Resource-efficient face verification for edge devices using mmwave-triggered vision

Abstract. This work presents the implementation of a face verification system designed for edge hardware, where mmWave radar is used to trigger image capture and a compressed deep learning model performs local recognition. The system integrates a 24 GHz mmWave sensor for intelligent human presence detection, an ESP32-CAM module for image capture and preprocessing, and an ESP32-S3 microcontroller for real-time face embedding extraction and matching. A knowledge distillation framework compresses a MobileFaceNet 1.0× model into lightweight student variants enabling INT8 quantized inference fully on-device. Experimental results demonstrate sub-second recognition delay with minimal accuracy degradation, making the system suitable for smart home and IoT-based access control applications.

1 INTRODUCTION

Smart home automation and Internet of Things (IoT) security solutions have gained immense importance in contemporary living spaces. Traditional doorbell solutions usually involve passive infrared (PIR) sensors or basic motion detection systems that merely offer basic notifications without the need for intelligent identity verification [1]. These systems are prone to false alarms caused by changes in the environment and do not support efficient authentication. With the increasing need for privacy-preserving and real-time security solutions, there is an increasing demand for intelligent visitor identification systems that can work efficiently on resource-constrained hardware.

Recent developments in edge computing have made it possible to execute deep learning models on embedded hardware without relying on cloud infrastructure and with low latency [2], [3]. However, the execution of face recognition tasks on microcontrollers like ESP32-S3 is challenging due to memory limitations, computational complexity, and strict power constraints. Deep neural networks with high capacity need to be compressed using techniques like knowledge distillation and quantization to make them amenable to embedded hardware execution [4]. Recent TinyML based facial recognition systems have shown that compressed deep learning models can be deployed effectively on low power microcontrollers while maintaining competitive accuracy [5]. On the other hand, mmWave radar sensing has been recognized as a trustworthy substitute for PIR sensors by allowing precise human presence and micro-motion detection in diverse environmental settings [6].

In this paper, a hybrid edge-based face verification framework is proposed by combining mmWave-activated

sensing, distributed image preprocessing on ESP32-CAM and ESP32-S3 modules, and knowledge-distilled MobileFaceNet models with INT8 quantization. The proposed framework is able to ensure low latency, improved privacy, and efficient processing on embedded hardware.

2 RELATED WORK

The cloud-offloaded facial recognition improves embedding-based matching accuracy using optimized FaceNet embeddings for smart home access control. Nevertheless, their method still depends on cloud infrastructure, which is prone to latency and privacy concerns not ideal for decision-making applications.

Ahmad et al.[7] introduced LwFLNeT, a lightweight CNN-based model for real-time liveness detection on embedded IoT devices. The addition of parallel dropout layers increased spoofing detection accuracy and minimized model size, which is directly applicable to edge computing but not particularly for face verification in low-power devices.

Al-Hakeem et al.[8] introduced a multimodal anti-spoofing approach that integrates blink detection, depth information, and texture analysis for door security systems. Although it effectively suppressed false positives, their method is sensor-intensive and computationally expensive for edge devices.

Shah and Khan[9] introduced an AI-assisted smart door lock that utilized ESP-based hardware with cloud-based processing. Notably, they showed that cloud processing alleviates latency and improves accuracy—a result that justifies our edge-computing alternative to avoid cloud reliance.Knowledge distillation is widely used to reduce model size by transferring learned representations from a larger network to a smaller one. In this approach, a compact student model is trained to approximate the embedding outputs generated by a larger teacher model. This method successfully retains discriminative features while significantly reducing model size. It has been successfully applied to face recognition tasks by maintaining the embedding space geometry, which is

essential for identity matching.While prior work addresses individual components (mmWave sensing, face recognition, edge deployment, model compression), no existing solution systematically integrates all four—particularly, combining intelligent mmWave triggering with on-device quantized inference for zero-cloud-dependency face verification. This work fills that gap.

