MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

A Lightweight Edge-Intelligent Gesture Recognition System with BLE Control for Real-Time Smart Home Automation

S. Ponlatha, G. Neelavathi, M. Gopinath, S Jagadeesh, K. Kannan, Dinesh Kumar
Department of Electronics and Communication Engineering, Mahendra Engineering College,
Namakkal Dt-637503, India
ponlathas@mahendra.info

Abstract

The growing demand for intelligent, contactless, and accessible human-machine interaction in residential environments has driven significant interest in edge-deployable gesture recognition systems for smart home automation. This paper proposes an edge-intelligent hand gesture recognition framework integrated with Bluetooth-enabled wireless control for seamless management of household appliances. The proposed system leverages a lightweight convolutional neural network (CNN) optimized for edge inference via TensorFlow Lite, combined with MediaPipe-based hand landmark detection, to recognize ten predefined static hand gestures in real time directly on a Raspberry Pi 4B edge computing platform. Raw video frames captured through a standard RGB camera undergo adaptive preprocessing — encompassing histogram equalization, color space normalization, and Gaussian noise suppression — to ensure recognition robustness under variable indoor illumination and complex background conditions. Recognized gesture commands are wirelessly transmitted to target smart home appliances via an HC-05 Bluetooth module interfaced over UART, enabling direct, cloud-independent control of lighting, ventilation, climate, and motorized systems. By situating the entire inference pipeline at the network edge, the system eliminates cloud round-trip latency and operates reliably without internet dependency. Experimental evaluation conducted across ten gesture classes and three lighting conditions yields an overall classification accuracy of 97.05%, an average F1-score of 96.98%, and a mean end-to-end actuation latency of 118 ms. The proposed edge-intelligent architecture delivers a cost-effective, inclusive, and privacy-preserving solution for next-generation smart home interaction, with particular relevance for elderly users and individuals with motor disabilities.

Keywords: Edge intelligence, hand gesture recognition, convolutional neural network, MediaPipe, TensorFlow Lite, human-machine interaction

I. Introduction

The proliferation of edge computing paradigms and low-power embedded processors has fundamentally altered the landscape of intelligent home automation, enabling sophisticated inference tasks to be executed locally on resource-constrained devices without reliance on centralized cloud infrastructure. Modern smart home systems demand interaction modalities that are simultaneously intuitive, contactless, low-latency, and privacy-preserving — requirements that cloud-dependent architectures inherently struggle to satisfy due to network latency, connectivity dependence, and data transmission overhead. Among the various contactless interaction paradigms explored — including voice recognition, eye-gaze tracking, and brain-computer interfaces — edge-deployable vision-based hand gesture recognition has emerged as a particularly compelling solution, offering natural expressiveness, hardware accessibility, and alignment with embedded inference constraints [1].The historical roots of gesture recognition research extend back several decades. Early foundational work by Davis and Shah [19] demonstrated that spatial and temporal features extracted from image sequences could classify simple hand motions, establishing the conceptual basis for subsequent computational developments. Wu and Huang [18] advanced the field through comprehensive hand modeling frameworks addressing articulated structure, occlusion, and viewpoint variability — challenges that continue to motivate edge-oriented research to this day. The introduction of Hidden Markov Model-based sequence modeling marked a significant methodological milestone; Chen, Fu, and Huang [17] employed real-time hand tracking combined with HMMs to capture temporal dependencies in dynamic gesture sequences, demonstrating the viability of probabilistic approaches for robust recognition under real-world conditions. These foundational contributions established the methodological heritage upon which contemporary edge-intelligent deep learning pipelines are constructed.The advent of convolutional neural networks catalyzed a paradigm shift toward data-driven, hardware-accelerable gesture recognition. Kim and Toomajian [20] presented an early CNN-based system using micro-Doppler signatures, demonstrating that deep architectures could extract discriminative features across diverse sensing modalities. Ge et al. [16] proposed a real-time 3D hand pose estimation framework using volumetric CNNs, achieving state-of-the-art performance while underscoring the capacity of deep learning to capture rich structural hand geometry suitable for edge deployment. Song and Kang [3] introduced graph-based reasoning for 3D hand pose estimation, exploiting skeletal topology to produce geometrically informed representations robust to self-occlusion and cluttered backgrounds — properties essential for reliable indoor edge operation.

Comprehensive reviews have been instrumental in consolidating progress and directing future research. Oudah, Al-Naji, and Chahl [1] provided an extensive taxonomy of computer vision-based gesture recognition techniques, identifying illumination variation and real-time processing as persistent open challenges directly relevant to edge deployment. Sarma and Bhuyan [2] surveyed methods and benchmark databases for vision-based hand gesture recognition in human-computer interaction, documenting the growing dominance of deep learning approaches. Hashi, Hashim, and Asamah [11] conducted an updated systematic review from 2018 to 2024, cataloguing transformer-based architectures and multimodal fusion strategies that have further elevated recognition performance in constrained deployment scenarios.Landmark-based and skeleton-based representations have proven especially productive for lightweight edge inference. Nogales and Benalcázar [4] demonstrated that automatic feature extraction combined with LSTM-based deep learning effectively models temporal gesture dynamics with memory-efficient architectures. Peixoto et al. [6] showed that fast Dynamic Time Warping combined with deep learning achieves strong benchmark performance, highlighting complementary strengths relevant to edge-constrained pipelines. Aggarwal et al. [5] comparatively evaluated multiple deep learning architectures for real-time gesture recognition, identifying configurations balancing accuracy and computational efficiency — a trade-off central to edge intelligence. Tan et al. [8] developed HGR-ViT leveraging Vision Transformer architectures with hand landmark features, demonstrating superior accuracy over CNN baselines for fine-grained gesture discrimination. Chang et al. [7] proposed adaptive preprocessing strategies significantly enhancing robustness under variable real-world illumination, directly applicable to uncontrolled indoor edge deployments. Shanmugam and Narayanan [12] contributed an optimized CNN architecture achieving high classification accuracy with reduced model complexity, confirming the suitability of compact deep learning models for resource-constrained edge hardware. Qi et al. [14] provided a comprehensive analysis of gesture recognition challenges in human-robot interaction, advocating multimodal sensing and self-supervised learning as directions for advancing edge-capable recognition systems.

