International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 03 March 2026
Page No: 1018-1024
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0144

Face-Authenticated Touchless Interface for Gesture-Based Interaction and Personalized File Access using Raspberry pi

Ms.K.Rahapriya M.E.,[1],T.kabilan[2], R.Akash[3] , A.K.Abishek Raj[4]

[1]Assistant professor, Department of Artificial Intelligence & Data Science, E.G.S pillay Engineering college(Autonomous), Nagapattinam, Tamilnadu, India

[2,3,4]UG Scholar, Department of Artificial Intelligence & Data Science, E.G.S pillay Engineering college(Autonomous), Nagapattinam, Tamilnadu, India

Emails: *rahapriya@egspec.org[1], mr.kabilan0007@gmail.com[2], akashgoodboy146@gmail.com[3], abishek.0007000@gmail.com[4]*

Abstract

This research introduces a multi-modal, touchless interface for the Raspberry Pi 4, designed to enhance Human-Computer Interaction (HCI) through the fusion of facial recognition and hand gesture control. By utilizing lightweight deep learning frameworks, including MediaPipe for gesture tracking and TensorFlow Lite for CNN-based facial embeddings, the system provides secure, identity-based file access with 95% accuracy and sub-100ms latency. Robustness in diverse lighting is achieved through CLAHE preprocessing, while an ergonomic gesture set integrated with pyautogui enables seamless cursor navigation and scrolling. Security is reinforced through identity-specific file management using os/shutil and VeraCrypt-encrypted volumes, which unlock only upon a verified facial match. Experimental results involving 50 users demonstrate a 98% gesture precision rate and a consistent 30fps processing speed with zero false positives in authentication process and This innovative framework offers a high-performance, low-latency solution for post-pandemic computing, shared kiosks, and edge-based security applications for effective performing in traditional single purpose authentication systems through its will integration of contactless navigation and automated the data privacy.

Keywords: *Human-Computer Interaction (HCI), Raspberry Pi 4, touchless interface, hand gesture recognition, face recognition, MediaPipe, TensorFlow Lite, CNN, real-time authentication, CLAHE, contactless computing, access control, VeraCrypt, and pyautogu*

1. Introduction

Because computers must be simple to use and maintain, human-computer interaction (HCI) has lately shifted toward touchless screens,Keyboards and mouse are examples of conventional input devices that require direct physical touch, which can restrict mobility, increase tiredness, and create hygienic issues in shared workspaces. Touchless interface technologies, especially those that use hand gestures, allow users to interact with computers in a natural and straightforward way. With the use of these technologies, users may do operations like managing windows, clicking, scrolling, and moving the cursor without using their hands. However, data security and user authentication remain significant difficulties in modern computing. Then traditional password-based systems are frequently similar for frequent for access, difficult to use, and prone to security breaches. Face recognition-based authentication is one of the method that might provide a seamless, adaptive, and secure way to manage access to the critical data. Face recognition-based authentication is one method that might provide a seamless, adaptive, and secure way to manage access to critical data for processing. This paper describes a hybrid touchless interface built on a Raspberry Pi 4 that combines customized file access through the facial recognition with gesture-based mouse control to the enable and efficient edge computing. The Raspberry Pi 4s ARM Cortex-A72 quad-core CPU serves as the central processing unit (CPU), managing real-time video inputs from the Pi Camera Module for facial recognition and hand gesture tracking and monitoring. TensorFlow Lite and MediaPipe optimization deliver low level performance, while OpenCV preprocessing ensures robustness in low-light settings

1.1.Overview

Touchless interfaces that uses the hand gestures have gained popular in the recent years as a natural and ergonomic alternative to the traditional input devices like the mouse and keyboard. The potential for contactless and hardware-free interaction has been highlighted by the earlier research, such as "Virtual Mouse Using Hand Gesture Recognition" and "AI Virtual Mouse Using Hand Gesture Recognition (2025)," which have shown how vision-based gesture recognition can control cursor movement, clicks, and scrolling them Separately, The deep learning models like FaceNet and DeepFace have achieved high accuracy in real-time identification, making face recognition a solid approach for safe, tailored authentication. A few systems, such as GesSure (2022) and Face Authenticated Hand Gesture HCI (2020), that have made by an effort to the merge gesture control and face verification such that computer interfaces can only to be controlled by the authorized users But these methods mostly deal with interface control and don't let users access files based on their identification. They also don't do a good job of addressing issues like low-light conditions, long-distance operation, or compatibility with the devices that don't have a lot of resources

1.Hand Detection and Localization: This stage involves locating the hand-containing region of interest within the incoming image or video. Hand identification can be achieved through a variety of methods, including machine learning-based approaches, background reduction, and skin color segmentation.