3 System Architecture

A.High-Level System Design

The proposed system operates as a three-stage pipeline:
(i)Sensing & Triggering: mmWave radar detects human presence and triggers downstream processing
(ii)Image Acquisition & Preprocessing: ESP32-CAM captures images, detects faces, and preprocesses regions of interest (ROI)
(iii)Inference & Decision: ESP32-S3 executes quantized face embedding inference, performs matching, and outputs recognition decisions

When a person approaches the door, the mmWave sensor detects motion and sends a GPIO trigger to the ESP32-CAM. The camera captures the image, performs face detection, and forwards the cropped face to the ESP32-S3, where embedding extraction and identity matching are carried out before generating an audio response.

The Waveshare Human Micro-Motion Detection mmWave Sensor (24 GHz, FMCW-based) device is used for face detection.The sensor uses Frequency Modulated Continuous Wave (FMCW) radar principles. Transmitted chirp signals are reflected by human targets, and the received frequency shift is analyzed to estimate:

i. Distance: Correlated with the reflection's time lag Doppler shift between successive chirps is used to detectmotion or micro-motion.

ii. Presence state: detection of motion, micromotion, or stationary

iii. Range gates: These allow for localization up to about 10 m by dividing the detection range into 16 distinct zones,each of which is about 0.7 m wide.

B. Benefits of PIR sensors:

Both motion and micro-motion (standing subjects) are detected. No erroneous triggers from airflow or temperature. Changes dependable outdoor performance [9].

4 MODEL COMPRESSION VIA KNOWLEDGE DISTILLATION

Full-capacity face recognition models (e.g., ResNet-50 for VGGFace2, approximately 100 MB) cannot fit in the ESP32-S3's 16 MB flash Knowledge distillation preserves discriminative feature learning while compressing models.

A. Dataset: VGGFace2 (approximately 3.14M training images, approximately 170k identities)[10]

B Preprocessing:

i.Input images resized to 112 × 112 pixels

ii.Pixel normalization: (x - 127.5) / 128

iii.Data augmentation:Random horizontal flipping

iv.Improves generalization to pose variations[10]

C Metric Loss (ArcFace/CosFace): The selected loss function reduces variation within the same identity while increasing separation between different identities in the embedding space[11]. Mathematically, the ArcFace loss is designed to minimize intra-class variance and maximize angular margins between distinct identities [11].

D Distillation Loss (L2/Cosine): This method uses feature-space alignment to transfer knowledge from teacher to student. The weighting of the combined metric and distillation losses is determined by the parameter alpha.

Teacher-Student Architecture:

ThE Table 1 compares the architectural specifications between the teacher and student networks used in the knowledge distillation process.

Table 1: Teacher-Student model specifications for knowledge distillation

Component	Teacher	Student
Architecture	MobileFaceNet 1.0×	MobileFaceNet 0.5× / 0.35×
Output Dimension	128-D embeddings	128-D embeddings
Parameters	~5.7M	~1.4M / ~0.6M
Status During Training	Frozen weights	Updated via gradient descent

The teacher network (MobileFaceNet 1.0×) serves as a high-capacity reference model with 5.7 million parameters, while the student networks (MobileFaceNet 0.5× and 0.35×) are progressively compressed versions containing 1.4 million and 0.6 million parameters respectively. Both networks produce 128-dimensional face embeddings, which are critical for identity matching.

During training, the teacher network weights remain frozen while the student network parameters are updated via gradient descent, allowing the student to learn from the teacher's learned representations without being influenced by subsequent changes to the student network.

In Distillation Dynamics, during training, only the student network parameters are updated. The frozen teacher provides soft supervisory signals through its embeddings, guiding the student to learn a compressed feature space that closely mimics the teacher's discriminative representations.

