International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 37, No. 4, August, 2023, pp. 881-887
Journal homepage: http://iieta.org/journals/ria

A Deep Learning Framework for Hand Gesture Recognition and Multimodal Interface Control

Issam Elmagrouni[1*], Abdelaziz Ettaoufik[2], Siham Aouad[3], Abderrahim Maizate[1]

[1] RITM-EST/CED-ENSEM, Hassan II University, Casablanca 20000, Morocco
[2] LTIM, FS Ben M'SIK, Hassan II University, Casablanca 20000, Morocco
[3] SSL ENSIAS, Mohamed V University, Casablanca 10000, Morocco

Corresponding Author Email: magrouni@gmail.com

https://doi.org/10.18280/ria.370407

Received: 29 April 2023
Revised: 25 May 2023
Accepted: 30 May 2023
Available online: 31 August 2023

Keywords:
deep learning, gesture recognition, hand gesture, computer vision, hand tracking, gesture interface, human-computer interaction

ABSTRACT

Hand gesture recognition (HGR) is an essential technology with applications spanning human-computer interaction, robotics, augmented reality, and virtual reality. This technology enables more natural and effortless interaction with computers, resulting in an enhanced user experience. As HGR adoption increases, it plays a crucial role in bridging the gap between humans and technology, facilitating seamless communication and interaction. In this study, a novel deep learning approach is proposed for the development of a Hand Gesture Interface (HGI) that enables the control of graphical user interfaces without physical touch on personal computers. The methodology encompasses the analysis, design, implementation, and deployment of the HGI. Experimental results on a hand gesture recognition system indicate that the proposed approach improves accuracy and reduces response time compared to existing methods. The system is capable of controlling various multimedia applications, including VLC media player, Microsoft Word, and PowerPoint. In conclusion, this approach offers a promising solution for the development of HGIs that facilitate efficient and intuitive interactions with computers, making communication more natural and accessible for users.

1. INTRODUCTION

The objective of this article is to support individuals with motor impairments and physical limitations while minimizing physical contact with electronic devices and equipment, particularly in the aftermath of the Coronavirus pandemic. This study strives to advance techniques and methods employed in the field of gesture recognition systems (GRS).

One of the primary advantages of GRS is their capacity to provide a natural and intuitive form of interaction with technology. This enables users to control computers or devices using hand movements, obviating the need for traditional input devices such as a mouse or keyboard. GRS are especially beneficial for individuals who may experience difficulty utilizing conventional input devices.

It is worth noting that GRS have numerous applications, encompassing gaming, entertainment, industrial, and medical settings. For instance, in the gaming industry, GRS can be employed to control characters or objects within a game. In the medical field, they can be utilized to operate robotic surgical equipment [1], enhancing precision and control.

Traditional gesture recognition algorithms often face limitations regarding accuracy and response time. The primary goal of this paper is to address these limitations by introducing a novel architecture capable of significantly improving the accuracy and response time of hand GRS. This is achieved through the implementation of an intermediate layer for controlling graphical user interfaces using hand gestures.

By advancing the field of GRS and addressing existing limitations, this study contributes to the development of more efficient and accessible technologies for individuals with motor impairments and physical limitations, as well as for general users who seek to minimize physical contact with electronic devices and equipment.

It is important to mention, this paper presents two main contributions:

- A comprehensive review of recent literature on gesture recognition, highlighting various studies and approaches.
- A novel approach for HGR and Interface Development.

Moreover, the structure of the paper is outlined as follows: Section 2 provides an overview of relevant research, Section 3 details the new approach, Section 4 showcases the findings of the experiments conducted, and Section 5 concludes the paper.

2. LITERATURE REVIEW

Machine learning is used in the developing subject of hand gesture identification to precisely detect human hand motions. Several studies have been read in order to obtain insight into how hand gesture recognition algorithms work. Both dynamic and static gestures are covered in the literature on hand gesture recognition. Static hand gestures are represented by a single image per gesture, whereas dynamic hand gestures are moving gestures that are Showcased by a series of images. Dynamic gestures involve processing several frames, which makes them difficult for real-time applications and devices with limited computational power like mobile phones. Different types of gesture detection techniques can be categorized. by the type of input device, such as Kinect or Leap Motion, or by other types of gestures, for instance hand gestures, facial expressions, body gestures, etc. [2-5].