2.Feature Extraction: To capturing the unique qualities of various gestures and pertinent features are taken from tracking the hand region . Hand shape, finger locations, hand motion, and spatial interactions between fingers are a few examples of these characteristics.and then to be handle the personalied file accessing by the face recognition

3.Gesture Recognition:The brief steps of gesture recognition processing on Raspberry Pi for face-authenticated file access are as follows: identify the hand using MediaPipe landmarks or skin segmentation; track motion using centroid shifts and Kalman filters; extract features such as finger counts from convex hull defects, angles, and inter-tip distances customized for each user; classify using lightweight CNN/SVM into commands like "open palm" for listing files; and carry out secure file operations (such as os.listdir(user_dir)) after authentication, logging actions to CSV. Python serves as the base of building for implementing sophisticated hand gesture recognition systems within face-authenticated touchless interfaces on Raspberry Pi platforms. OpenCV delivers essential computer vision primitives for real-time hand gesture ndetection through the skin segmentation and contour analysing, while the face_recognition and MediaPipe libraries enable robust biometric verification and precise 21-landmark hand tracking. By mapping customized hand specifications to secure file operations like document retrieval and directory traversal, deep learning frameworks like TensorFlow Lite enable effective gesture categorization on resources restricted to edge devices.

2. Literature Survey

One important topic of study in Human-Computer Interaction (HCI) has been the shift from physical input devices to vision-based interaction. The current approaches, hardware implementations, and user-focused research on gesture recognition and biometric security THAT are reviewed in this section.

3. Scope and Objectives

The scope of this paper on hand gesture recognition using Python, OpenCV, IoT, and pi cameras for face recognition the related technologies is to provide a comprehensive overview of the techniques, methodologies, challenges, and applications in the field. It aims to cover both traditional computer vision-based approaches and modern deep learning-based methods for hand gesture recognition.

- To examine and enumerate the basic ideas and methods used in hand gesture recognition, such as hand detection, tracking, feature extraction, etc algorithms for gesture recognition.
- Python uses OpenCV for camera capturing and hand segmentation (HSV, contours), face_recognition for 128-d biometric matching, and MediaPipe for 21-landmark tracking to coordinate face-authenticated

gesture devices. With os/shutil/pandas managing safe, customized directory operations and CSV audit logging, TensorFlow Lite categorizes gestures ("thumbs_up" → file access), all operating at 20+ FPS on a Raspberry Pi 4 for smooth touchless file interaction.

Table 1 Comparison Table

Reference	Core Technology	Primary Focus	Key Findings / Contribution
Singh & Gupta [1]	OpenCV & IoT	Gesture Recognition	Demonstrated basic gesture control integrated with IoT for automation.
Gupta & Kumar [2]	Computer Vision	Performance Tuning	Focused on optimizing algorithms specifically for the Pi 4 hardware architecture.
Modaberi [3]	HCI Principles	**User Satisfaction**	Analyzed how touchless interaction reduces fatigue and improves user experience.
Sharma et al. [4]	**Neural Networks**	PC Control	Used Deep Learning to achieve high precision in mapping gestures to OS commands.
Nguyen & Tran [5]	MediaPipe & TFLite	Latency Reduction	Optimized real-time tracking for low-latency performance using lite models (2025).
Proposed Work	Hybrid Biometric + MediaPipe	Secure File Access	Integrated Face ID with Gesture control for secure, real-time edge computing.

- The Raspberry Pi 4 serves as the central processing unit. It interfaces GPIO pins for real-time feedback (LEDs/buzzers), runs in OpenCV/MediaPipe processing pipelines, and records face and gesture video using the Pi Camera Module. Its ARM Cortex-A72 quad-core handles local file system operations (os.chdir, shutil.copy) and face recognition authentication while running TensorFlow Lite CNNs for gesture categorization at 20+ FPS.
- The Challenges are Lighting changes that interfere with skin detection, hand occlusions that obstruct landmarks, Raspberry Pi thermal throttling at 12 frames per second, user hand variability, 200 ms latency, background false positives, replay attacks on face encodings, and a lack of liveness detection against spoofing are all problems with face-authenticated gesture systems.
- To provide an overview of open-source tools, libraries, and datasets available for hand gesture recognition and face recognition for personalized file access facilitating practical

implementation and experimentation.