In the Model Optimization for Edge Constraints Quantization-Aware Training (QAT), the Student model

trained to simulate INT8 arithmetic. Robust inference on edge hardware with minimal accuracy loss. Bit-width: 8-bit (INT8) for weights and activations. Recent studies on low bit precision face recognition further confirm that quantized models can achieve efficient inference with minimal accuracy degradation on edge devices [12]. In the case of Pruning & Deployment, during inference, teacher network and classification layers removed. Retain only the lightweight embedding backbone. Final exported model: 128-D feature vector generator (approximately 1.2 MB). TensorFlow Lite Micro Conversion: Convert quantized model to TensorFlow Lite format (.tflite). Serialize as constant byte array in flash memory. Enable TensorFlow Lite Micro interpreter for on-device execution[13].

5 ON-DEVICE INFERENCE PIPELINE

A Image Acquisition and Preprocessing (ESP32-CAM)
Motion Trigger:

mmWave GPIO signal activates camera capture routine. Capture and Decode: Raw camera frame captured at resolution 320×240 or 160×120, decoded to RGB888 format for pixel-level processing. Face Detection: Lightweight face detector applied (e.g., MTCNN or TensorFlow Lite Face Detector), outputs bounding box coordinates[13]. Face Crop and Resize: Extract bounding box region, resize to fixed 112 × 112 pixels (model input requirement), normalize using (x - 127.5) / 128. Image Transmission: Re-encode cropped face to JPEG format, transmit via HTTP POST to ESP32-S3 PSRAM address, reduces bandwidth from approximately 260 KB (full frame) to approximately 8–12 KB (cropped face). Benefit: Offloading face detection to ESP32-CAM reserves ESP32-S3 resources for critical inference tasks[14].

B Embedding Extraction and Matching (ESP32-S3)
Memory Management:

Model weights are kept in the.rodata section of flash memory. The "Tensor Arena" is an external PSRAM that contains intermediate tensors and activation buffers. The typical PSRAM size is 2–8 MB, which is enough for a 5 MB model plus 3 MB of buffers .

(i) Execution of Inference: TensorFlow Lite Micro Interpreter starts with a quantized model. loads the input from the cropped face image that was received. uses registered operators to carry out INT8 inference.

(ii) Compact 128-D face embedding is the output. Identity matching involves comparing the extracted embedding to reference embeddings that have been stored.

(iii) Similarity metric: Cosine similarity is calculated by dividing Euclidean norms by the dot product. Results are contrasted with predetermined cutoff points.

(iv) High threshold (e.g., 0.65), multi-threshold decision logic:
(a) "Welcome [Name]" is output when similarity exceeds the threshold, indicating a confirmed match.
(b) Low threshold (e.g., 0.45): "Unknown visitor" is produced when the similarity is below the threshold, indicating no match.
(c) In between thresholds: Log for manual review, unclear classification.

(v) Latency Profile:
(a) mmWave detection to GPIO: roughly 10 ms.
(b) ESP32-CAM preprocessing and capture: about 200 ms.
(c) Transmission time via HTTP: roughly 50 ms.
(d) ESP32-S3 inference: INT8 quantized, about 80 ms.
(e) Decision-making and identity matching: roughly 5ms.
(f) The overall end-to-end latency is about 345 ms (sub-0.5 s).

(vi) DFPlayer Mini (MP3 player module, SD card storage) is the interface for the system output and audio feedback output. Procedure: ESP32-S3 uses UART (115200 baud) to send the recognition result to DFPlayer Mini.
Preloaded audio files on the SD card are mapped to audio commands:
(a) "Welcome [Authorized Person Name]" is the File 001.
(b) "Unknown visitor detected" in File 002.
(c) "Please confirm identity" in File 003.

DFPlayer: It plays the selected audio through the external speaker. After completion of the audio play, the system goes back to the idle state, waiting for the next trigger. The advantages of the system are as follows: There is immediate feedback without a visual interface. Communication with the household members is clear. Asynchronous UART prevents inference blocking.