Ameur et al. [6] introduced an approach for recognizing dynamic gestures through non-contact gestures on a leap motion system. To analyze the sequential time-series data captured from the Leap Motion, the researchers utilized a recurrent neural network, specifically employing the long short-term memory (LSTM). To enhance the accuracy of their model, the authors used both unidirectional and bidirectional LSTMs and further improved their model by introducing a hybrid bidirectional unidirectional LSTM prediction network (HBU-LSTM). This additional component enabled the model to take advantage of the strengths of both unidirectional and bidirectional LSTMs, resulting in better performance. This technique shows promise in accurately recognizing dynamic gestures through non-contact gestures, which has important implications in various fields, including robotics, virtual reality, and human-computer interaction. On the other hand, this method is more time-consuming compared to other models.

dos Santos et al. [7] presented an innovative approach for dynamic gesture recognition consisting of two primary stages: preprocessing and classification. In the preprocessing phase, they transformed the star representation of each input video into an RGB image, enabling more efficient processing. For gesture classification, a set of convolutional neural networks (CNNs) was utilized to train a classifier for dynamic gestures. The preprocessed images were then fed into two pre-trained CNNs, and the results were weighted using a soft attention mechanism before being fed into a fully connected layer. Finally, the softmax classifier determined the class of the gesture. This approach is a significant improvement over previous methods, as it simplifies the input processing and classification tasks while achieving higher accuracy. The use of pre-trained CNNs and a soft attention mechanism allows for faster training and better recognition of dynamic gestures, with fewer computational resources required. The approach achieved an accuracy of 94.58%. Such result reaches the state-of-the-art when considering this dataset and only color information. For GRIT dataset, our proposal achieves more than 98% of accuracy.

Almasre and Al-Nuaim [8] developed a novel approach for recognizing dynamic sign language sentences by utilizing the Kinect as a sensor. Their method is referred to as the dynamic prototype model (DPM). The DPM employs three different algorithms, namely K-nearest neighbors (KNN), support vector machine (SVM), and random forest (RF), with different parameter values. After conducting various tests, it was observed that the SVM model exhibited the most precise recognition accuracy rates for difficult words.

In their research, Ertugrul et al. [9] suggested a novel gestural interface that leverages Finite State Machine (FSM) technology to allow users to personalize GUI activities. This is achieved by modifying gesture-specific parameters, including the distance from the camera, distance between hands, and timing of events. To recognize gestures and extract their properties, the authors employed the RealSense SDK. By leveraging the extracted properties, static gestures can be triggered and executed as dynamic gestures, resulting in a more natural and intuitive interaction with graphical user interfaces (GUIs). Additionally, the authors enhanced the overall efficiency, convenience, and user experience of the gesture-based GUI by integrating supplementary functionalities. This novel approach has significant potential for enhancing human-computer interaction and enabling more accessible and intuitive control of GUIs. However, this approach does not adequately support a wider range of hand gestures with increased complexity, particularly when it comes to mobile interfaces.

Lee et al. [10] suggesteda new gesture-based interface that can identify five distinct gestures within operating room (OR) environments without the need for skeleton data from Leap MotionTM or personal baseline training. By comparing various deep learning algorithms, DCNN and CapsNet, the study found that CapsNet exhibited exceptional performance in accurately detecting intricate hand movements. This implies that CapsNet could be an ideal touchless interface for monitoring clinical applications in the OR. The gesture-based interface has the potential to improve clinical practices in ORs by allowing surgeons and medical staff to interact with equipment and devices without physically touching them. This approach has several potential benefits, including a significant reduction in contamination risks. Moreover, the study's findings indicate that the use of CapsNet as the primary gesture recognition model could lead to more accurate and efficient detection of hand gestures in real-time.

To address the challenge of detecting continuous gestures on a large scale, Mahmoud et al. [11] put forward a recognition method that utilizes both depth and gray-scale input images. The mechanism consisted of two distinct steps. The initial stage involved segmenting the uninterrupted sequences of gestures into individual gestures, which was achieved through mean velocity data obtained from deep optical flow estimates. Subsequently, for each isolated gesture, deep signature features were extracted to represent the movement's position and orientation.

Meghana et al. [12] designed a robotic vehicle that can be controlled through voice commands and gestures, with an Android smartphone serving as the core component. The system has potential applications for aiding individuals with disabilities as well as for industrial use cases.