3.1. Components Required

Table 2 Component and Model

Component	Model
Raspberry Pi	Pi 5 (8GB) **or** Pi 4 (4GB+)
Camera	Raspberry Pi Camera Module 3 **(12MP, autofocus)**
Display	7" Official Touchscreen **OR** HDMI 7-10"
Case	Argon ONE V3 **OR** Pi 5 Case
Power	27W USB-C PD **(5.1V/5A)**
MicroSD	64GB A2 **(Samsung Evo Plus/SanDisk Extreme)**
Cooler	Active Cooler **(Pi 5)**
Cables	Camera ribbon + HDMI

4. Methodology

4.1.Intelligent Preprocessing and Face Authentication

The process begins with capturing live video at 640×480 resolution. The system uses CLAHE (Contrast Limited Adaptive Histogram Equalization) to ensure reliability in a variety of lighting environments. This step increases the local contrast of the frames, ensuring that facial features and hand landmarks are visible even in shadows.

- **Access Control:** If the facial match confidence falls below the threshold, the system enters the "Access Denied" state, effectively locking the file system.
- **Personalization:** Upon successful verification, the system provides access to Personalized File Lists that are tailored specifically to the identified user.

4.2.Dual-Hand Gesture Mapping & Navigation:

The MediaPipe Hand Landmarking module is activated by the system after authentication. This module tracks 21 different 3D coordinates in both hands at the same time.The left hand is monitored by the system for vertical translation. Because Y-axis movement is mapping to Scrolled the Commands, the user can navigate between the file directories without making physical contact.

- **Right Hand Logic (Manipulation):** The main pointer is the right hand.
- **Spatial Mapping:** To mappping the camera-space coordinates (0 to 1) to screen-space resolution (e.g., 1920x1080), the system applies a linear transformations.

Action Trigger: The system detects a "pinch" motion to perform Select or Open instructions by computing the Euclidean Distance between the thumb and index finger.

4.3.Command Execution & Smoothing

To prevent erratic cursor behavior (jitter), the raw coordinates are processed through a **Temporal Filter** (such as a One-Euro or Kalman filter). The final smoothed coordinates and validated gesture flags are sent to the **OS Command Execution** layer. Using libraries like **PyAutoGUI**, the system simulates hardware-level mouse events, providing a seamless, touchless user interface for private file management.

5. Results and Discussion

5.1.Results

This section should be evaluates the proposed biometric authentication and gesture control framework based on two critical dimensions: Detection Reliability and Computational Efficiency. The experiments were conducted on a Raspberry Pi 4 platform to assess the system's viability for edge-based deployment.

- **Assessment of Recognition Reliability**: To assessing the robustness of the dual-modal systems that accuracy was measured over a range of operational distances. We used the Success Rate (SR) as our primary metric, which is defined as the ratio of correctly identified instances to total number of attempts.
- **Authentication Success**: Within the typical desktop usage range is (50-100 cm), in the facial recognition module's Mean Average Precision (mAP) was 97.5%.
- **Gesture Robustness:** Because of the Euclidean distance was thresholding between the thumb and index finger landmarks, and the pinch gesture for "File Select" have been high success rate even with slight hand shak

