Gesture Controlled System to Automate Shutdown, Screenshot and Volume Toggle

Prisha Bhosale[1], Ananya Dandekar[2], Ria DCosta[3], J. Aishwarya[4], Shagufta Rajaguru[5]

Dept. of Computer Engineering

Agnel Charities Fr. C Rodrigues Institute of Technology, Vashi, Navi Mumbai, Maharashtra, INDIA

[1] 1022118@COMP.fcrit.ac.in, [2]1022131@COMP.fcrit.ac.in, [3]1022132@COMP.fcrit.ac.in, [4]1022152@COMP.fcrit.ac.in, [5]shagufta.r@fcrit.ac.in

***Abstract*- Across 33 rich countries, only 5% of the population has high computer-related abilities, and only one third of people can complete medium-complexity tasks. Research shows that on dividing people based on their ability to use technology into 4 groups, 14% of the population still falls under the 'below level 1' category i.e., [1] can perform a maximum of one simple task on a computer. Keeping these users, along with any population with debilitations in mind, the need arises to automate certain tasks and simplify them, to ensure a steady growth in the number of efficient users of technology. A system working based on gestures, automating mundane and repetitive tasks is a step in the right direction to facilitate a technologically better trained and equipped environment, as well as improving time complexity in the fast-paced world of today. In this paper, a prototype for a gesture enabled shutdown is proposed. The system proposed is designed to build a framework for future systems based off of this skeleton, ensuring ease of use for every user.**

***Index Terms* -Debilitations, time complexity, gesture-enabled, shutdown, framework**

I. INTRODUCTION

Gesture control provides an intuitive and accessible way for human-computer interaction, especially beneficial for individuals with disabilities or limited mobility. It offers an alternative means to engage with technology without relying on traditional input devices. Hand recognition systems can interpret gestures to control software, enable text/speech input for those with speech impairments, and promote independence by reducing reliance on assistants. This technology also aids the elderly by allowing convenient gesture-based control of daily computing tasks. Beyond personal use, gesture control has applications in healthcare for monitoring fitness/health conditions through gesture detection. The goal is to enhance accessibility, implement touchless solutions across various settings like medical environments, and streamline tasks through intuitive gesture-based interaction. The proposed project utilizes OpenCV and Python to create gesture recognition software using a webcam for input, executing actions based on detected gestures to improve the user experience.

MOTIVATION AND ORGANIZATION

In a rapidly evolving technological landscape, demand for intuitive interfaces grow. Gesture control technology offers a solution, providing accessibility and immersion.

1. Intuitive Interaction: Gesture control enables natural interactions, broadening access to digital systems for all users.
2. Enhanced User Experience: Integration of gesture control revolutionizes user experiences, especially in entertainment, by enhancing immersion and engagement.
3. Novelty and Innovation: Incorporating gesture control signifies technological advancement and innovation, setting products apart and capturing public interest.
4. Multitasking and Efficiency: Gesture control streamlines tasks, enabling quick actions without disrupting workflow, thus enhancing efficiency.
5. Reduced Equipment Wear: Implementing gesture control extends device lifespan and reduces maintenance costs by minimizing wear and tear on conventional input mechanisms.

II. EXISTING SYSTEMS

The study reported in [2] proposes a hand gesture recognition system using a combination of fuzzy algorithm and RDBMS. The system aims to enhance the traditional data entry pattern, limited to wired communications, to the universal environment using wireless communication interfaces.

The proposed system utilizes the 5th Data Glove System, a popular input device in the haptic application field, to capture hand gestures. The system incorporates a fuzzy reasoning process and efficient recognition model using RDBMS, enabling real-time recognition of continuous and dynamic gestures. The limitations of using the 5th Data Glove System as the input device, including its accuracy, reliability, and potential constraints, are not discussed, so it can be assumed that however well-trained be the model, physical conditions may hamper the output.