Chen and Koskela [13] proposed a novel approach for dynamic HGR using the Intel RealSense sensor. Their method involved extracting both finger motion and global features from the skeleton sequence of two datasets, DHG-14/28 and SHREC'17. The finger motions were represented using the relative positions of the finger joints, and the global features were derived from the hand orientation and distance to the camera. The authors employed the LSTM recurrent network to predict the class of input gestures. This approach demonstrated high accuracy in real-time gesture recognition and showed great potential for practical applications in human-computer interaction. Experiments demonstrate that MFA-Net achieves comparable uperformance with state-of-the-art methods on the public DHG-14/28 dataset and best performance on SHREC’17dataset.

Sarkar et al. [14] launched an innovative hand gesture recognition system that relies on depth information gathered by a time-of-flight system and operates in real-time. The method involves training and recognition through the use of a deep LSTM network known as D-LSTM. To assess the effectiveness of their system, the researchers utilized a publicly accessible dataset, achieving impressive accuracy in real-time processing. As a result, their approach is adaptable to various applications.

Zhang et al. [15] provided an innovative approach to gesture recognition that combines a hand shape adaptive algorithm with an effective area ratio calculation. The authors collected data and categorized the samples into different groups based on the shape of the subjects' palms, using this information to train their algorithm. The smallest bounding rectangle was then used to compute the gesture's effective-area ratio, improving accuracy. The initial gesture was detected using the effective-area ratio feature approach. Moreover, the number of detectable gestures was limited.

Zhang et al. [16] offered an improved version of the HU moment classic for recognition by modifying the characteristic values of the HU moment and calculating the similarity between the template and input image. The approach also utilized Kinect sensors to capture gesture information and extract hand contour information for tracking important palm modes.

In hand gesture recognition, researchers face the challenge of creating a robust framework that delivers accurate and dependable results while overcoming common limitations like illumination changes, background issues, and multi-gesture detection. Non-machine learning algorithms are also used, but they are less adaptable and accurate than machine learning approaches, which have been preferred in recent studies. Despite this, real-time processing of hand gestures is still subject to some limitations, for example distance range and lighting changes. The Table 1 below provides a summary of the different approaches to gesture recognition based on the type of sensor and method used.

Table 1. An overview of the various features of gesture recognition

Related Work	Methods	Sensor	Year	Case Study
Almasre and Al-Nuaim	SVM, RF, KNN	Kinect	2020	Yes
Ameur et al.	BU LSTM	Leap Motion	2020	Yes
dos Santos et al.	CNNs	_	2020	Yes
Mahmoud et al.	SpyNet	_	2020	Yes
Zhang et al.	Hand-type Algo	_	2022	Yes

3. PROPOSED METHODOLOGY

3.1 System overview

To enhance the development of gesture systems for various graphical interfaces, we introduce A Deep Learning Framework for Hand Gesture Recognition and Multimodal Interface Control (DL-HGRMIC) as depicted in Figure 1. The approach outlines key techniques and describes the roles in a gesture project, consisting of three main phases: identification, specification, and realization. Each phase is composed of multiple steps detailing the artifacts to be delivered and recommending suitable techniques [17-19].

Identification phase: The identification phase of designing touch gesture recognition user interfaces involves three approaches: (i) a top-down approach for native gesture applications that provides a list of actions to be implemented; (ii) a bottom-up approach for analyzing existing systems to identify low-level actions; and (iii) a hybrid approach (middle-out) that consists of goal-based gesture modeling and aims at determining the most important gestures and actions. These approaches take into consideration the three factors of usability, collection of actions, and low physical effort to ensure that the gestures are intuitive and easy.

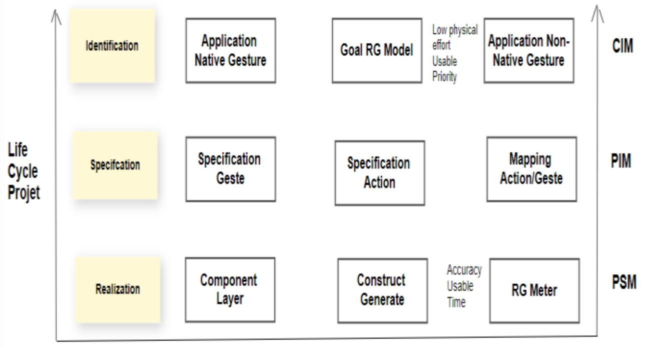

Figure 1. Overview f dl-hgrmic

Specification phase: This phase is in charge of interpreting the detected hand gestures and translating them into specific actions that can be seamlessly used to control and navigate the corresponding application, as shown in Figure 2. To identify multiplicity in the mappings between gesture and action, the context should be considered [20, 21].