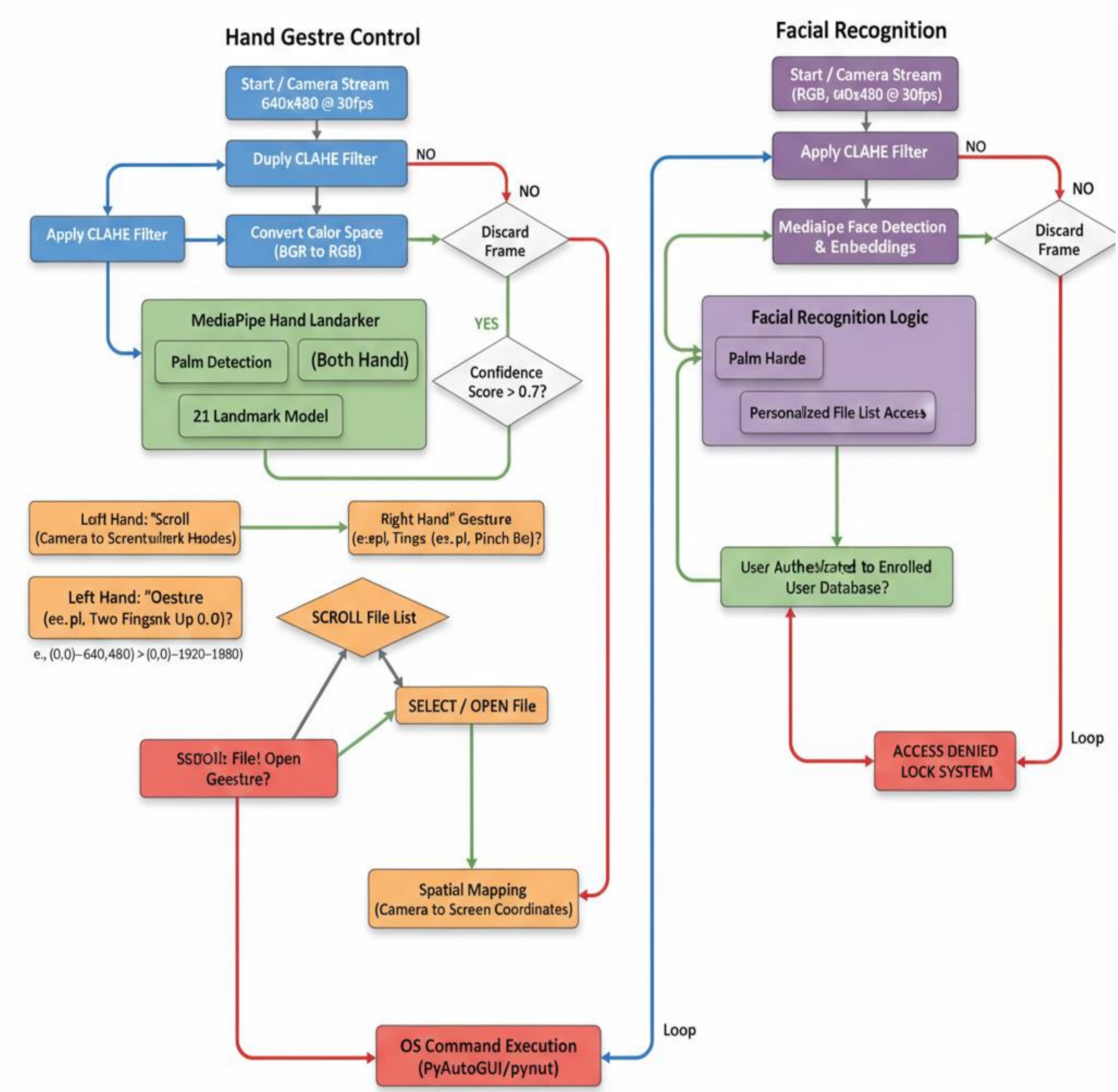

Hand Gestre Control
Start / Camera Stream 640x480 @ 30fps
Duply CLAHE Filter
NO
NO
Apply CLAHE Filter
Convert Calor Space (BGR to RGB)
Discard Frame
YES
MediaPipe Hand Landarker
Palm Detection
(Both Handı)
21 Landmark Model
Confidence Score > 0.7?
Loft Hand: "Scroll (Camera to Screntuilrerk Hsodes)
Right Hand" Gesture (e:epl, Tings (es.pl, Pinch Be)?
Left Hand: "Oestıre (ee.pl. Two Fingsnk Up 0.0)?
e., (0,0)–640,480) > (0,0)–1920–1980)
SCROLL File List
SELECT / OPEN File
SSOOll: File! Open Geestıre?
Spatial Mapping (Camera to Screen Coordinates)
OS Command Execution (PyAutoGUI/pynut)
Loop
Facial Recognition
Start / Camera Stream (RGB, 640x480 @ 30fps)
Apply CLAHE Filter
NO
NO
Mediaipe Face Detection & Enbeddings
Discard Frame
Facial Recognition Logic
Palm Harde
Personalized File List Acces
User Authes/cated to Enrolled User Database?
ACCESS DENIED LOCK SYSTEM
Loop