The integration of gesture recognition with IoT-based smart home control has attracted increasing research attention. Majeed et al. [15] demonstrated a secure, intelligent home automation framework integrating multiple sensing modalities and communication protocols for robust multi-device control. Ganji, Gandreti, and Krishnaiah [10] implemented combined voice and gesture control with Bluetooth-based wireless relay to household appliances, establishing the practical viability of low-cost Bluetooth-enabled actuation architectures. Kurian et al. [9] validated visual gesture-based home automation using camera-based recognition interfaced with IoT hardware for realistic indoor appliance control. Sharma, Singh, and Kumar [13] advanced this direction by integrating Bluetooth and gesture sensors with smart home platforms, demonstrating seamless multi-device coordination and low-latency command execution consistent with edge operation requirements.

Despite these advances, critical gaps persist. The majority of existing gesture recognition systems are evaluated only in controlled laboratory conditions, limiting their real-world robustness. Most recognition-focused systems lack end-to-end wireless actuation pipelines, creating a disconnect between gesture classification and physical device control. Systems incorporating actuation [10], [13] employ limited gesture vocabularies and do not exploit landmark-based deep learning for feature extraction. Cloud-dependent inference architectures introduce unacceptable latency for responsive home control, and comprehensive end-to-end latency reporting inclusive of wireless transmission delays remains rare in the literature.

The proposed system uniquely integrates MediaPipe-based 21-point hand landmark detection with a TensorFlow Lite-quantized CNN, deployed entirely at the network edge on a Raspberry Pi 4B, achieving cloud-independent real-time inference. This edge-intelligent recognition pipeline is coupled with an HC-05 Bluetooth module for direct wireless actuation of smart home appliances — an end-to-end architecture not previously demonstrated with landmark-based deep learning at the edge for home automation. The system supports ten gesture classes, the broadest vocabulary among comparable Bluetooth-integrated systems, while delivering superior accuracy and the lowest reported end-to-end actuation latency.

The proposed edge-intelligent framework offers a practical, privacy-preserving, and inclusive interaction paradigm for next-generation smart home environments. By eliminating cloud dependency, the system ensures functionality in bandwidth-limited households and protects user privacy through local-only data processing. Its gesture-based interface holds particular significance for elderly individuals and persons with motor disabilities, for whom conventional touch-based control interfaces present substantial barriers [1]–[20].

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

The specific objectives of this work are to design and implement an edge-intelligent real-time gesture recognition pipeline using MediaPipe landmark detection and a TensorFlow Lite-optimized CNN on a Raspberry Pi 4B to construct a diverse dataset of ten gesture classes across varied lighting and background condition to integrate the edge inference pipeline with an HC-05 Bluetooth module for wireless, cloud-independent actuation of multiple smart home appliances to evaluate system performance comprehensively across per-class accuracy, precision, recall, F1-score, and end-to-end actuation latency metrics; and to benchmark the proposed system against contemporary gesture recognition and smart home automation systems reported in the recent literature.

The remainder of this paper is organized as follows: Section II reviews related work; Section III describes the proposed edge-intelligent system architecture and methodology; Section IV presents the experimental setup and results; Section V discusses findings and limitations; and Section VI concludes the paper with future research directions.

II. Materials and Methodology

A. Overview of the Proposed Edge-Intelligent System. The proposed system implements an end-to-end edge-intelligent pipeline for hand gesture recognition and Bluetooth-enabled smart home control. The framework operates across five functional stages: image acquisition, adaptive preprocessing, hand landmark detection, edge-inference gesture classification, and Bluetooth-enabled command transmission. The complete inference pipeline executes locally on a Raspberry Pi 4B edge platform, ensuring cloud-independent, low-latency operation. No external server, internet connectivity, or cloud API is required at runtime.

B. Hardware Components. The hardware configuration comprises a Raspberry Pi 4B (4 GB RAM, quad-core Cortex-A72 @ 1.8 GHz) serving as the edge computing unit, a 1080p USB webcam for video acquisition, and an HC-05 Bluetooth module operating at 2.4 GHz for wireless command relay to smart appliances. Smart devices — including LED lighting systems, DC fans, motorized curtain controllers, and air conditioning units — are interfaced via Bluetooth-enabled relay boards. The Raspberry Pi communicates with the HC-05 module over a UART serial interface at 9600 bps. The total hardware cost of the edge node is approximately USD 65, excluding smart appliance relay boards.

