A Novel and Refined Contactless User Feedback System for Immediate On-Site Response Collection

Harold Harrison[1], Mazlina Mamat[2]*, Farrah Wong[3], Hoe Tung Yew[4]
Applied Predictive Analytics Research Group-Faculty of Engineering-Universiti Malaysia Sabah, Sabah, Malaysia[1, 2, 3]
Electronic Engineering (Computer) Program-Faculty of Engineering-Universiti Malaysia Sabah, Sabah, Malaysia[4]

***Abstract*—This paper introduces a Contactless User Feedback System (CUFS) that provides an innovative solution for capturing user feedback through hand gestures. It comprises a User Feedback Device (UFD), a mobile application, and a cloud database. The CUFS operates through a structured sequence, guiding users through a series of questions displayed on an LCD. Using the Pi Camera V2 for contactless hand shape capture, users can express feedback through recognized hand signs. A live video feed enhances user accuracy, while secure data transmission to a database ensures comprehensive feedback collection, including timestamp, date, location, and a unique identifier. A mobile application offers real-time oversight for administrators, presenting facility status insights, data validation outcomes, and customization options for predefined feedback categories. This study also identifies and strategically addresses challenges in image quality, responsiveness, and data validation to enhance the CUFS's overall performance. Innovations include optimized lighting for superior image quality, a parallel multi-threading approach for improved responsiveness, and a data validation mechanism on the server side. The refined CUFS demonstrates recognition accuracies consistently surpassing 93%, validating the effectiveness of these improvements. This paper presents a novel and refined CUFS that combines hardware and software components, contributing significantly to the advancement of contactless human-computer interaction and Internet of Things-based systems.**

Keywords*—*Contactless; human-computer interaction; Internet of Things; machine learning

I. Introduction

The Feedback System (FS) is a platform that empowers users to actively engage and provide feedback, thereby establishing a channel for communication between clients and the respective company or authorities. This system has gained widespread adoption, particularly by companies heavily reliant on customers' opinions as a requirement for product and service improvement [1]. These opinions encompass various aspects such as product evaluation, facility experience, customer treatment, and overall user experience [2-3]. By going through user reviews, organizations can maintain user loyalty [4] by targeting their varying tastes [5] and ensuring they're satisfied with the current services.

Within facilities management, user feedback constitutes contextual information that demands prompt attention and an immediate response from the responsible party. The user feedback system allows facility users to alert workers [6] of any infrastructure-related problems and their overall experiences while using it [7]. There has been significant progress in the development of feedback systems tailored to facility monitoring, driven by a desire to better understand how people utilize buildings [8]. One such innovation is a user feedback system that analyzes user behavior within building environments. Building occupants can provide feedback about their comfort levels within specific spaces, enabling a comparison of this feedback with sensor data to evaluate whether building services meet user needs [9]. This synergy of user input and sensor data can potentially enhance building efficiency and user comfort.

A. Existing Facilities Management Feedback System

Among the most prevalent mechanisms for collecting such feedback are QR codes, which direct users to dedicated websites or online survey forms designed for feedback submission [3]. Using a QR code or physical note is a valid and practical alternative for gathering feedback, but it has some limitations. Users must actively scan a QR code or navigate to a specific URL to provide feedback [3]. Lengthy or complex forms can overwhelm users, leading them to abandon the feedback process [10-11]. Some users may lack the knowledge to access links provided in QR codes or have limited experience with web-based forms. These users are more accustomed to verbal communication rather than written or contextual communication. Furthermore, data collected through QR codes and forms is generally not in real-time. Responses become available only after users submit the form or scan the QR code [12], which might not be ideal for time-sensitive scenarios where immediate feedback is crucial.

Previous studies have incorporated sensors to establish a connection between the comfort levels in facilities. Users are required to provide reviews, which are then used to create a comfort map in conjunction with the sensor data. This information allows the sensors to serve as indicators for when the facilities may require maintenance or the attention of the facility owner. The problem with this method is its limited scalability and adaptability. The system struggles to accommodate an increasing number of facilities and to adapt to different types of facilities, such as transitioning from public services to laboratories.

The on-site feedback system could address many of these limitations and provide a more engaging and efficient feedback experience. This immediate feedback mechanism can lead to more accurate and timely responses. The real-time nature of the system allows for dynamic adjustments and improvements based on the data collected, contributing to a more responsive and user-centric environment. The adaptability of the system increases if facility owners have the ability to modify the review questions. However, when deployed in a public setting, such a system would affect the sanitary level due to the high volume of physical interactions [13] it necessitates. A contactless user feedback system presents a promising solution as it minimizes physical contact and reduces the risk of spreading germs or infections.

B. Existing Contactless Human-Computer Interaction

A contactless system operates through sensors capable of detecting various human signals, such as body and hand movements [14], as well as reactions [15]. Several innovative solutions can be employed to develop this technology, using ultrasonic [16] and infrared sensors [17] as virtual buttons. With this approach, users simply need to hover their hands near one of the sensors corresponding to their desired response, eliminating the need for physical contact.

Another viable option involves gesture sensors [18], which can detect changes in light and discern the direction of motion of an object in front of them. By utilizing this technology, users can select or provide feedback by moving their hand in specific directions, for instance, from left to right [19], without the need to touch the feedback system monitor. The method is viable for an on-site review system but is limited by the number of inputs it can accept. For instance, the ultrasonic method can only handle a few inputs and is sensitive to its surroundings at certain angles. This has been improved with infrared sensors, which offer better accuracy but have limited range. As a result, users must interact with the system within a restricted distance, and any potential infrared interference in the environment can cause the device to malfunction.