When developing native or non-native gesture applications, the AOP paradigm is often adopted to manage cross-cutting concerns. AOP, short for Aspect-Oriented Programming, is a powerful software development paradigm that provides a way to modularize concerns into separate units called aspects. This helps isolate and manage code that would otherwise be scattered throughout the system, making it more maintainable, reusable, and flexible. In the context of gesture applications, the AOP paradigm can be represented by the following action formula:

Action= {ID-Gesture, Context, Aspect}.

In this notation, ID-Gesture represents the identifier of the gesture, Context refers to the name of a context, and Aspect denotes the aspect related to this context.

Realization phase: During the realization phase, architectural decisions and high project risk factors are validated through technical feasibility exploration and extensible prototypes developed early on. All DL-HGRMIC steps are performed iteratively and incrementally. The realization phase proposes a backend architecture, as shown in Figure 2.

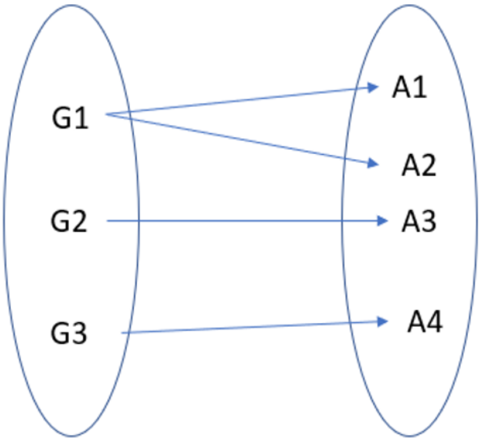

Figure 2. Mapping between action and gesture

3.2 Backend architecture

Our back-end system has been developed using a combination of Python, PyAutoGUI, OS Module, and OpenCV libraries to achieve accurate and efficient hand gesture recognition. The system is divided into four modules: camera, hand typology, detection, and action/gesture mapping, as shown in Figure 3.

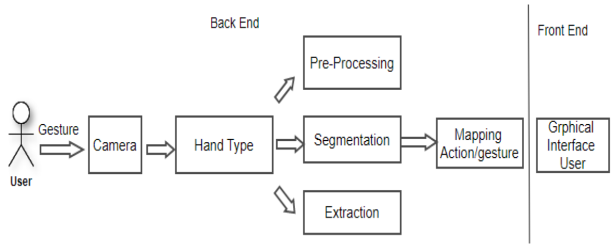

Figure 3. Backend architecture

Our system consists of five steps, each of which is explained in detail:

Camera module: The camera module is the first step in the GRS. It captures input through various image detectors and is responsible for collecting the input images based on hand The module then sends the images to the next step.

Module Hand Type: The Hand Type Module is responsible for categorizing hand measurements into six types: (XS), small (S), medium (M), large (L), (XL), and (XXL). This categorization allows the system to use fewer resources and achieve a faster response time by using specific datasets for each category. The length of one's palm is shown in Figure 4.

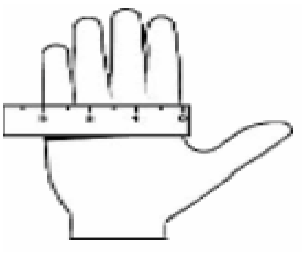

Figure 4. The length of one's palm

The six groups of subjects were determined by performing a weighting calculation using Table 2.

Table 2. Hand parameters

	XS	S	M	L	XL
The width of hand (cm)	5.08-6.35	6.35-7.62	7.62-8.89	8.89-10.16	10.16-11.43

Pre-Processing: To analyze and interpret the images captured by the camera module, the proposed system uses a region of interest (ROI) approach. This approach involves selecting only the important area of the image instead of the entire frame, which helps reduce computation time. Additionally, the system converts the selected region into a grayscale image to increase processing efficiency.

