i', path,
        path.replace('.webm', '.mp4')
    ])
```

FFmpeg Code Snippet

```
# FFmpeg command to extract 20 frames from MP4 video
ffmpeg -i input.mp4 -vf "fps=2" frame_%03d.jpg
```

GitHub Repository Link

https://github.com/Obamium37/Face-Data-Collecter