6 EXPERIMENTAL SETUP AND RESULTS

A Training Setup

The model was trained with 90 percent training and 10 percent validation data from VGGFace2. A batch size of 128 and SGD with momentum 0.9 were used.

Table 2: Training hyperparameters and dataset configuration

Parameter	Value

Dataset	VGGFace2 (3.14M images, ~170k identities)
Train/Val Split	90% / 10%
Batch Size	128
Learning Rate	0.1 (initial), cosine annealing decay
Optimizer	SGD with momentum ($\mu = 0.9$)
Epochs	50
Distillation Temperature	4.0
Distillation Weight (α)	0.7

The learning rate started at 0.1 with cosine annealing. Distillation temperature 4.0 and weight 0.7 controlled the knowledge transfer from teacher to student.

B Model Compression Metrics

Table 3: Model compression effectiveness across multiple metrics

Metric	**Teacher**	**Student (0.5×)**	**Compression Ratio**
Parameters	5.7M	1.4M	4.07×
Model Size	22.8 MB	5.6 MB	4.07×
Inference Time (FP32)	145 ms	42 ms	3.45× faster
INT8 Inference Time	N/A	28 ms	Baseline + QAT

Table 3 describes the effectiveness across the metrics such as Knowledge distillation + INT8 quantization shrinks MobileFaceNet 0.5× student model 4.07× (22.8 MB → 5.6 MB) vs. teacher. Inference: 145 ms → 42 ms (FP32, 3.45× faster) → 28 ms (INT8, 5.2× total). Real-time on MCUs.

C Face Recognition Accuracy

The face verification accuracy of the teacher model (MobileFaceNet 1.0×, FP32), the distilled student model (FP32), and the quantized student model (INT8) on three benchmark datasets: LFW, CALFW, and CFP-FP, are shown in Table 4.

Table 4: Face recognition accuracy across standard evaluation protocols

Protocol	**Teacher (FP32)**	**Student (FP32)**	**Student (INT8)**
LFW	99.63%	99.41%	99.35%
CALFW	96.27%	95.89%	95.67%
CFP-FP	97.84%	97.22%	96.98%
Avg. Accuracy Loss	—	0.32%	0.58%

On the LFW benchmark, which is designed to test the accuracy under unconstrained real-world conditions, the teacher model achieves 99.63% accuracy, while the student model achieves 99.41% accuracy in FP32 and 99.35% accuracy in INT8 quantization, showing a total loss of only 0.28%. On the CALFW benchmark, which is more challenging due to the presence of age factors, the teacher model achieves 96.27%, while the student model achieves 95.89% accuracy in FP32 and 95.67% accuracy in INT8 quantization, showing a maximum loss of 0.60%.

On the CFP-FP benchmark, which tests the accuracy in cross-pose conditions, the teacher model achieves 97.84%, while the student model achieves 97.22% accuracy in FP32 and 96.98% accuracy in INT8 quantization, showing a total loss of only 0.32% on average for the distilled FP32 model and 0.58% for INT8 quantization.

The Fig.1 illustrates the sequence of operation carried by the visitor access control system.

Observations:

- Knowledge distillation preserves approximately 99% of teacher accuracy
- INT8 quantization introduces less than 1% additional accuracy loss
- Achieved sub-500ms end-to-end latency with less than 1% recognition degradation