A more advanced solution involves the implementation of Artificial Intelligence (AI), using hand-sign optical images as input [20-22]. Available AI options include MediaPipe, an open-source framework that enables developers to construct complex pipelines for object detection, face detection, hand tracking, pose estimation, and more [23-25]. MediaPipe provides a solid foundation for building real-time multimedia processing pipelines, but the accuracy varies based on the task being performed [26-27]. Accurate interpretation of user responses is very important for the contactless system to ensure reliability. Accurateness is particularly crucial within the context of performance reviews, where feedback, provided at the right moments and for the right purposes, plays a pivotal role. This method incorporates the use of an AI model to map hand landmarks to specific movements, simulating mouse movements or keyboard keystrokes. However, this approach requires significant computational power, leading to an unpleasant stuttering experience on small form factor devices. Potential solutions include optimizing the AI model or implementing a transition algorithm to convert the AI model's output into computer input more efficiently.

C. The Contribution and Objective of the Study

This paper presents a pioneering contactless user feedback system that converges hardware and software technologies to revolutionize user interaction and feedback processes. By integrating the Raspberry Pi 4B+ microcomputer, Pi Camera V2, and the MediaPipe framework, the system introduces a contactless paradigm for real-time hand sign recognition, providing users with an intuitive medium for expressing immediate feedback. The study makes a substantial contribution to contactless human-computer interaction by systematically addressing and overcoming challenges associated with image quality, responsiveness, and data validation.

II. METHODOLOGY

A. The Contactless User Feedback System Assembly

A prototype of a Contactless User Feedback System (CUFS) depicted in Fig. 1 was developed in this study. It comprises a User Feedback Device (UFD), a cloud database, and a mobile application. The central processing unit for the CUFS is the Raspberry Pi 4B+ microcomputer, chosen for its ability to connect to an external display, robust processing power for AI applications [28-29], internet connectivity for cloud database integration, and support for computer vision applications.

The contactless approach is established using a Pi Camera V2, where it is used to capture the user's hand signs. Five hand signs, as illustrated in Fig. 2, are used to facilitate response submission. The hand signs are interpreted using MediaPipe, a machine-learning pipeline that provides a wide range of pre-built solutions and tools for tasks like object detection, pose estimation, hand tracking, and face recognition.

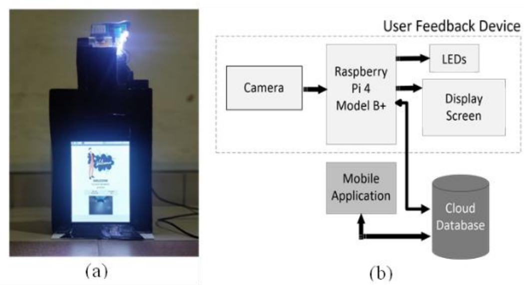

Fig. 1. Contactless user feedback system (a) UFD (b) Block diagram.

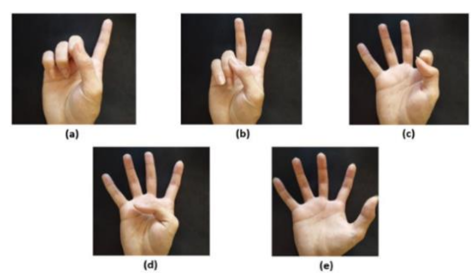

Fig. 2. The recognized hand signs are (a) one, (b) two, (c) three, (d) four, and (e) five.

To facilitate user interaction with the UFD, a seven-inch LCD screen provides instructions and guidance to users. Upon activating the UFD, users are welcomed with a message and clear instructions for assisting them throughout the process. Subsequently, the device presents a question, prompting users to respond. After receiving the response, the next question is displayed for further input. This sequential process repeats until all questions have been addressed. Upon completing the feedback process, the device securely transmits the data to the cloud database. The data includes valuable information such as the timestamp, date, location, and unique identifier, ensuring comprehensive and insightful feedback collection. Fig. 3 defines the CUFS's flow of operation.

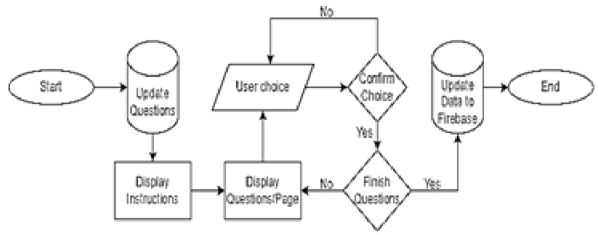

Fig. 3. The CUFS's flow of operation.

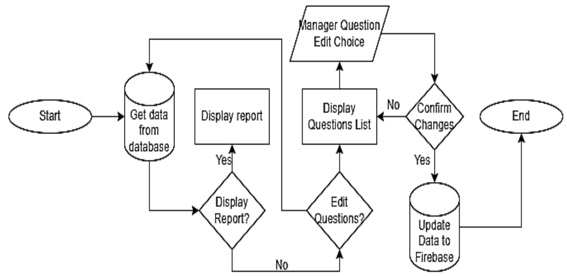

Fig. 4. Managerial application of the CUFS.