Hand Region Segmentation: To isolate the hand from the background, a technique known as background subtraction was employed. This involved obtaining a dependable background model using the running average principle. The system captured a particular scene for at least 20 frames, during which the running average was calculated for each frame, including the current and preceding ones [22]. To perform the background subtraction, the system placed the hand in front of the camera and calculated the absolute difference between the running average background and the current frame, as depicted in Figure 5.

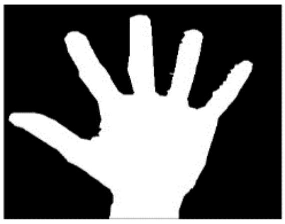

Figure 5. The result of the hand region segmentation process

Extraction and Recognition: We propose a method for extracting gesture features from images of hands using the area-perimeter ratio as shown in Figure 6. The area-perimeter ratio (C) of a gesture can be calculated using the following formula:

$$C = S / L \quad (1)$$

where, S is the area of the gesture and L is the perimeter of the gesture. The perimeter (L) can be calculated as the sum of the distances between adjacent pixels along the boundary of the gesture, which can be expressed mathematically as:

$$L = \sum f(x, y) \quad (2)$$

where, $f(x, y)$ Eq. (1) if the pixel (x, y) is on the boundary of the gesture and 0 otherwise.

Similarly, the area (S) can be calculated by counting the number of pixels within the boundary of the gesture, which can be expressed mathematically as:

$$S = \Sigma q(x, y) \quad (3)$$

where, $q(x, y)$ Eq. (1) if the pixel (x, y) is within the boundary of the gesture and 0 otherwise.

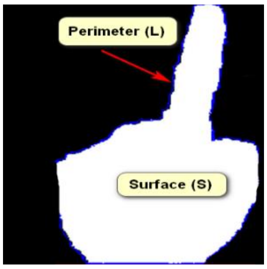

Figure 6. Gesture feature

4. RESULT AND DISCUSSION

This section presents our approach used throughout the experiments, including the implementation and training of the architecture, as well as the evaluation of the results. Two different experiments were conducted: the first focused on implementing the gesture recognition system, while the second aimed to evaluate our approach.

4.1 Implementation and training

Our approach involves three phases to identify gestures, specify actions, and implement the gesture recognition system for various applications such as VLC Media Player and Internet Explorer. In the identification phase, we conducted a preliminary study of the intricate organization involved in recognizing gestures using a legacy system. Real-time hand gesture recognition was achieved by capturing images from a webcam. The system can recognize various types of gestures, as shown in Table 3.

Table 3. The list of gestures

Name	Gesture
Palm	✋ or 🤚
Fist	✊
Thumbs Up	👍
Thumbs Down	👍
Index Right	👉

In the specification phase, we elaborated on the actions and mapped the action/gesture components identified in the previous phase. Figure 7 illustrates this process.