In [3] Hand gesture recognition is an active area of research in computer vision and machine learning, with applications in human-computer interaction (HCI). Vision-based gesture recognition systems have been developed to solve specific problems and work in a particular manner. The proposed solution, Gesture Learning Module Architecture (GeLMA), allows for the definition of commands based on static and dynamic gestures, and can be easily integrated into various applications. The experiments showed that the system was able to reliably recognize vowels in real-time and can be extended to recognize the rest of the alphabet. The proposed framework provides a generic and solid foundation for the development of hand gesture recognition systems in any HCI application. The limitations of the proposed framework in terms of computational requirements, real-time performance, and robustness to varying lighting conditions and hand orientations are not discussed in detail in the paper.

A viable and popular solution for improving human-computer interaction through hand gesture recognition is given in [4] that offers a simulation using OpenCV and Python 2.7, which are widely used and accessible tools for developers. It utilizes a histogram-based approach to separate the hand from the background image, which can lead to accurate hand gesture recognition i.e. Implements background cancellation techniques to produce optimum results, enhancing the accuracy of the hand gesture recognition system. Processes and models the detected hand by finding contours and convex hull, enabling the recognition of finger and palm positions and dimensions. Overall, the paper provides a practical and effective approach to hand gesture recognition using readily available tools and techniques, making it a valuable resource for researchers and developers in the field. The paper does not mention the specific dataset used for training and testing the hand gesture recognition system, which could affect the generalizability of the results.

In the study presented in [5] the implementation is divided into four main steps: Image Enhancement and Segmentation, Orientation Detection, Feature Extraction and Classification. While this study concentrates on the aforementioned four categories, a significant challenge arises from the swift colour shifts that occur due to varying lighting conditions, potentially leading to inaccuracies or complete breakdowns in the system. The main drawback of the system is it does not consider gesture recognition of the motion of gestures, and it is unable to classify images with complex backgrounds. Furthermore, the edge detection and segmentation algorithms used here are not very efficient when compared to neural networks. The dataset which is considered here is very small and can be used to detect selected few gestures.

The Hand contour-based neural networks training in [6] is evidently faster than complex moments-based neural networks training (at least 4 times faster where hand contour based neural network took roughly between 110 and 160 epochs to converge, whereas complex moment-based neural network required at least between 450 and 900 epochs to convergence). This suggests that the hand contour method is more suitable than the complex moments method in real-world applications that need faster training, such as online training systems. Complex moments-based neural networks (86.37%) proved to be more accurate than the hand contour-based neural networks (71.30%). In addition, the complex moments-based neural networks are shown to be resistant to scaling (96.15%) and translation (100%), and to some extent to rotation (65.38%) in some gestures (for example: open (100%), Maximum (80%). The results of the study indicate that the complex moments approach outperforms the hand contour method due to its greater accuracy, particularly in applications like desktop and offline training where training speed is not as important. Hand contour features are less distinguishable compared to complex moments features. The high number of unrecognized cases predicted via the hand contour method makes this evident (11.90%) of the testing cases for hand contour against (1.20%) for complex moments).

In [7] we talk about the implementation of a novel 7-channel sEMG armband, which can be employed as HMI for gaming control and rehabilitation support. It is based on the bio-inspired technique known as ATC, which allows the device to be low-power and to reduce the complexity of the input space for classification tasks. One advantage is that accuracy is higher due to reading of muscle motion i.e., the armband reads the motion of the muscle involved in performing a particular gesture. When the number of features or the dimension of the feature vector, which is equal to the number of nodes in the input layer, is moderate, neural networks seem to operate more efficiently. (e.g., the complex moments method with 10 nodes is more accurate than the hand contour method with 1060 nodes). Future perspectives for the designed armband involve the need and interest to test it with further classification algorithms, such as SVM, DT or KNN, to analyse which is the most suitable solution. Moreover, we are planning to develop a serious game to validate the use of our device in an application scenario besides evaluating its classification performance.

A. Research Gaps

1. Focus on specific input devices (e.g., 5th Data Glove, sEMG Armband), with limited discussions on their limitations and potential alternatives.
2. Limited discussion on the robustness of the proposed systems in real-world scenarios, considering factors like varying lighting conditions, complex backgrounds, and user-specific physical conditions.
3. Lack of detailed documentation on the datasets used for training and testing.