Fig. 5. Illumination area.

The mobile application offers two navigational choices, each catering to distinct aspects of system oversight. The first choice displays a 'Report Page,' a comprehensive dashboard that offers insights into overall facility status, data validation outcomes, and current operational conditions. This report page provides statistical data for facility conditions across different timeframes, encompassing daily, weekly, monthly, and annual perspectives. The second choice leads to a 'Question Edit Page', where the predefined questionnaire can be modified. This page displays the existing questions, with options to select, edit, or delete them. Any editing action prompts a confirmation step, ensuring that changes are deliberate. Once confirmed, the mobile application updates the amended list of questions in the database. The flowchart in Fig. 4 outlines the mobile application for the CUFS.

B. *On-Site Testing*

The CUFS underwent on-site testing, which uncovered deficiencies in image quality, responsiveness, resource management, and data validation. The subsequent sections delve into these limitations and detail the enhancements implemented to address them.

1) Image quality: The CUFS faces difficulties in hand detection and landmark identification, occasionally initiating premature detection and calculations, leading to data collection errors. Factors like background color, reflected light intensity from the hand, and background lighting contribute to suboptimal image quality, hindering accurate user interaction detection. MediaPipe misidentifying landmarks adds complexity, potentially affecting the precision of user gestures. Despite efforts with smart algorithms [30] to address low-light issues, solutions often involve increased computational load or sensor modifications [31-32].

In this study, the solution to address the challenge of poor hand images involves optimizing the intensity of reflected light from the user's hand. This improvement is achieved by strategically redirecting light at a specific angle, illuminating only the user's hand, as depicted in Fig. 5. This enhanced lighting also serves as a guide, indicating where to place the hands. The software was improved to initiate hand sign detection only when all hand landmarks (Fig. 6) are within the frame. To rectify the issue of incorrect landmark position identification, a collection of hand images is obtained for each question, with the most frequently occurring hand sign serving as the response.

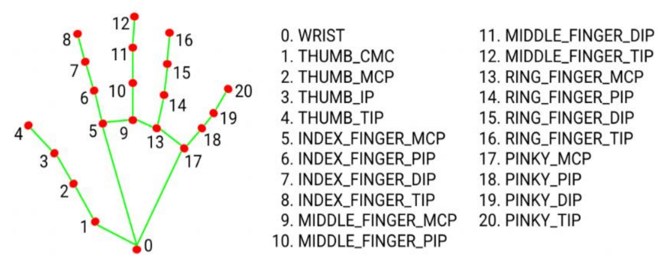

Fig. 6. Hand landmarks with keypoint localization of 21 hand-knuckle coordinates.

The hand images are processed using Algorithm 1. The algorithm takes the parameters of all the hand landmarks in Fig. 6, identified by the hand recognition algorithm offered by the MediaPipe Solutions. It is intended to accept right-hand signs by focusing on the index, middle, ring, and pinkie fingers to compare the tip and proximal interphalangeal joint (PIP) positions. However, when dealing with the thumb, a distinct approach is taken due to its horizontal movement, in contrast to the vertical movement of the other four fingers during hand contraction. Consequently, only the x-axis coordinates are considered for the thumb, while the y-axis coordinates are used for the other fingers.

2) Responsiveness and Resource management: The initial CUFS features a sequential programming flow, as detailed in Fig. 7. This programming structure, while simple, contributes to decreasing responsiveness over time. Video frames undergo extensive processing steps to display questions and recognize responses. These procedures significantly strain the system, leading to a noticeable decrease in frame rate. Additionally, delays can compound when the display process is required to wait for concurrent tasks such as answer handling, page management, and Tkinter rendering. Consequently, extended waiting times pose a challenge to the system's capacity to deliver prompt and responsive feedback.

Algorithm 1: Finger Counting Algorithm

```
Initialize finger up count to 0
Compute
While (Get hand landmarks) do
  | For (every hand-knuckle coordinate) do
  | Get Thumb IP's and TIP's x-axis coordinates
  | If (x-axis coordinates of IP > TIP)
  |       | Increase finger up count by one
  |       | For (the rest of the fingers) do
  |       |   | Get y-axis coordinates of PIP and TIP
  |       |   | If (y-axis coordinates of PIP > TIP) then
  |       |   |     Increase finger up count by one
  | End   |   | End
  | End

Return finger up count
```

To address these challenges, multi-threading was initially implemented as a promising solution. However, over time, complications arose, leading to delays that gradually accumulated. After extended periods of inactivity, the system encountered an average delay of 3159 milliseconds, roughly a 3-second lag to accept a response. Resource queuing is an important aspect contributing to this issue, where resources can only be released after being used by all threads requiring them. This approach introduces complexities when managing threads with differing resource usage times and burst rates, ultimately affecting system performance and responsiveness.

The issues regarding responsiveness were effectively addressed through the implementation of parallel programming [33] and multi-threading. The program was logically divided into three components to incorporate multi-threading in Python, as shown in Fig. 8. The initial component focused on hand recognition, encompassing all hand landmark retrieval and computation tasks. The subsequent component handled the program's logic for processing user inputs. The final component was responsible for managing the feedback display from the Pi camera. Before implementing multi-threading, a few rules must be followed to maintain the integrity of the different threads.