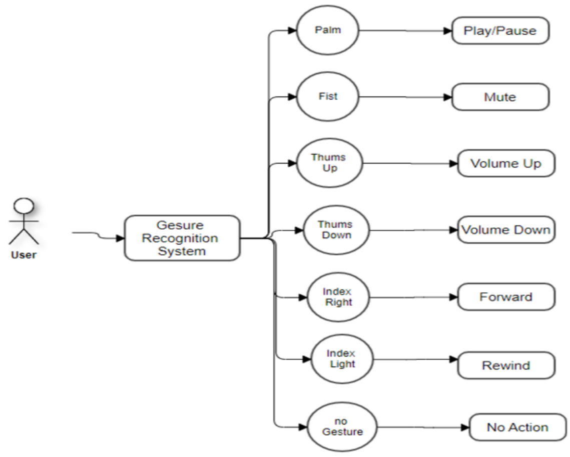

Figure 7. System design flow

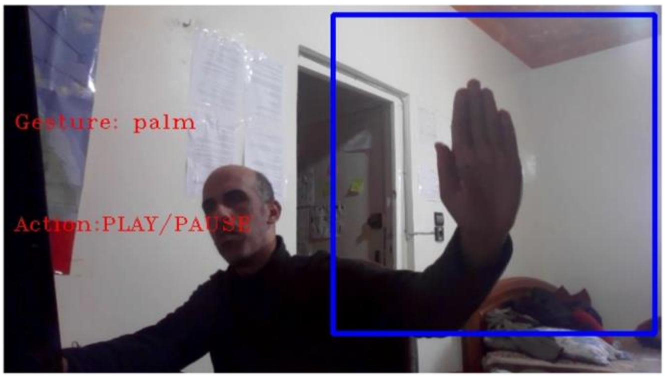

Figure 8. Symbol palm for initialize our application

For the realization phase, we utilized a Windows 10 PC with an Intel Core i7-10850H CPU operating at 2.7GHz and 32GB of RAM. Our goal was to revolutionize computer interaction by enabling touch-free and remote-free control through hand gestures. Our solution leveraged cutting-edge technologies, including OpenCV for image processing, a 2D Convolutional Neural Network for feature extraction and classification, and the PyAutoGUI library for integrating keyboard commands with our intuitive user interface built using the Streamlit web framework.

Figure 8 illustrates the Symbol Palm gesture, which serves as a means to initialize or halt our application.

To increase the volume, the user must perform a Thumbs Up gesture as shown in Figure 9 to confirm their selection.

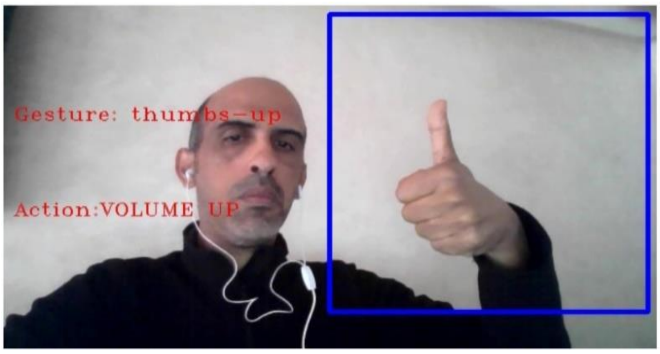

Figure 9. Symbol thump for volume up

4.2 Performance evaluation

To evaluate the user experience with our application, we conducted tests with approximately 4 different users who were asked to provide objective feedback through a questionnaire. The questionnaire included questions such as the overall experience with the media control application using gesture recognition, the ease of use, responsiveness to gestures, physical discomfort, and the application's improvement in accessibility for people with physical limitations.

Based on the objective feedback received, we identified four factors for describing and evaluating our approach:

- Usability: The gestures must be intuitive and enable effortless control of the existing GUI, ensuring ease of use and comfort.

- Response time: Users reported discomfort caused by delays in gesture recognition, especially during rapid transitions between different gestures. Improving responsiveness was identified as crucial.

- Accuracy: The accuracy of gesture recognition was considered important for the application's performance.

- Range of actions: Users expressed satisfaction with the offered range of actions.

Figure 10 depicts the results of our user study, indicating an overall positive user experience, efficient design, comfort, and low physical effort required and between 2/4 and 3.5/4.

In this study, six gestures commonly used in life were selected for recognition experiments, as shown in Table 3. In this experiment, 40 subjects were reselected according to the above selection rules. The experiment was conducted under the conditions of stable illumination, less noise, and no face appears in the picture. For each subject, five experiments were performed for each gesture (60 experiments for each gesture) at distance (the distance from the camera): 40cm.

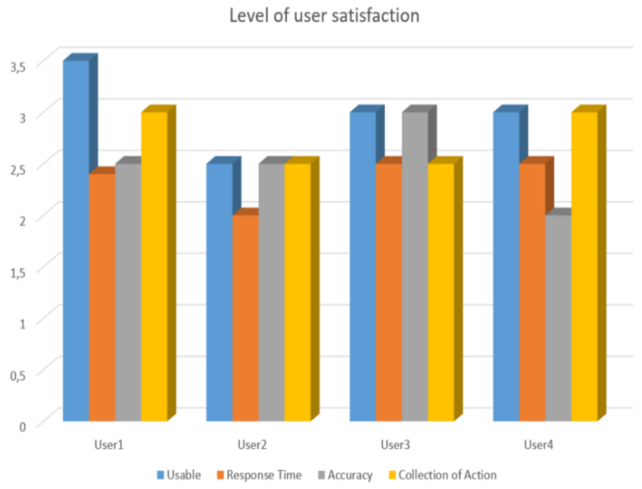

Figure 10. The results of our user study

The experimental results of the system are summarized in Table 4. The system achieved a recognition rate of 94%, with the highest recognition rate of 100% achieved under clear background and medium lighting conditions.