B. Broad areas that need to be worked upon

1. Integration of multiple input devices, exploring their synergies and compensating for individual limitations.
2. Conduct comprehensive real-world testing to evaluate the systems' performance under diverse conditions.
3. Clear documentation of the datasets used in experiments, and exploration of the performance of the systems with diverse datasets.

III. IMPLEMENTED SYSTEM

A. System design

The implemented system integrates computer vision and machine learning techniques to enable real-time hand gesture recognition for triggering system actions. As seen in Figure 1, the system commences with capturing the hand gestures performed by the user and the program accesses the camera to capture video frames. The detected gestures are analysed with the help of machine learning models. Upon successful recognition of a gesture, the system proceeds to verify whether the detected gesture corresponds to any predefined gestures stored within the system's database. A gesture if matched, the corresponding action associated with it is executed. The actions associated with the gestures include automated processes such as system shutdown, capturing a screenshot, adjusting the volume levels (increasing and decreasing).

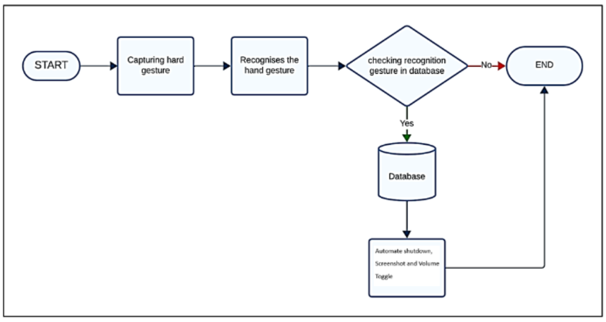


Fig. 1. Block Diagram of the Implemented System

Fig. 2. Use Case Diagram of the Features of System

B. Implementation details

The software components utilized in the development of this system are Python, PyCharm, Jupyter Notebook, Anaconda, Mediapipe, OpenCV and Tensorflow. Leveraging OpenCV for image processing, the system captures video frames from a camera, which are then processed by the MediaPipe library for hand tracking, identifying hand landmarks in each frame. The system loads models and labels required for hand tracking and classification. The KeyPointClassifier and PointHistoryClassifier are the custom models for classifying hand gestures based on keypoints and point history, respectively. A class named CvFpsCalc is instantiated for calculating frames per second (FPS). Subsequently, the landmark data is forwarded to a TensorFlow-based gesture recognition model, allowing the system to classify and recognize gestures in real-time. Upon detection of a specific gesture indicative of a system action, such as a shutdown gesture, the system utilizes the subprocess module and PyAutoGUI to initiate the corresponding action, effectively enabling users to interact with the system through intuitive hand gestures. This architecture ensures seamless integration of various components for efficient and accurate gesture recognition, paving the way for intuitive human-computer interaction in diverse applications.

C. Results

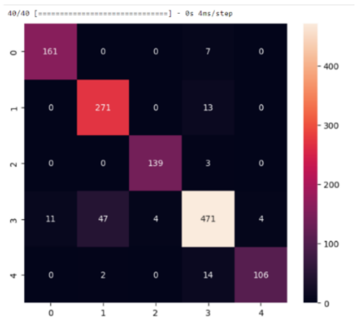
Real-time hand gesture recognition and action execution according to the gesture recognized.

1. The peace sign gesture when shown will shut down the system
2. The open palm gesture when shown will take a screenshot of the screen and display a notification indicating that a screenshot has been taken
3. The thumbs-up gesture when shown will result in an increase in the system volume
4. The thumbs-down gesture when shown will result in a decrease in the system volume
5. These functionalities are part of a hand gesture recognition system designed to interpret specific gestures and trigger corresponding actions, providing users with a hands-free interaction experience with their computing environment.

Features and Benefits of Gesture Recognition System:

1. Intuitive Interaction: Provides a natural and intuitive way to interact with computers and devices, enhancing user experience and ease of use.
2. Accessibility: Improves accessibility for individuals with disabilities or limited mobility, opening up new opportunities for inclusivity and participation.
3. Enhanced Entertainment: Enhances gaming and entertainment experiences, enabling immersive gaming, virtual reality applications, and interactive multimedia content.
4. Hygienic Control: Facilitates touchless control in healthcare settings, promoting hygiene and safety, particularly beneficial for surgeons during surgery.
5. Security Enhancement: Enhances security and access control through gesture recognition, offering secure access based on recognized gestures and improving surveillance capabilities.
6. Scalability: Designed to work with various hardware setups, offering scalability and adaptability to different environments and requirements.