A resource queuing solution was implemented to resolve this resource contention (Fig. 9). This introduced a third module or component, referred to as the capture component, which took on the responsibility of capturing video frames and placing them in designated queues. Each component operated independently as threads, ensuring a smoother and more responsive program execution.

3) Data validation: The developed CUFS faces issues with data validation, where it could not validate the feedback provided by users. This limitation raises concerns regarding the legitimacy and authenticity of the feedback received. In practical terms, this limitation can manifest as a scenario where employees submit feedback to inflate their ratings, potentially compromising the integrity of the data collected. As such, the absence of validation mechanisms underscores the need for improvements to ensure the trustworthiness and reliability of the CUFS.

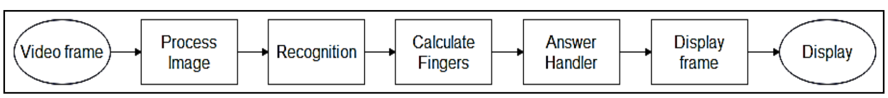

Fig. 7. Series programming flow of the UFD.

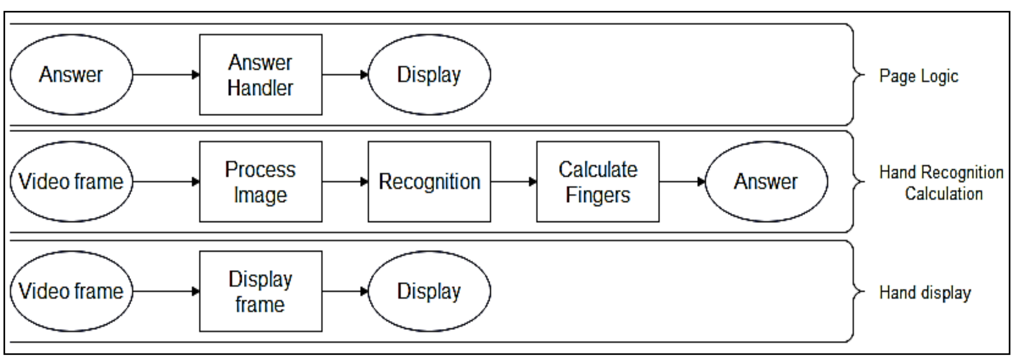

Fig. 8. The components of the UFD program.

For improvement, data validation is performed in the server by using the ratings and the time of those reviews. The server will run a scheduled validation for every prescribed duration. Rather than fetching all the data, including previously validated reviews, the server accesses a validation history stored in the database. This history will be used to identify and retrieve only the reviews that have yet to be validated. This approach significantly reduces the volume of data that requires review, which is particularly advantageous when dealing with large datasets. The data will be systematically sorted and organized upon retrieval into a list that aligns with the validation algorithm's requirements. The resulting list will then be processed, and the outcomes will be uploaded to the database. This approach streamlines the validation process and optimizes data handling, improving overall efficiency and resource utilization. Fig. 10 shows the flowchart for the data validation server on the local machine.

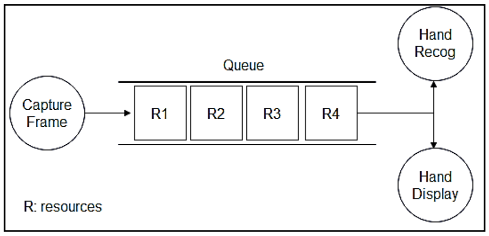

Fig. 9. Implementation of resource sharing from one-to-many threads.

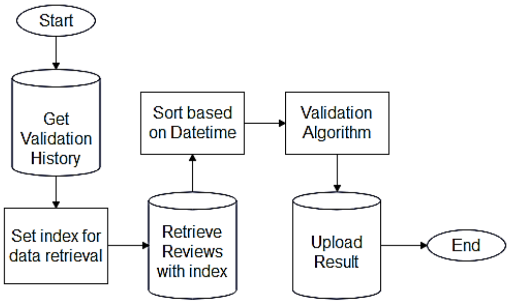

Fig. 10. Flowchart of data validation.

Algorithm 2 provides a structured approach to ensure the integrity of feedback data. This algorithm consists of two key functions, the first of which involves converting the ratings from the qualitative "good" and "bad" labels into a binary representation (1 for "good" and 0 for "bad"). This conversion simplifies the comparison process, making it more efficient. Once the ratings are converted into binary form, the algorithm proceeds with the analysis. The function responsible for the calculation takes two parameters: the rating list and the time period of the listings. By default, the time period is set to every 1 hour. However, this can be adjusted according to the facility owner's preferences to accommodate different facilities. The algorithm iterates through the binary list and checks for changes in ratings over time. Whenever the binary rating at the current time period differs from the previous one (excluding the initial binary), the change counter increments by one.