Table 4. Hand gesture recognition rate

Name	Gesture	N° of Input	N° of Recognized	Rate %
Palm	🖐 or ✋	60	54	90
Fist	✊	60	60	100
Up	👍	60	60	100
Down	👎	60	60	100
Right	👉	60	54	90
Left	👈	60	54	90

To demonstrate the innovation and benefits of our system, we conducted comparative experiments to evaluate its performance in terms of accuracy, real-time response, recall, and precision. By comparing our algorithm with other similar design concepts, we observed an improvement of nearly 3% in the overall accuracy rate. When compared with three other excellent algorithms under the same experimental conditions, our algorithm achieved a slightly higher recognition rate. To further validate our results, we conducted experiments using Hu-moment algorithms and deep learning, and the performance was comprehensively compared and presented in Table 5 and Figure 11.

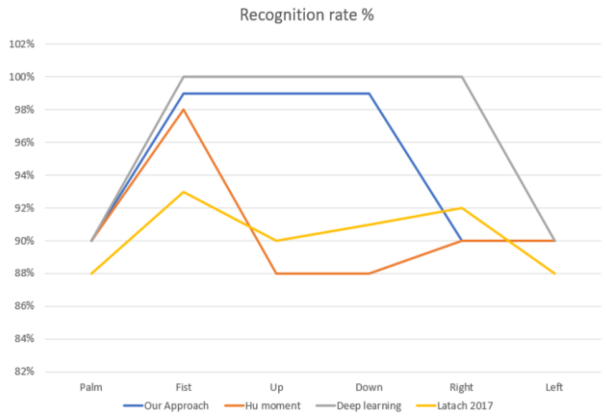

Figure 11. Comparison of recognition rates

Table 5. Further comparaison in term of response time, accuracy, recall and precision

Methods	Accuracy	Speed of Response(s)	Recall	Precision
Deep learning [23]	95.90	0.088	-	-
Hu moment [16]	90.55	0.076	-	-
Our approach	94.0	0.058	95.0	96.0

5. CONCLUSIONS

The present study introduces a software engineering approach, which facilitates the creation of end-to-end gesture recognition solutions. This method is divided into three phases: identification that identifies the gestures, specification which elaborate the action and mapping action/gesture, and realization which implement the layer to control graphical user interface. Additionally, the system will help prevent the spread of COVID-19 by eliminating the need for human interaction and dependency on devices to control the computer.

In this research had several limitations. The number of subjects and gestures was relatively small.

For further, we will extend our approach to support a larger number of hand gestures with higher complexity and adapt the approach to mobile interfaces.

REFERENCES

[1] Qian, K., Niu, J., Yang, H. (2013). Developing a gesture based remote human-robot interaction system using kinect. International Journal of Smart Home, 7(4): 203-208.

[2] Huang, Y., Yang, J. (2021). A multi-scale descriptor for real time RGB-D hand gesture recognition. Pattern Recognition Letters, 144: 97-104. https://doi.org/10.1016/j.patrec.2020.11.011

[3] Chen, X., Wang, G., Guo, H., Zhang, C., Wang, H., Zhang, L. (2019). MFA-net: Motion feature augmented network for dynamic hand gesture recognition from skeletal data. Sensors, 19(2): 239. https://doi.org/10.3390/s19020239

[4] Shen, X., Zhu, F., Sun, Z., Zhao, S. (2021). Research on bone age automatic judgment algorithm based on deep learning and hand x-ray image. Journal of Medical Imaging and Health Informatics, 11(1): 156-161. https://doi.org/10.1166/jmihi.2021.3443

[5] Dayal, A., Paluru, N., Cenkeramaddi, L.R., Yalavarthy, P.K. (2021). Design and implementation of deep learning based contactless authentication system using hand gestures. Electronics, 10(2): 182. https://doi.org/10.3390/electronics10020182

[6] Ameur, S., Khalifa, A.B., Bouhlel, M.S. (2020). A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion. Entertainment Computing, 35: 100373. https://doi.org/10.1016/j.entcom.2020.100373