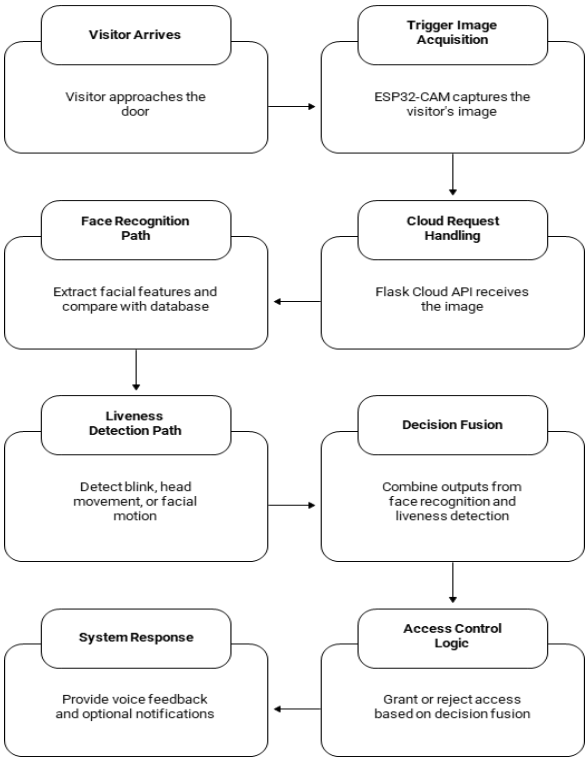

Figure 1 : Visitor Access Control System Sequence

7 RESULTS DISCUSSION

This paper has demonstrated a resource-effective face verification system developed for edge computing by combining mmWave-triggered sensing, distributed image preprocessing, and a knowledge-distilled MobileFaceNet model with INT8 quantization. The proposed system allows for fully edge-based inference on ESP32-based hardware, making it cloud-independent while still providing reliable recognition performance.

Experimental results showed a 4.07× model size reduction using knowledge distillation, with less than 0.6% average accuracy drop on LFW, CALFW, and CFP-FP datasets. The system provided around 345 ms end-to-end latency, which is much lower than that of cloud-assisted approaches, making it applicable for real-time access control. The mmWave radar sensor integration also enhanced triggering robustness over traditional PIR-based solutions by enabling micro-motion detection and suppressing false triggers.

In summary, the proposed solution strikes a good balance between model compression, efficiency, latency, and privacy preservation under the hardware limitations of embedded microcontrollers. These findings validate the feasibility of fully edge-based deployment for high-accuracy and privacy-preserving face verification systems. Future research directions include identity management, multimodal liveness detection, and improved performance under challenging lighting and environmental conditions.

Table 5: Comparative analysis of edge versus cloud-based face verification systems

Study	**Processing Type**	**Model Optimization**	**Latency**	**Privacy**	**Edge Deployment**	**Accuracy**
Patnaik & Verma [2]	Cloud-based	Deep embeddings (FaceNet)	500–2000 ms	Low (Cloud dependency)	No	99%+
Shah & Khan [5]	Cloud-assisted	Standard CNN	~800 ms	Low	Partial	98–99%
Ahmad et al. [3]	Edge-based	Lightweight CNN	~400 ms	Moderate	Yes	Not focused on verification
Embedded Detection [14]	Edge-based	Basic detection optimization	~300 ms	High	Yes	Detection only
Proposed System	**Fully Edge-based**	**Knowle-dge Distillation + INT8 QAT**	**~345 ms**	**High (Fully Offline)**	**Yes (ESP32-S3)**	**99%+ (0.58% loss)**

The following table highlights the benefits of the proposed system over the existing approaches. Unlike cloud-assisted systems [2], [5], which are susceptible to high latency due to network communication and lack privacy, the proposed system is completely offline with a response time of less than 350 ms. Compared to embedded models that are lightweight and target detection [14], the proposed system offers end-to-end face verification via deep embedding with knowledge-distilled compression and INT8 quantization, which can be achieved within tight microcontroller memory constraints. Furthermore, while the existing embedded systems are efficient, they do not support intelligent mmWave triggering and compressed deep embedding inference. The proposed system has been demonstrated to offer competitive recognition accuracy of 99%+ with low latency and end-to-end privacy preservation.