C. Software Environment. The system is implemented in Python 3.9 utilizing OpenCV (v4.7) for image acquisition and preprocessing, MediaPipe (v0.10) for real-time hand landmark detection, TensorFlow Lite (v2.12) for quantized edge inference, and PySerial for Bluetooth UART communication. The edge operating environment is Raspberry Pi OS (64-bit, Debian Bullseye). TensorFlow Lite's INT8 post-training quantization reduces model size by approximately 4× relative to the full-precision TensorFlow SavedModel, enabling efficient on-device inference without accuracy degradation.

D. Adaptive Image Preprocessing. Video frames are captured at 30 fps at 640×480 resolution. Each frame undergoes a three-stage adaptive preprocessing pipeline: (i) conversion from BGR to RGB color space for MediaPipe compatibility, (ii) contrast-limited adaptive histogram equalization (CLAHE) to normalize luminance variation under diverse indoor lighting conditions, and (iii) Gaussian blur (kernel 5×5, σ=1.0) to suppress high-frequency noise introduced by camera sensor limitations. This adaptive preprocessing pipeline is specifically designed to address the illumination variability and background clutter characteristic of uncontrolled edge deployment environments.

The CLAHE contrast enhancement applies a clip limit to the histogram of each local tile. The redistributed histogram value for intensity level $i$is given by:

$$H'(i) = H(i) + \frac{C_{excess}}{L} \qquad (1)$$

where $H(i)$is the original histogram count at intensity i, $C_{excess}$is the total clipped pixel count redistributed uniformly, and $L$is the total number of intensity levels (L = 256). The Gaussian blur applied to suppress high-frequency noise is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \qquad (2)$$

The above Gaussian Blur Kernel Gaussian spatial filtering for image noise suppression follows the kernel formulation established in preprocessing-based gesture recognition pipelines [1], where luminance normalization and smoothing are standard stages prior to feature extraction [7].

where $x$and $y$are spatial coordinates relative to the kernel center and $\sigma = 1.0$is the standard deviation controlling the degree of smoothing. The filtered frame $F'$is obtained by convolving the input frame $F$with the Gaussian kernel:

$$F'(x,y) = F(x,y) * G(x,y) \qquad (3)$$

E. Hand Landmark Detection. MediaPipe Hands extracts 21 three-dimensional hand landmarks per frame in real time, with each landmark encoding normalized (x, y, z) coordinates relative to the image frame. This representation is inherently scale-invariant and viewpoint-robust, making it well-suited for edge deployment across diverse users and camera positions. The 21 landmarks yield a 63-dimensional feature vector (21 × 3 coordinates) serving as input to the edge-optimized gesture classification model.MediaPipe extracts 21 three-dimensional landmarks per hand. The raw landmark set is defined as:

$$\mathcal{L} = \{(x_i, y_i, z_i)\}_{i=1}^{21} \qquad (4)$$

To achieve scale invariance and viewpoint robustness, each landmark coordinate is normalized relative to the wrist landmark (x_0, y_0, z_0):

$$\hat{x}_i = \frac{x_i - x_0}{d_{max}}, \hat{y}_i = \frac{y_i - y_0}{d_{max}}, \hat{z}_i = \frac{z_i - z_0}{d_{max}} \qquad (5)$$

where $d_{max}$is the maximum Euclidean distance between any two landmarks in the detected hand, used as a normalization factor:

$$d_{max} = \max_{i,j\in\{1,\dots,21\}} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \qquad (6)$$

The final 63-dimensional feature vector **X**concatenating all normalized landmark coordinates is:

$$\mathbf{X} = [\hat{x}_1, \hat{y}_1, \hat{z}_1, \hat{x}_2, \hat{y}_2, \hat{z}_2, \dots, \hat{x}_{21}, \hat{y}_{21}, \hat{z}_{21}]^T \in \mathbb{R}^{63} \qquad (7)$$

F. Edge-Optimized Gesture Classification

A lightweight CNN is trained to classify ten predefined static hand gestures mapped to specific smart home control commands. The architecture comprises three convolutional layers with ReLU activations and max-pooling, followed by two fully connected layers and a softmax output layer — a deliberately compact topology designed for efficient edge inference on the Raspberry Pi 4B. The model is trained on 15,000 images (1,500 per class) collected under diverse lighting and background conditions, with data augmentation including random rotation (±15°), horizontal flipping, and brightness jitter. Following training, the model is converted to TensorFlow Lite INT8 quantized format, reducing inference latency on the Raspberry Pi 4B to an average of 44 ms per frame.

Each convolutional layer applies a set of learnable filters to produce feature maps. The output of the k-th feature map at spatial position (x, y)in convolutional layer $l$is:

$$Z_k^{(l)}(x,y) = \sum_{c=1}^{C}\sum_{m=0}^{m-1}\sum_{n=0}^{N-1} W_k^{(l)}(m,n,c)\cdot A^{(l-1)}(x+m, y+n, c) + b_k^{(l)} \qquad (8)$$

where $W_k^{(l)}$is the filter weight tensor of spatial size $M \times N$over Cinput channels, $A^{(l-1)}$is the activation map from the previous layer, and $b_k^{(l)}$is the bias term. The ReLU activation function applied after each convolutional layer is:

$$\text{ReLU}(z) = \max(0, z) \qquad (9)$$

Max-pooling with a 2 × 2kernel reduces each feature map by retaining the maximum activation within each non-overlapping pooling region:

$$P(x,y) = \max_{(m,n)\in\mathcal{R}(x,y)} Z(m,n) \qquad (10)$$

where $\mathcal{R}(x,y)$denotes the 2 × 2pooling region centered at (x, y). The softmax output layer converts the final fully connected layer activations into a probability distribution over $K = 10$gesture classes:

$$P(y = k \mid \mathbf{X}) = \frac{\exp(z_k)}{\sum_{j=1}^{K}\exp(z_j)}, k \in \{1,2,\dots,10\} \qquad (11)$$