[7] dos Santos, C.C., Samatelo, J.L.A., Vassallo, R.F. (2020). Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation. Neurocomputing, 400: 238-254. https://doi.org/10.1016/j.neucom.2020.03.038

[8] Almasre, M.A., Al-Nuaim, H. (2020). A comparison of Arabic sign language dynamic gesture recognition models. Heliyon, 6(3): e03554. https://doi.org/10.1016/j.heliyon.2020.e03554

[9] Ertugrul, E., Li, P., Sheng, B. (2020). On attaining user-friendly hand gesture interfaces to control existing GUIs. Virtual Reality & Intelligent Hardware, 2(2): 153-161. https://doi.org/10.1016/j.vrih.2020.02.001

[10] Lee, A.R., Cho, Y., Jin, S., Kim, N. (2020). Enhancement of surgical hand gesture recognition using a capsule network for a contactless interface in the operating room. Computer Methods and Programs in Biomedicine, 190: 105385. https://doi.org/10.1016/j.cmpb.2020.105385

[11] Mahmoud, R., Belgacem, S., Omri, M.N. (2022). Deep signature-based isolated and large scale continuous gesture recognition approach. Journal of King Saud University-Computer and Information Sciences, 34(5): 1793-1807. https://doi.org/10.1016/j.jksuci.2020.08.017

[12] Meghana, M., Kumari, C.U., Priya, J.S., Mrinal, P., Sai, K.A.V., Reddy, S.P., Vikranth, K., Santosh Kumar, T., Panigrahy, A.K. (2020). Hand gesture recognition and voice controlled robot. Materials Today: Proceedings, 33: 4121-4123. https://doi.org/10.1016/j.matpr.2020.06.553

[13] Chen, X., Koskela, M. (2014). Using appearance-based hand features for dynamic RGB-D gesture recognition. In 2014 22nd International Conference on Pattern Recognition. IEEE, pp. 411-416. https://doi.org/10.1109/ICPR.2014.79

[14] Sarkar, A., Gepperth, A., Handmann, U., Kopinski, T. (2017). Dynamic hand gesture recognition for mobile systems using deep LSTM. In Intelligent Human Computer Interaction: 9th International Conference, IHCI 2017, Evry, France, December 11-13, 2017, Springer International Publishing. Proceedings, 9: 19-31. https://doi.org/10.1007/978-3-319-72038-8

[15] Zhang, Q., Xiao, S., Yu, Z., Zheng, H., Wang, P. (2021). Hand gesture recognition algorithm combining hand-type adaptive algorithm and effective-area ratio for efficient edge computing. Journal of Electronic Imaging, 30(6): 063026. https://doi.org/10.1117/1.JEI.30.6.063026

[16] Zhang, T., Gao, X., Li, J. (2020). The improved hu moment and its application in gesture recognition. In 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), pp. 577-580. https://doi.org/10.1109/CVIDL51233.2020.00-24

[17] El Magrouni, I., Ettaoufik, A., Siham, A., Maizate, A., Lotfi, B. (2022). Approach for the construction of gestural interfaces to control graphical interfaces based on artificial intelligence. 2022 9th International Conference on Wireless Networks and Mobile Communications (WINCOM), Rabat, Morocco, pp. 1-6. https://doi.org/10.1109/WINCOM55661.2022.9966424

[18] El Magrouni, I., Ettaoufik, A., Siham, A., Maizate, A. (2021). Approach for improving user interface based on gesture recognition. EDP Sciences, 297: 01030. https://doi.org/10.1051/e3sconf/202129701030

[19] El Magrouni, I., Ettaoufik, A., Siham, A., Maizate, A. (2023). Using a gesture recognition modeling approach to improve graphical user interface. In ARPN Journal of Engineering and Applied Sciences, pp. 381-392. https://doi.org/10.59018/022358.

[20] Li, C., Hou, Y., Wang, P., Li, W. (2017). Joint distance maps based action recognition with convolutional neural networks. IEEE Signal Processing Letters, 24(5): 624-628. https://doi.org/10.1109/LSP.2017.2678539

[21] Wu, D., Shao, L. (2014). Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 724-731.

[22] Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster r-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28.

[23] Wu, X.Y. (2020). A hand gesture recognition algorithm based on DC-CNN. Multimedia Tools and Applications, 79: 9193-9205. https://doi.org/10.1007/s11042-019-7193-4