8 CONCLUSION

This project has demonstrated a real-world face verification system that can run entirely on edge hardware by integrating mmWave sensing with a compressed deep learning model. The face verification system is distributed across the ESP32 CAM and ESP32 S3 boards, with knowledge distillation and integer quantization to compress the model without affecting recognition accuracy. The experimental results demonstrate that the compressed student model retains over ninety-nine percent accuracy with an average degradation of less than zero point six percent compared to the full precision teacher model, with an overall response time of approximately three hundred forty-five milliseconds.

Compared to cloud-based smart door solutions, the proposed solution carries out all computations on edge hardware, which is more privacy-preserving and efficient. The experimental results have verified that accurate and efficient face verification can be achieved on microcontrollers without substantial performance degradation. Future research will be conducted to expand the identity database and enhance robustness to challenging environmental conditions.

Acknowledgment
The authors would like to thank SRM Valliammai Engineering College for providing lab facilities. Special thanks to the VGGFace2 dataset team for providing face recognition training data.

References

1. Shaout, M. Theisen, State of the art—Smart doorbell systems, in Proc. 22nd Int. Arab Conf. Inf. Technol. (ACIT), Muscat, Oman (2021) 1–8. https://doi.org/10.1109/ACIT53391.2021.9677313
2. K. Patnaik, A. Verma, Facial recognition for smart home access control using enhanced deep embeddings, IEEE Access 12, 1–15 (2024). https://doi.org/10.1109/ACCESS.2024.3347821

3. S. Ahmad, T. Bukhari, M. Hussain, Lightweight face liveness detection using LwFLNeT, J. Inf. Secur. Appl. 78, 103567 (2025). https://doi.org/10.1016/j.jisa.2025.103567
4. R. Al-Hakeem, L. Samer, P. Jordan, Multimodal anti-spoofing for smart door security systems, Sensors 24, 745 (2024). https://doi.org/10.3390/s24030745
5. N. Shah, M. Khan, AI-enabled smart door lock system with real-time cloud processing, Int. J. Intell. Syst. Appl. 17, 1–15 (2025). https://doi.org/10.5815/ijisa.2025.01.05
6. G. Hinton, O. Vanhoucke, J. Dean, Distilling the knowledge in a neural network, in Proc. NIPS 2014 Deep Learning Workshop (2014) https://doi.org/10.48550/arXiv.1503.02531
7. Q. Cao, L. Shen, W. Xie, O. M. Parkhi, A. Zisserman, VGGFace2: A dataset for recognising faces across pose and age, in Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. (FG), Xi'an, China (2018) 67–74. https://doi.org/10.1109/FG.2018.00020
8. Yahyati, I. Lamaakal, K. El Makkaoui, I. Ouahbi, Y. Maleh, TinyML based facial recognition for embedded systems, in Proc. Int. Conf. Circuit, Systems and Communication (2025). https://doi.org/10.1109/ICCSC66714.2025.11134957
9. J. Chen, J. Wang, P. Liu, mmWave radar for indoor presence detection, IEEE Trans. Aerosp. Electron. Syst. 56, 3852–3864 (2020) DOI:10.3390/rs16142572
10. Espressif Systems, ESP32-S3 technical reference manual, https://www.espressif.com/sites/default/files/documentation/esp32-s3_technical_reference_manual_en.pdf (accessed February 2026)
11. J. Deng, J. Guo, N. Xue, S. Zafeiriou, ArcFace: Additive angular margin loss for deep face recognition, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) (2019) 4690–4699. https://doi.org/10.1109/CVPR.2019.00482
12. W. Gazali, J. M. Kho, J. Santoso, Williem, Ef QuantFace: Streamlined face recognition with small data and low bit precision, arXiv:2402.18163 (2024) https://doi.org/10.48550/arXiv.2402.18163
13. TensorFlow Lite Micro, Microcontroller inference guide, https://www.tensorflow.org/lite/microcontrollers (accessed February 2026)
14. Y. Peng, Y. Li, M. Zhang, Real-time face detection on embedded systems, J. Embed. Syst. 14, 112–128 (2023) DOI:10.3390/s19092158