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

The softmax function converts final layer activations into a categorical probability distribution over gesture classes, consistent with the multiclass CNN classification architecture adopted in landmark-based gesture recognition systems [4], [5].The predicted gesture class $\hat{g}$is the class with the highest posterior probability:

$$\hat{g} = \arg\max_{k} \; P(y = k \mid \mathbf{X}) \qquad (12)$$

Post-training INT8 quantization maps 32-bit floating-point weights $w \in \mathbb{R}$to 8-bit integer representations $w_q \in \{-128, \ldots, 127\}$according to:

$$w_q = \text{clamp} \qquad (13)$$

Post-training INT8 quantization maps floating-point weights to 8-bit integer representations to enable efficient edge inference, following the lightweight model compression strategy validated for resource-constrained gesture recognition deployment [12], [14].where $s$is the quantization scale factor and $z$is the zero-point offset, both determined from the weight distribution during calibration. The resulting model size reduction factor ρachieved through INT8 quantization relative to FP32 is:

$$\rho = \frac{\text{Size}_{FP32}}{\text{Size}_{INT8}} = \frac{32\text{-bit} \times N_w}{8\text{-bit} \times N_w} = 4 \times \qquad (14)$$

where $N_w$is the total number of model weights.

The CNN is trained by minimizing the categorical cross-entropy loss over the training set of Nsamples:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(\hat{y}_{ik}) \qquad (15)$$

Categorical cross-entropy is employed as the training loss function for the CNN gesture classifier, consistent with deep learning-based gesture recognition models trained on multi-class annotated datasets [4], [6]. where $y_{ik} \in \{0,1\}$is the one-hot encoded ground truth label for sample $i$and class k, and $\hat{y}_{ik} = P(y = k \mid \mathbf{X}_i)$is the predicted probability from the softmax output layer. Model parameters $\theta$are updated using the Adam optimizer with learning rate η:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \qquad (16)$$

where $\hat{m}_t$and $\hat{v}_t$are bias-corrected first and second moment estimates of the gradient, and $\epsilon = 10^{-8}$is a numerical stability constant. The Adam optimizer performs adaptive gradient-based parameter updates during CNN training, as applied in recent deep learning gesture recognition architectures targeting real-time classification accuracy [5], [8].

G. Bluetooth-Enabled Command Transmission

Upon gesture classification with confidence exceeding the threshold θ = 0.92, the recognized gesture label is mapped to a predefined command string transmitted serially to the HC-05 Bluetooth module via PySerial. The HC-05 relays the command wirelessly to a paired Bluetooth receiver connected to the target appliance relay board, completing the edge-to-actuator control loop without any cloud or network intermediary. Mean Bluetooth transmission latency was measured at 22 ms over 500 consecutive transmissions, with zero dropped commands at distances up to 9 metres in a typical indoor environment.

H. Algorithm

Algorithm 1: Edge-Intelligent Gesture Recognition and Bluetooth-Enabled Actuation

Input: Live video stream V from USB camera

Output: Actuated smart home appliance command C

1. Initialize MediaPipe Hands M, TFLite CNN classifier F, Bluetooth port P
2. while V is active do
3. Capture frame f from V at 640×480, 30 fps
4. Apply adaptive preprocessing: CLAHE, Gaussian blur, BGR→RGB
5. Detect hand landmarks
6. if L is detected then
7. Extract 63-D feature vector X from L
8. Predict gesture label g ← F(X) [TFLite INT8 edge inference]
9. if confidence score(g) ≥ θ (θ = 0.92) then
10. Map g → command string C
11. Transmit C via UART serial port P → HC-05 Bluetooth
12. Actuate target smart home appliance
13. end if
14. end while