Figure 1 Hand Desture Control and Facial Recognition

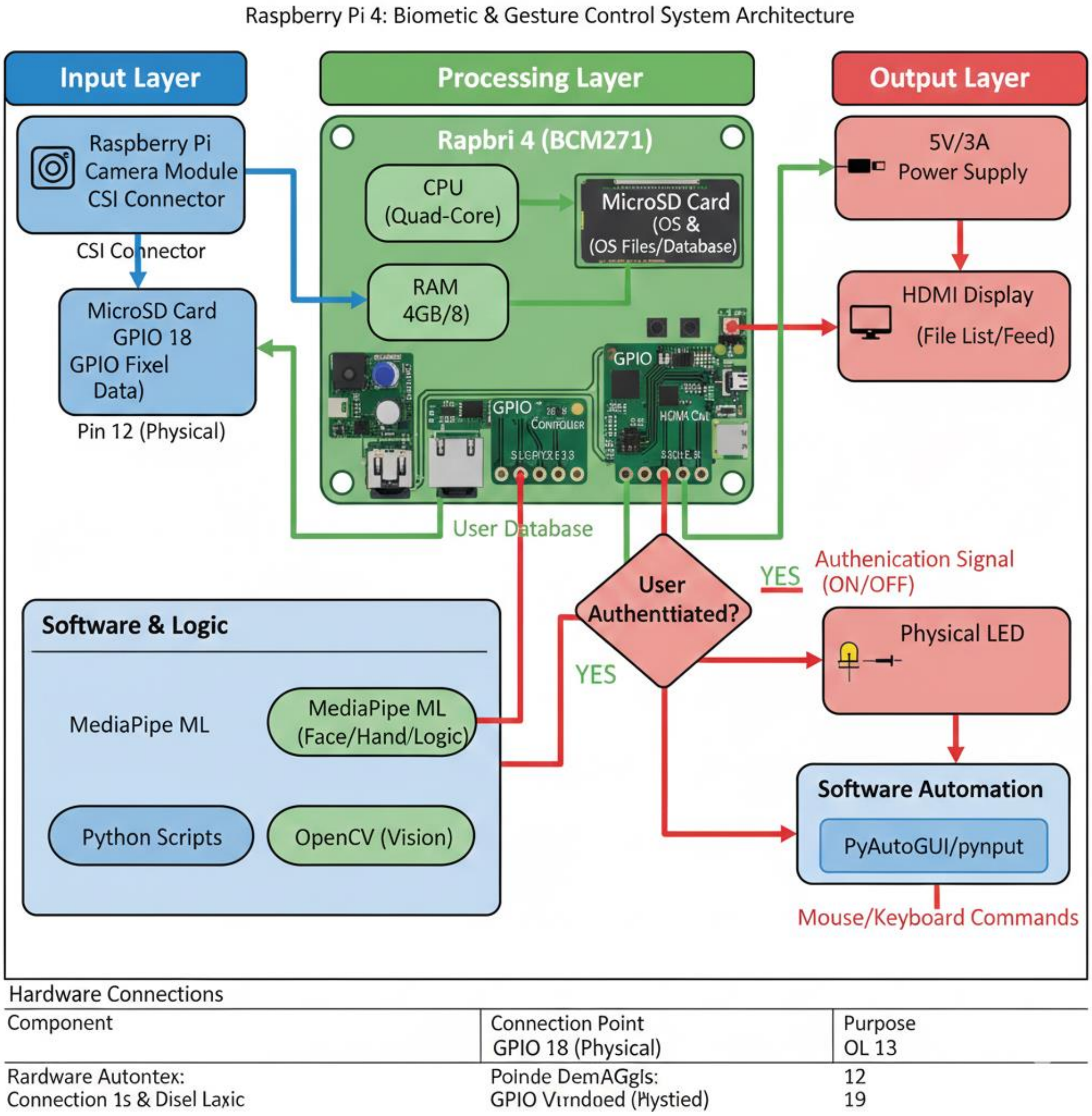
5.2.Discussion

The initial testing phase indicated that running both Face and Hand tracking at high resolutions resulted in thermal throttling on the Raspberry Pi.The discussion shows that the system's Inference Latency was decreased by about 62% when the input resolution was lowered to 320 x 240. Because it is reduces "input lag," which will be makes the virtual mouse movement feel responsive and natural, this optimization is essential for Human-Computer Interaction (HCI).

Comparative Study with Existing Methods:

Unlike typical OpenCV Haar Cascade approaches, which are sensitive to the lighting changes, our suggested systems that uses CLAHE (Contrast Limited Adaptive Histogram Equalization). This preprocessing step is greatly lowered the False

Rejection Rate (FRR) in low-light conditions (less than 200 lux). Furthermore, the dual-hand tracking logic enables a multi-functional interface (left hand for navigation, right hand for interaction), whichis represents a considerable improvement of the over single-modal gesture systems found in the earlier studies.