Algorithm 2: Validation Algorithm

```
Initialize finger up count to 0
rating list ex: {good, good, bad, good, bad, good}

Function converts ratings to binary (ratings list)
    | Initialize an empty binary list
    | For (every rating in rating list) do
    | If (rating is 'good') then
    |     | Append' 1' to binary list
    | Else (the rest of the fingers) then
    |     | Append' 0' to binary list
    | End
    | End
    Return binary list
End

Function calculates rate of change (list, time period)
    Compute binary list as converts ratings to binary (list)
    Initialize changes to 0
    For (every binary in binary list) do
        | If (current binary is not equal to previous binary) then
        |     | Increase changes by 1
        | End
        | End
    Compute rate of change as changes/ time period
    Return rate of change
End
```

This process continues until the algorithm reaches the end of the list. Finally, the algorithm calculates the rate of change by dividing the change count by the specified period for the sample being evaluated. This rate of change provides valuable insights into the consistency of feedback. Before executing the validation algorithm, it is imperative to ensure the data is pre-sorted based on date and time, facilitating a systematic analysis of feedback patterns. A simple pattern analysis has been developed to ensure the integrity of the feedback data. The algorithm collects feedback over a defined period and subjects it to pattern analysis. This approach helps detect anomalies and inconsistencies in the reviews. For instance, if a worker attempts to manipulate information by providing positive reviews while neglecting their responsibilities concerning the facility's condition, a pattern of inconsistent feedback emerges over time. If this deceptive behavior persists in subsequent periods, the system triggers alert to notify the manager about potential issues with the facility. This proactive approach ensures data accuracy and helps maintain the reliability of the feedback system.

III. Results and Discussion

A. *Post-Improvement Testing*

The improved CUFS was evaluated in real-time by six respondents using the following procedure:

- Respondents will receive a predefined sequence of five questions, guiding them through the CUFS.
- Respondents must answer each question by using hand sign feedback.
- If a respondent encounters a misidentification, they must notify the researcher for documentation.
- Respondents should resume and complete the sequence, notifying the researcher upon completion.
- Respondents must repeat the aforementioned steps five times. In the first cycle, the respondent must give a hand sign 'one'. In the second cycle, the respondent must give a hand sign 'two', etc.

In addition to real-time assessments, performance testing evaluates the system's responsiveness and stability under a certain workload [34]. Different performance tests were conducted: load, stress, and soak tests. Load tests simulate the maximum number of possible users that might use an application. Reproducing realistic usage and load conditions based on response times will identify potential bottlenecks [35]. Stress testing measures the performance of a system in peak activity, which involves an increment of users during the testing. This specific test helped identify any potential vulnerabilities in the system [36]. The last type of testing would be soaking test, which increases the number of users for a longer period to detect any drop in performance levels along the run.

Reviews of the CUFS were also conducted. To gather user feedback, the CUFS was left at the designated testing facility along with a QR code leading to a Google Form. Users were prompted to share their insights on the device's performance through a series of questions, each featuring a 5-point rating scale:

- Is the CUFS's operation smooth?
- Is the CUFS easy to use?
- Is the CUFS easy to understand?
- Does the CUFS accurately detect all my choices?

On the managerial side, reviews were primarily acquired through online channels and face-to-face interviews with facility personnel. Managers had the opportunity to use the system for several days, gaining hands-on experience. Subsequently, they provided feedback on various aspects, including responsiveness, identification of bugs, user-friendliness, and suggested improvements. This multifaceted approach ensures a thorough evaluation from both end-users and managerial perspectives.

B. Real-Time Recognition Performance

The performance of the CUFS in recognizing hand signs, post-improvement, is presented by the confusion matrices in Fig. 11. Two background scenarios were analyzed: simple (uniform) and complex (dynamic). The simple background does not contain any objects, while the complex background contains objects with various colors. A total of 30 hand inputs coming from six users (each contributed five hand sign) were collected for each class, resulting in 150 hand inputs for each background scenario. Each of the user have varied skin tone from dark to pale. The average accuracy, precision, recall, and F-1 score have been computed to assess the overall recognition performance (Table I). Results show that the CUFS achieved impressive accuracy levels, with more than 93% accuracy in simple and complex backgrounds. The CUFS also demonstrated high precision, recall, and F-1 scores across both background scenarios. This consistency highlights the CUFS's reliability in correctly identifying hand signs and its ability to minimize false detection.

Further analysis reveals that most discrepancies observed are attributed to variations in the reviewer's response attitude. Certain reviewers did not form the hand shape properly and made intermittent or sudden changes when giving the response. Though slight, the decrease in performance metrics in complex backgrounds points to the system's sensitivity to background variations. This sensitivity suggests a potential area for improvement, particularly in enhancing the system's ability to distinguish hand signs from visually noisy backgrounds. While the system performs well in controlled settings, its application in real-world scenarios, where background complexity and user behavior are less predictable, may present challenges. Understanding the limitations in these contexts is crucial for further development and deployment of the CUFS.

C. System Performance

The system device, operating on a Raspberry Pi, exhibits efficient resource utilization. The application, primarily running the hand recognition solutions, consumes an average of 42% of the CPU processing resources, as anticipated. This usage aligns with the computational demands of the machine learning algorithms employed. In terms of memory utilization, the system operates justly, utilizing only approximately 170.1 MB during runtime.

This accounts for less than 10% of the total available memory on the Raspberry Pi 4, indicating a well-optimized use of resources. Moreover, the storage footprint is minimal, with only 338.3 KB utilized for this project. This represents the lowest utilization among the system's resources, highlighting an efficient design that minimizes storage requirements while maintaining the necessary functionality. Table II records the resource utilization of the UFD.