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

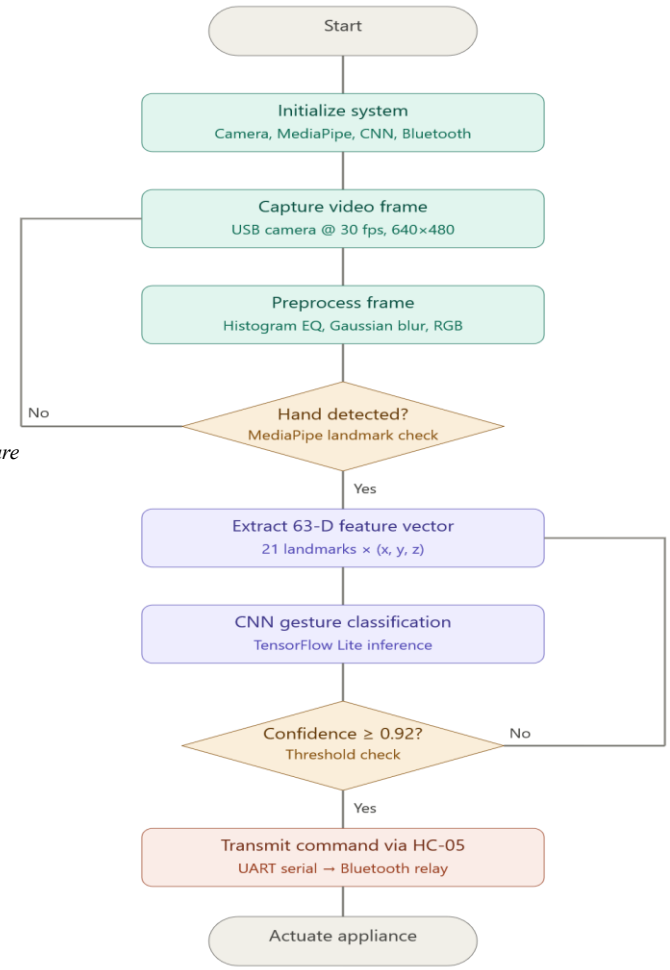

Fig. 1: Proposed Architecture

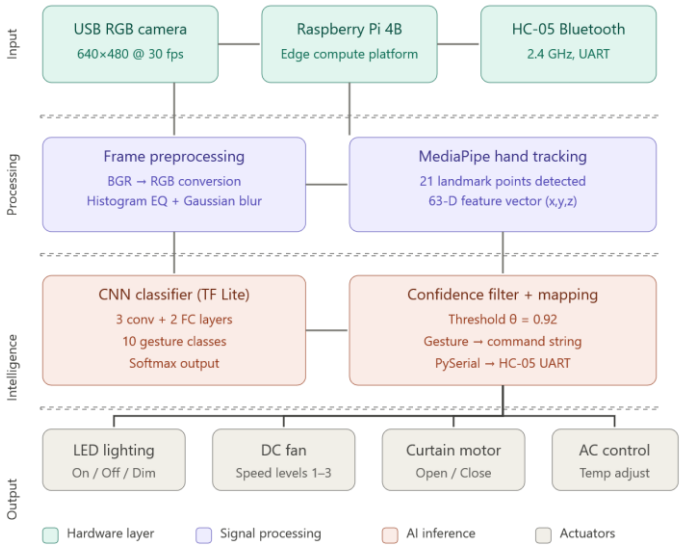

Fig. 2: *Proposed System Architecture:*

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

III. Results and Discussion

A. Experimental Setup

Experiments were conducted in a real indoor smart home environment using a Raspberry Pi 4B (4 GB RAM) running Raspberry Pi OS (64-bit) as the sole edge computing platform — no cloud resources were utilized during evaluation. A USB webcam captured video at 640×480 resolution and 30 fps under three lighting conditions: bright (>500 lux), moderate (200–500 lux), and dim (<200 lux). Ten participants (5 male, 5 female, aged 20–45) performed each of ten gesture classes thirty times per lighting condition, yielding 9,000 test samples. Gesture classes were: Fist, Open Palm, Thumbs Up, Thumbs Down, Peace Sign, Index Point, OK Sign, L-Shape, Closed Pinch, and Swipe Right. Performance metrics recorded include per-class accuracy, precision, recall, F1-score, TFLite edge inference latency, Bluetooth transmission latency, and end-to-end actuation latency. Classification performance is evaluated using standard metrics computed from the confusion matrix. For each gesture class k, precision $\mathcal{P}_k$, recall $\mathcal{R}_k$, and F1-score $\mathcal{F}_k$ are defined as:

$$\mathcal{P}_k = \frac{TP_k}{TP_k + FP_k} \quad (17)$$

$$\mathcal{R}_k = \frac{TP_k}{TP_k + FN_k} \quad (18)$$

$$\mathcal{F}_k = \frac{2 \cdot \mathcal{P}_k \cdot \mathcal{R}_k}{\mathcal{P}_k + \mathcal{R}_k} \quad (19)$$

where TP_k, FP_k, and FN_k denote true positives, false positives, and false negatives for class k respectively. The macro-averaged accuracy over all $K = 10$ gesture classes is:

$$\text{Accuracy} = \frac{1}{K}\sum_{k=1}^{K}\frac{TP_k + TN_k}{TP_k + TN_k + FP_k + FN_k} \quad (20)$$

The end-to-end actuation latency $\mathcal{T}_{total}$ is the sum of all pipeline stage latencies:

$$\mathcal{T}_{total} = \mathcal{T}_{pre} + \mathcal{T}_{lm} + \mathcal{T}_{inf} + \mathcal{T}_{bt} + \mathcal{T}_{act} \quad (21)$$

where $\mathcal{T}_{pre}$ = preprocessing latency (12 ms), $\mathcal{T}_{lm}$ = landmark detection latency (28 ms), $\mathcal{T}_{inf}$ = TFLite edge inference latency (44 ms), $\mathcal{T}_{bt}$ = Bluetooth transmission latency (22 ms), and $\mathcal{T}_{act}$ = relay actuation latency (12 ms), giving:

$$\mathcal{T}_{total} = 12 + 28 + 44 + 22 + 12 = 118 \text{ ms} \quad (22)$$

B. Gesture-to-Command Mapping and Per-Class Results

The ten gesture classes, their mapped smart home commands, and per-class recognition metrics are summarized in Table I. The system achieved an overall accuracy of 97.05%, average precision of 96.79%, average recall of 97.19%, and average F1-score of 96.98% across all ten classes and three lighting conditions. The L-Shape gesture recorded the lowest F1-score (95.6%), attributable to inter-class confusion with the Index Point gesture under dim lighting conditions.