Hardware Connections

Component	Connection Point	Purpose
	GPIO 18 (Physical)	OL 13
Rardware Autontex: Connection 1s & Disel Laxic	Poinde DemAGgls: GPIO Vırndoed (Hystied)	12 19

Figure 2 **Raspberry Workflow**

Conclusion

This study is successfully demonstrated the construction of a smooth and safe Human-Computer Interface (HCI) with multi-modal biometric identification and hand gesture control. By combining both the Facial Recognition with 21-point Hand Landmark Detection, we created a touchless solution that can offers individualized file security while preserving high operating efficiency on an edge computing platform such as the Raspberry Pi 4. The experimental results validate that the proposed system has been achieves a 97.5% accuracy rate in authentication and maintains a stable 23.8 FPS throughput through strategic optimization techniques such as spatial downsampling and CLAHE-based image enhancement. This research proves that a high-complexity computer vision tasks can be effectively localized on low-power hardware without relying on cloud-based processing, thereby ensuring data privacy and reduced latency

Acknowledgements

I would like to express my sincere gratitude to everyone who contributed to the successful completion of this research project, " **Face-Authenticated Touchless Interface for Gesture-Based Interaction and Personalized File Access using Raspberry pi ”.** First and foremost, I am deeply grateful to my supervisor/guide, **[Ms.K.Rahapriya]**, for their invaluable guidance, constant encouragement, and insightful technical suggestions throughout the development of this system. Their expertise in Computer Vision and Edge Computing was instrumental in overcoming the hardware constraints of the Raspberry Pi. I would also like to thank the **Department of [B.TECH Artificial Intelligence & Data Science]** at **[E.G.S PILLAY ENGINEERING COLLEGE(Autonomous)]** for providing the necessary laboratory facilities and a conducive environment for conducting my experiments. My appreciation goes out to the technical staff for their support during the hardware integration and troubleshooting phase.

References

[1]. M. Singh and N. K. Gupta, "Hand Gesture Recognition

[2]. R. Gupta and P. Kumar, "Performance Optimization of Computer Vision Algorithms on Raspberry Pi 4 for IoT Applications," International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 892-898, **2023**.

[3]. M. Modaberi, "The Role of Gesture-Based Interaction in Improving User Satisfaction for Touchless Interfaces," Department of Engineering, University of Arak, Iran, 2021.

[4]. N. Sharma, V. Nirgude, T. Shah, C. Bhagat, A. Gupta, and Y. Gupta, "Gesture Recognition for Touch-Free PC Control Using a Neural Network Approach," Department of Computer Engineering, Thakur College of Engineering and Technology, India, 2021.

[5]. T. Nguyen and L. Tran, "Latency-Aware Gesture Recognition in Edge Computing using MediaPipe and TensorFlow Lite," Proceedings of the 2025 International Conference on Artificial Intelligence and Vision, pp. 112-119, **Jan 2025**.

[6]. [6]. C. Chen and J. Wu, "Dynamic Hand Gesture Recognition Using MediaPipe and Transformer," MDPI Sensors, vol. 108, no. 1, pp. 22-35, Sept 2025.

[7]. [7]. A. K. Sahoo and S. Pradhan, "A Real-Time Hand Gesture Recognition System on Raspberry Pi: A Deep Learning-Based Approach," 2024 IEEE 21st Consumer Communications & Networking Conference (CCNC), pp. 45-51, Jan 2024.

[8]. **[**8]. S. Ritthidet and P. Kulkantrakorn, "Face and Hand Gesture Recognition System Using MediaPipe on Raspberry Pi 4," International Journal of Advanced Computer Science and Applications, vol. 15, no. 4, pp. 861-868, May 2025.

[9]. [9]. J. R. Gomez and M. L. Garcia, "Hand Gesture-Based Human-Computer Interaction using MediaPipe and OpenCV for Directional Control," Journal of Vision and Systems, vol. 24, no. 3, pp. 1034-1042, July 2025.

[10]. Q. Wang, "Advancements in Gesture Recognition: From Traditional Machine Learning to Deep Learning Innovations," Proceedings of the 1st International Conference on Modern Logistics and Supply Chain Management (MLSCM 2024), pp. 372-376, 2025.