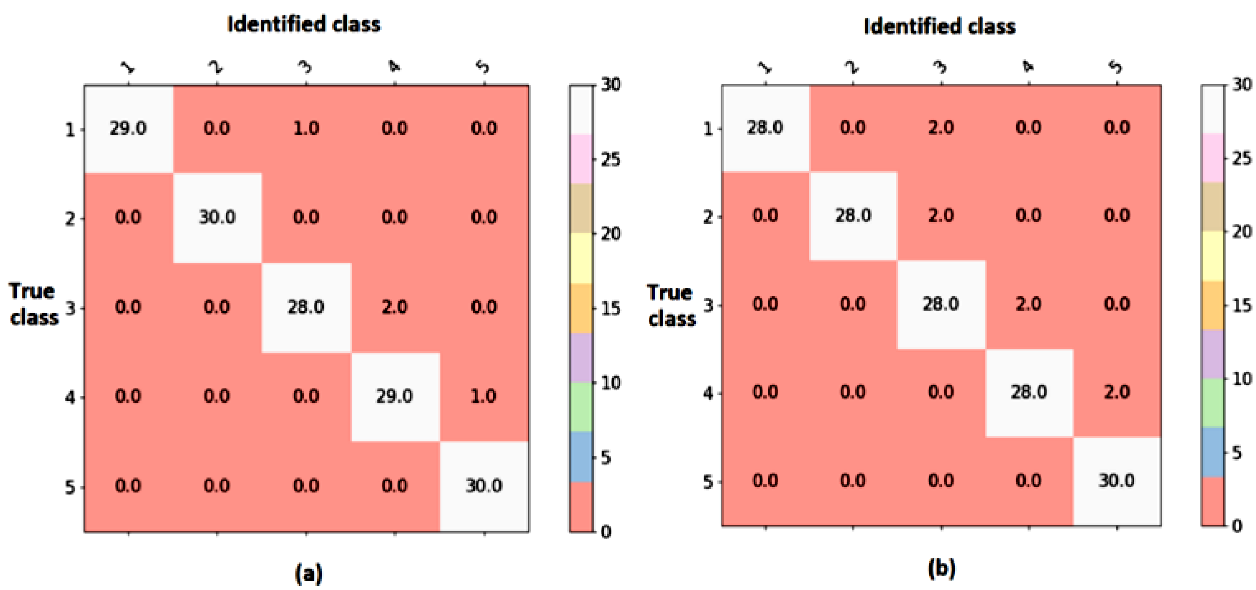

Fig. 11. Confusion matrices for (a) Simple background and (b) Complex background.

TABLE II. REAL-TIME RECOGNITION PERFORMANCE

Background	Simple						Complex					
Class	*1*	*2*	*3*	*4*	*5*	*Average*	*1*	*2*	*3*	*4*	*5*	*Average*
Accuracy	0.967	1.000	0.933	0.967	1.000	0.9734	0,933	0,933	0,933	0,933	1.000	0.9464
Precision	0.967	1.000	0.917	0.933	0.983	0.960	0.933	0.933	0.867	0.900	0.967	0.920
Recall	0.967	1.000	0.933	0.966	1.000	0.973	0.933	0.933	0.933	0.933	1.000	0.946
F-1 score	0.967	1.000	0.922	0.944	0.988	0.964	0.933	0.933	0.889	0.911	0.978	0.929

The soak test involved assessing the device's prolonged functionality under continuous operation to identify potential errors over an extended period. This test was conducted by leaving the device operational overnight, and it demonstrated robust performance, operating without any noticeable issues. Evaluating stress testing outcomes is intricate because this process typically involves multiple programs with diverse requests. It is essential to note that the current system has only one prototype available for testing, limiting the ability to comprehensively analyze requests from multiple users. Most reviews gathered for the CUFS were verbal, involving six users. They agree on the system's responsiveness and user-friendly interface. However, some expressed concerns about the small screen size and challenging font readability. Additional feedback included complaints about color schemes, styles, overall size, and minor features. In response to this feedback, specific enhancements were made, incorporating the addition of page numbers, a color indicator highlighting the current answer, and adjustments to font size.

The results confirm the effectiveness of the enhanced system development, as outlined in the methodology, which incorporates hardware and software modifications. The parallel multi-threading implementation has successfully addressed the low frame rate issue during the camera input display. Additionally, improvements in lighting, finger calculation, and the mod selection algorithm have notably boosted detection accuracy. Transitioning from a static image to a dynamic question format, editable by administrators, adds flexibility to the graphical user interface. Furthermore, the inclusion of page selection and robust data validation enhances the system's functionality. A summary of the comprehensive enhancements is provided in Table III, illustrating significant advancements in both features and overall system performance for the contactless user feedback device and the mobile application.

IV. CONCLUSION

This paper presents an innovative Contactless User Feedback System (CUFS) designed to enhance user interaction and feedback reliability. Using Raspberry Pi 4B+ microcomputer, Pi Camera V2, and MediaPipe for hand shape interpretation, the CUFS successfully integrates hardware and software components. Despite initial image quality and responsiveness challenges, the study systematically addresses these issues through a series of improvements. Post-testing refinements, including optimized lighting for improved image quality, parallel multi-threading for enhanced responsiveness, and a data validation mechanism, underscore the commitment to refining the CUFS's performance. Real-time recognition performance and comprehensive system testing further validate the effectiveness of these enhancements. User feedback in real-time assessments and reviews has been pivotal in shaping the system's evolution, highlighting the CUFS's responsiveness to end-users needs. In addition, adjustments based on user insights, such as font size, color schemes, and feature enhancements, demonstrate a user-centric approach. The CUFS study contributes valuable insights into human-computer interaction, offering a comprehensive understanding of challenges, innovative solutions, and iterative improvements necessary for developing reliable and user-centric feedback mechanisms. Future considerations should encompass scalability assessments and a broader exploration of system generalizability to different environments and user demographics.