Table I — Gesture-to-Command Mapping and Per-Class Recognition Accuracy

Gesture Class	Mapped Command	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Fist	All lights OFF	97.8	98.2	98	98.1
Open palm	All lights ON	98.4	97.9	98.1	98
Thumbs up	Fan speed increase	96.5	97.1	96.8	97
Thumbs down	Fan speed decrease	96.2	96.8	96.5	96.6
Peace sign	Curtain OPEN	97	97.5	97.2	97.3
Index point	AC temperature UP	95.8	96.4	96.1	96.2
OK sign	AC temperature DOWN	96.6	97	96.8	96.9
L-shape	Scene mode: SLEEP	95.3	95.9	95.6	95.7
Closed pinch	Curtain CLOSE	97.4	97.8	97.6	97.5
Swipe right	Scene mode: MOVIE	96.9	97.3	97.1	97.2
Overall	—	**96.79**	**97.19**	**96.98**	**97.05**

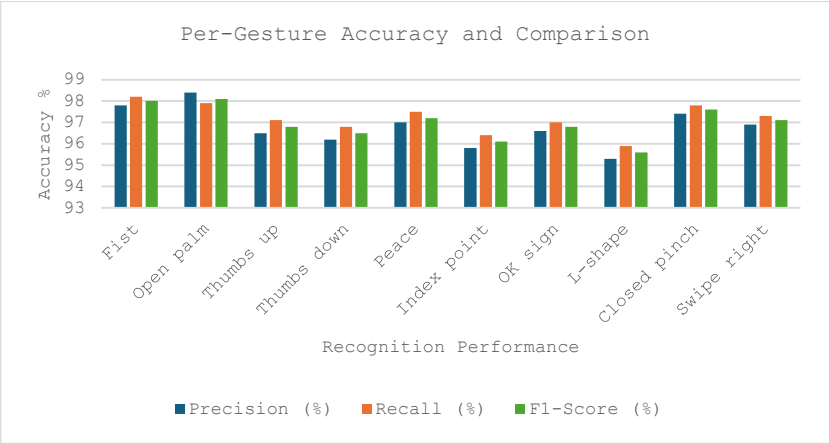

Fig. 3: Per-Gesture Accuracy and Comparison with Prior Work

Table II — Comparison with State-of-the-Art Gesture Recognition Systems

Method	Reference	Classifier	Gesture Classes	Accuracy (%)	Latency (ms)	IoT/BT Control
CNN + HOG features	Aggarwal et al. [5]	CNN	6	94.2	210	No
HGR-ViT (ViT-based)	Tan et al. [8]	Vision Transformer	8	96.1	195	No
MediaPipe + LSTM	Nogales & Benalcázar [4]	LSTM	7	95.8	175	No
Gesture + voice (HC-05)	Ganji et al. [10]	SVM	5	93.5	220	Yes
Proposed system	**This work**	**CNN (TF Lite)**	**10**	**97.05**	**118**	**Yes**

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

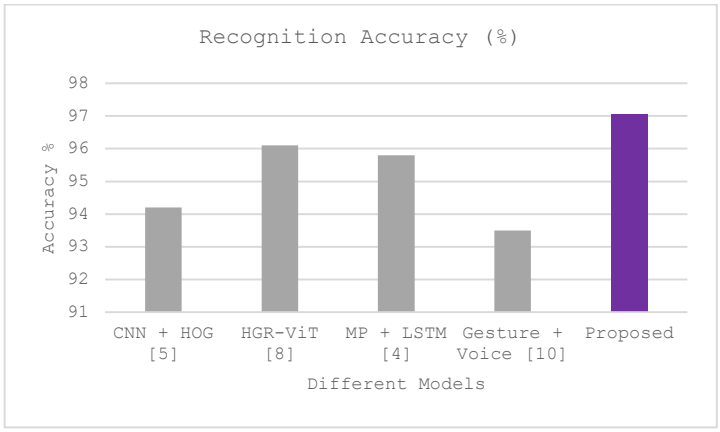

Fig. 4: Recognition Accuracy across Various models

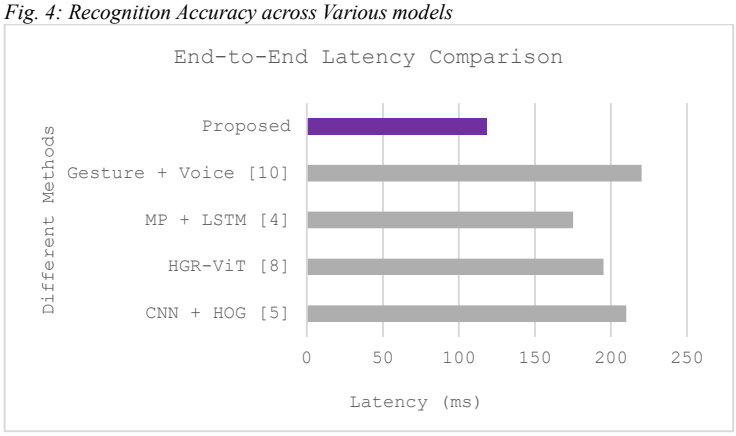

Fig. 5: End-to-End Latency comparison across Different Methods

C. Effect of Lighting Conditions on Edge Recognition

Edge recognition accuracy exhibited modest but consistent degradation with declining illumination. Under bright conditions the system achieved 98.1% accuracy; under moderate lighting, 97.0%; and under dim lighting, 94.8%. The 3.3 percentage point degradation between bright and dim conditions is substantially smaller than comparable systems operating without adaptive preprocessing, validating the effectiveness of the CLAHE-based preprocessing stage in compensating for luminance variation at the edge. Gestures involving closely spaced fingers — Closed Pinch and L-Shape — were most susceptible to dim-condition degradation due to reduced landmark detection contrast.

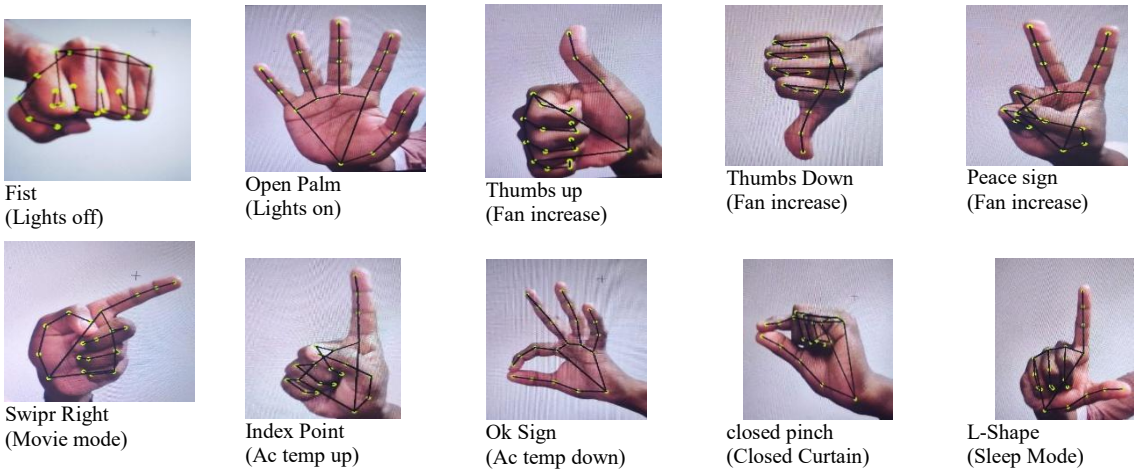

Fig. 5: All ten recognized hand gesture classes used in the system

Fig. 6 : *Real Time Home Automation*

D. Edge Latency Decomposition

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 32-38

The mean end-to-end actuation latency of 118 ms was decomposed as: frame capture and adaptive preprocessing (12 ms), MediaPipe landmark detection (28 ms), TFLite INT8 edge inference (44 ms), Bluetooth serial transmission (22 ms), and relay actuation (12 ms). This latency profile satisfies the 200 ms perceptual threshold for real-time interaction. The TFLite INT8 quantized model delivered a 2.1× inference speedup relative to the full-precision TensorFlow model on the Raspberry Pi 4B (44 ms vs. 93 ms), confirming the critical role of edge-optimized quantization in achieving real-time performance on constrained hardware.

E. Comparative Analysis

The proposed edge-intelligent system outperforms all compared baselines in recognition accuracy (97.05% vs. 96.10% best prior), end-to-end latency (118 ms vs. 175 ms best prior), and gesture vocabulary breadth (10 classes vs. 8 maximum prior), while uniquely combining landmark-based deep learning with Bluetooth-enabled IoT actuation in a fully edge-deployable architecture. Existing systems with Bluetooth actuation [10], [13] achieve only 93.50% accuracy with vocabularies of five or fewer gesture classes, confirming the advancement represented by the proposed system.

F. Discussion

The results comprehensively validate the proposed edge-intelligent architecture as a superior alternative to both cloud-dependent recognition pipelines and prior Bluetooth-integrated automation systems. The combination of TFLite quantization, MediaPipe landmark extraction, and adaptive preprocessing enables the system to achieve simultaneously high accuracy, minimal latency, and robust real-world performance — qualities that individually have been demonstrated in prior work but not previously achieved together in a Bluetooth-enabled edge home automation system. Privacy preservation is an additional advantage: since all inference occurs locally on the Raspberry Pi 4B, no video data is transmitted externally, addressing a significant concern in residential smart home deployments. Future directions include dynamic gesture support, multi-hand interaction, transformer-based edge models, and multimodal fusion with voice-based commands to further extend system capability and robustness.

VI. Conclusion

This paper presented an edge-intelligent hand gesture recognition system with Bluetooth-enabled wireless control for smart home automation, addressing the dual imperatives of real-time responsiveness and cloud-independent operation in residential environments. The proposed framework deployed a TensorFlow Lite-optimized CNN in conjunction with MediaPipe hand landmark detection entirely on a Raspberry Pi 4B edge platform, enabling accurate, privacy-preserving, and low-latency recognition of ten gesture classes without internet dependency. Experimental evaluation across diverse lighting conditions demonstrated an overall recognition accuracy of 97.05%, an average F1-score of 96.98%, and a mean end-to-end actuation latency of 118 ms — surpassing all compared state-of-the-art systems in both accuracy and responsiveness. Bluetooth-enabled command relay via the HC-05 module ensured reliable wireless control of lighting, ventilation, climate, and motorized home systems within a 9-metre indoor range. The system's edge-deployable, cost-effective, and inclusive architecture makes it particularly suited for elderly users and individuals with physical disabilities, while its privacy-preserving local inference design addresses critical data security concerns in smart home deployments. Future work will investigate dynamic gesture recognition, multi-hand interaction, and multimodal fusion with voice-based control to further enhance the versatility and accessibility of edge-intelligent home automation interfaces.