TABLE III. RESOURCE UTILIZATION OF THE USER FEEDBACK DEVICE

	Resources		
	Average CPU Utilization (%)	*Memory Usage (MB)*	*Storage Size (KB)*
Min	32%	169.0	-
Max	43%	171.0	338.3
Mean	42%	170.1	-

TABLE IV. COMPARISON BETWEEN PRE- AND POST-IMPROVEMENT

Contactless User Feedback Device			Mobile Application		
Features	*Pre-improvement*	*Post-improvement*	Features	*Pre-improvement*	*Post-improvement*
Contactless	Yes	Yes	***Statistic Graph***	No	Yes
Accuracy	<60%	>90%	***Review Validation***	No	Yes
Detection Rate	<60%	>90%	***Question edit page***	No	Yes
Mod selection	No	Yes			
Multi-threading	No	Yes			
Responsive	No	Yes			
Lighting solution	No	Yes			
Video Feedback	No	Yes			
Page Navigation	No	Yes			
Choice Indicator	No	Yes			

ACKNOWLEDGMENT

This study was supported by the Malaysian Ministry of Higher Education (MOHE) Fundamental Research Grant Scheme (FRGS/1/2023/TK10/UMS/02/1) and Universiti Malaysia Sabah internal research grant SBK0466-2021.

REFERENCES

[1] S. Tolf, J. Mesterton, D. Söderberg, I. Amer-Wåhlin, and P. Mazzocato, "How can technology support quality improvement? Lessons learned from the adoption of an analytics tool for advanced performance measurement in a hospital unit," *BMC Health Serv. Res.*, vol. 20, no. 1, 2020.

[2] A. Klapalová, "Customer product returns – feedback and knowledge management," *Meas. Bus. Excell.*, vol. 23, no. 2, 2019, pp. 149–164.

[3] H. Abdel Rady and H. Elsbayee, "Evaluation of After Sales Services in Airline Companies Case study: EgyptAir," Minia J. Tour. Hosp. Res. MJTHR, vol. 8, no. 1, 2019, pp. 189–240.

[4] G. LEW, "Customer Relations in Building Value for the Customer in Commercial Enterprises," Humanit. Soc. Sci. Q., vol. 25, 2018.

[5] E. Soliha, A. Aquinia, R. Basiya, P. Waruwu, and M. Kharis, "Service Quality and Location towards Customer Value and the Effect on Customer Satisfaction BT - Proceedings of the International Conference on Banking, Accounting, Management, and Economics (ICOBAME 2018)," Atlantis Press, Jul. 2019, pp. 74–77.

[6] A. Ingrao, "Assessment by Feedback in the On-demand Era," 2018, pp. 93–111.

[7] M. Al Amin, M. S. Arefin, N. Sultana, M. R. Islam, I. Jahan, and A. Akhtar, "Evaluating the customers' dining attitudes, e-satisfaction and continuance intention toward mobile food ordering apps (MFOAs): evidence from Bangladesh," Eur. J. Manag. Bus. Econ., 2020.

[8] M. Monsberger, D. Koppelhuber, V. Sabol, H. Gursch, A. Spataru, and O. Prentner, "An innovative user feedback system for sustainable buildings," IOP Conf. Ser. Earth Environ. Sci., vol. 323, Sep. 2019, p. 12123.

[9] F. Zhou *et al.*, "Visually enhanced situation awareness for complex manufacturing facility monitoring in smart factories," *J. Vis. Lang. Comput.*, vol. 44, 2018, pp. 58–69.

[10] D. Ramsauer, M. Dorfmann, H. Tellioglu, and W. Kastner, "Human Perception and Building Automation Systems," *Energies*, 2022.

[11] R. E. Rice, S. P. Aagarwal, and P. T. Kortum, "Effects of Task Difficulty and Presentation Order in Subjective Usability Measurement," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 67, no. 1, 2023, pp. 2168–2172.

[12] S. Goemaere, W. Beyers, G.-J. De Muynck, and M. Vansteenkiste, "The paradoxical effect of long instructions on negative affect and performance: When, for whom and why do they backfire?," *Acta Astronaut.*, 2018.

[13] C. Rothe *et al.*, "Transmission of 2019-nCoV Infection from an Asymptomatic Contact in Germany.," *The New England Journal of Medicine*, vol. 382, no. 10. United States, Mar. 2020, pp. 970–971.

[14] . Karbasi, Z. Bhatti, R. Aghababaeyan, S. Bilal, A.E. Rad, A. Shah, and A. Waqas, "Real-time hand detection by depth images: A survey." Jurnal Teknologi, vol 78(2), 2016.

[15] R. M. Fouad, O. A. Omer, and M. H. Aly, "Optimizing Remote Photoplethysmography Using Adaptive Skin Segmentation for Real-Time Heart Rate Monitoring," *IEEE Access*, vol. 7, 2019, pp. 76513–76528.