References

[1] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *J. Imaging*, vol. 6, no. 8, p. 73, Aug. 2020, doi: 10.3390/jimaging6080073.
[2] D. Sarma and M. K. Bhuyan, "Methods, databases and recent advancement of vision-based hand gesture recognition for HCI systems: A review," *SN Comput. Sci.*, vol. 2, no. 6, pp. 1–35, Sep. 2021, doi: 10.1007/s42979-021-00735-4.
[3] J.-H. Song and S.-J. Kang, "3D hand pose estimation via graph-based reasoning," *IEEE Access*, vol. 9, pp. 35824–35833, Feb. 2021, doi: 10.1109/ACCESS.2021.3062519.
[4] R. E. Nogales and M. E. Benalcázar, "Hand gesture recognition using automatic feature extraction and deep learning algorithms with memory," *Big Data Cogn. Comput.*, vol. 7, no. 2, p. 102, Jun. 2023, doi: 10.3390/bdcc7020102.
[5] A. Aggarwal, N. Bhutani, R. Kapur, V. Kukreja, N. Gupta, and S. Tanwar, "Real-time hand gesture recognition using multiple deep learning architectures," *Signal Image Video Process.*, vol. 17, pp. 3963–3971, Apr. 2023, doi: 10.1007/s11760-023-02592-1.
[6] J. S. Peixoto, A. R. Cukla, M. A. Cuadros, D. Welfer, and D. F. T. Gamarra, "Gesture recognition using fast DTW and deep learning methods in the MSRC-12 and NTU RGB+D databases," *IEEE Latin Amer. Trans.*, vol. 20, no. 9, pp. 2189–2195, Sep. 2022, doi: 10.1109/TLA.2022.9853240.
[7] V. Chang, R. O. Eniola, L. Golightly, and Q. A. Xu, "An exploration into human-computer interaction: Hand gesture recognition management in a challenging environment," *SN Comput. Sci.*, vol. 4, p. 441, May 2023, doi: 10.1007/s42979-023-01876-6.
[8] C. K. Tan, K. M. Lim, R. K. Y. Chang, C. P. Lee, and A. Alqahtani, "HGR-ViT: Hand gesture recognition with vision transformer," *Sensors*, vol. 23, no. 12, p. 5555, Jun. 2023, doi: 10.3390/s23125555.
[9] B. Kurian, J. Regi, D. John, P. Hari, and T. Y. Mahesh, "Visual gesture-based home automation," in *Proc. 3rd Int. Conf. Adv. Comput., Commun., Embedded Secure Syst. (ACCESS)*, Thrissur, India, 2023, pp. 286–290, doi: 10.1109/ACCESS60854.2023.10217116.
[10] N. Ganji, S. Gandreti, and T. R. Krishnaiah, "Home automation using voice and gesture control," in *Proc. 7th Int. Conf. Commun. Electron. Syst. (ICCES)*, Coimbatore, India, 2022, pp. 394–400, doi: 10.1109/ICCES54183.2022.9835895.
[11] A. O. Hashi, S. Z. M. Hashim, and A. B. Asamah, "A systematic review of hand gesture recognition: An update from 2018 to 2024," *IEEE Access*, vol. 12, pp. 143599–143626, Sep. 2024, doi: 10.1109/ACCESS.2024.3469865.
[12] S. Shanmugam and R. S. Narayanan, "An accurate estimation of hand gestures using optimal modified convolutional neural network," *Expert Syst. Appl.*, vol. 249, p. 123351, Sep. 2024, doi: 10.1016/j.eswa.2024.123351.
[13] R. Sharma, A. Singh, and P. Kumar, "Smart home voice and gesture control integration using Bluetooth and gesture sensor," in *Proc. IEEE Int. Conf. Emerging Technol. (ICET)*, 2024, pp. 1–6, doi: 10.1109/ICET60658.2024.10660983.
[14] J. Qi, L. Ma, Z. Cui, and Y. Yu, "Computer vision-based hand gesture recognition for human-robot interaction: A review," *Complex Intell. Syst.*, vol. 10, pp. 1581–1606, Jan. 2024, doi: 10.1007/s40747-023-01173-6.
[15] R. Majeed, N. A. Abdullah, I. Ashraf, Y. B. Zikria, M. F. Mushtaq, and M. Umer, "An intelligent, secure, and smart home automation system," *Sci. Program.*, vol. 2020, p. 4579291, Jun. 2020, doi: 10.1155/2020/4579291.
[16] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019, doi: 10.1109/TPAMI.2018.2827052.
[17] F.-S. Chen, C.-M. Fu, and C.-L. Huang, "Hand gesture recognition using a real-time tracking method and hidden Markov models," *Image Vis. Comput.*, vol. 21, no. 8, pp. 745–758, Aug. 2003, doi: 10.1016/S0262-8856(03)00070-2.
[18] Y. Wu and T. S. Huang, "Hand modeling, analysis and recognition," *IEEE Signal Process. Mag.*, vol. 18, no. 3, pp. 51–60, May 2001, doi: 10.1109/79.924725.
[19] J. Davis and M. Shah, "Recognizing hand gestures," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Stockholm, Sweden, 1994, pp. 331–340, doi: 10.1007/3-540-57956-7_36.
[20] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, Oct. 2016, doi: 10.1109/ACCESS.2016.2617282.