[16] D. F. Q. Melo, B. M. C. Silva, N. Pombo, and L. Xu, "Internet of Things Assisted Monitoring Using Ultrasound-Based Gesture Recognition Contactless System," *IEEE Access*, vol. 9, 2021, pp. 90185–90194.

[17] L. Zhang, K. Liu, and L. Gao, "Infrared small target tracking in complex background based on trajectory prediction," in *International Conference on Graphic and Image Processing*, 2020, p. 67.

[18] L. Yu, H. Abuella, M. Z. Islam, J. O'Hara, C. Crick, and S. Ekin, "Gesture Recognition Using Reflected Visible and Infrared Lightwave Signals," *IEEE Trans. Human-Machine Syst.*, vol. 51, 2021, pp. 44–55.

[19] D. Tzionas, L. Ballan, A. Srikantha, P. Aponte, M. Pollefeys, and J. Gall, "Capturing Hands in Action Using Discriminative Salient Points and Physics Simulation," *Int. J. Comput. Vis.*, vol. 118, 2015, pp. 172–193.

[20] C. Dignan, E. Perez, I. Ahmad, M. Huber, and A. Clark, "An AI-based Approach for Improved Sign Language Recognition using Multiple Videos," *Multimed. Tools Appl.*, vol. 81, no. 24, 2022, pp. 34525–34546.

[21] B. Ben Atitallah *et al.*, "Hand Sign Recognition System Based on EIT Imaging and Robust CNN Classification," *IEEE Sens. J.*, vol. 22, 2022, pp. 1729–1737.

[22] R. Sousa, T. Jesus, V. Alves, and J. Machado, "Contactless Human-Computer Interaction Using a Deep Neural Network Pipeline for Real-Time Video Interpretation and Classification BT - Advanced Research in Technologies, Information, Innovation and Sustainability," T. Guarda, F. Portela, and M. F. Santos, Eds., Cham: Springer International Publishing, 2021, pp. 209–220.

[23] A. Latreche, R. Kelaiaia, A. Chemori, and A. Kerboua, "Reliability and validity analysis of MediaPipe-based measurement system for some human rehabilitation motions," Measurement, vol. 214, no. 112826. 2023.

[24] J. Lu and E. Peng, "Face gesture recognition module optimization of intelligent security system based on Raspberry Pi," in Proc.SPIE, Aug. 2023, p. 1275430.

[25] S. Alyami, H. Luqman, and M. Hammoudeh. "Isolated Arabic Sign Language Recognition Using a Transformer-based Model and Landmark Keypoints," ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 23 (1), no. 3. 2024.

[26] H. Orovwode, J. A.i Abubakar, O. C. Gaius, and A. Abdullkareem, "The Use of Hand Gestures as a Tool for Presentation," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 14, no 11, 2023.

[27] I. Rodríguez-Moreno, J.M. Martínez-Otzeta, I, Goienetxea, and B. Sierra. "Sign language recognition by means of common spatial patterns: An analysis," PLoS ONE, vol. 17, no 10. 2022.

[28] J. S. Muslim, H. M. Ahmed, R. S. S. Hussain, U. Vali, and R. Mobashar, "Automatic Image Annotation for Small and Ad hoc Intelligent Applications using Raspberry Pi," *MATEC Web Conf.*, 2019.

[29] N. Johari, M. Mamat, Y. Hoe Tung, and A. Kiring, "Effect of Distance and Direction on Distress Keyword Recognition using Ensembled Bagged Trees with a Ceiling-Mounted Omnidirectional Microphone," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 14, no. 6, 2023.

[30] Z. Rahman, M. Aamir, Y.-F. PU, F. Ullah, and Q. Dai, "A Smart System for Low-Light Image Enhancement with Color Constancy and Detail Manipulation in Complex Light Environments," *Symmetry (Basel).*, vol. 10, 2018.

[31] H. Tang, H. Zhu, H. Tao, and C. Xie, "An Improved Algorithm for Low-Light Image Enhancement Based on RetinexNet," *Applied Sciences (Switzerland)*, vol. 12, no. 14, 2022.

[32] H. H. Huang, T. Y. Huang, C. H. Liu, S. Di Lin, and C. Y. Lee, "32 × 64 SPAD Imager Using 2-bit In-Pixel Stack-Based Memory for Low-Light Imaging," *IEEE Sens. J.*, vol. 23, no. 17, 2023, pp. 19272–19281.

[33] A. Dovier, A. Formisano, G. Gupta, M. V. Hermenegildo, E. Pontelli, and R. Rocha, "Parallel Logic Programming: A Sequel," *Theory Pract. Log. Program.*, vol. 22, no. 6, 2022, pp. 905–973.

[34] A. Wert, H. Schulz, C. Heger, and R. Farahbod, "Generic instrumentation and monitoring description for software performance evaluation," *ICPE 2015 - Proc. 6th ACM/SPEC Int. Conf. Perform. Eng.*, 2015, pp. 203–206.

[35] Z. Jiang and A. Hassan, "A Survey on Load Testing of Large-Scale Software Systems," *IEEE Trans. Softw. Eng.*, vol. 41, 2015, pp. 1091–1118.

[36] F. Waheed, F. Azam, M. W. Anwar, and Y. Rasheed, "Model Driven Approach for Automatic Script Generation in Stress Testing of Web Applications," in Proceedings of the 2020 6th International Conference on Computer and Technology Applications, in ICCTA '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 46–50.