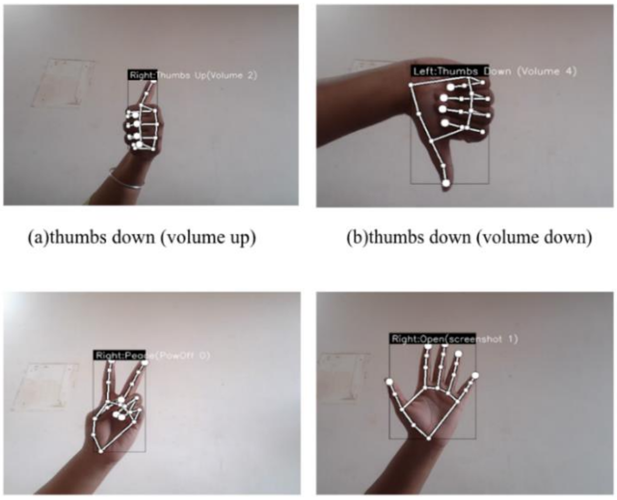
Figure 3 displays a confusion matrix and classification report, which are commonly used to evaluate the performance of a machine learning classification model across multiple classes.
The numbers 0 to 4 represent the gestures that the model has been trained to recognise: 0 being the Peace sign (Power Off) function, 1 being Open Palm (Screenshot) function, 2 being Thumbs up (Volume up), 4 being Thumbs down (Volume Down).
Number 3 is the storage for all null gestures, like an open palm supporting the user's face, or gestures used to emphasize speech. Using the NULL index we have tried to minimize misrecognition of gestures during run-time.
The main part of the image is a heatmap grid, where each cell represents the number of instances classified into a particular combination of predicted class (columns) and actual/true class (rows). The diagonal cells from top-left to bottom-right show the correctly classified instances for each class, while the off-diagonal cells indicate misclassified instances.
Below the heatmap, there is a table that provides various performance metrics for each class, such as precision, recall, f1-score, and support.
The bottom rows of the table show overall metrics for the entire dataset, including accuracy, macro average, and weighted average.
Based on the values in the heatmap and the performance metrics, we can assess the model's performance in classifying instances into different classes. Higher values along the diagonal and high precision, recall, and f1-scores indicate good performance, while off-diagonal values and lower scores suggest potential misclassifications or areas for improvement.



Classification Report

	precision	recall	f1-score	support
0	0.94	0.96	0.95	168
1	0.85	0.95	0.90	284
2	0.97	0.98	0.98	142
3	0.93	0.88	0.90	537
4	0.96	0.87	0.91	122
accuracy			0.92	1253
macro avg	0.93	0.93	0.93	1253
weighted avg	0.92	0.92	0.92	1253

Fig 3. Heatmap of proposed system



(a)thumbs down (volume up) (b)thumbs down (volume down)

(c) peace sign (power off) (d) open palm (screenshot)

Fig. 4. Gestures in Implemented System

IV. CONCLUSION

In conclusion, the implementation of hand gesture recognition for automated shutdown, screenshot, and volume toggle functionalities presents a promising and innovative approach to human-computer interaction. Through the integration of advanced computer vision algorithms and machine learning models, the system demonstrates a robust capability to accurately interpret and respond to user hand gestures. This technology not only enhances user convenience but also contributes to a more intuitive and efficient user interface base framework for many future projects involving gesture recognition and automation and reduces the time and effort in performing small and seemingly insignificant tasks. In this paper, we have attempted to gather and display the simplest and most efficient framework possible. We have reviewed multiple ideas to come to the most feasible conclusion with regards to ease of execution, ease of performance, user friendliness, speed, accuracy, portability and inclusivity.

Further research could focus on optimizing the system for different environments and conditions, such as low-light settings or noisy backgrounds. Continuously improving the accuracy and reliability of gesture recognition algorithms would enhance user experience and system performance. This could involve refining existing algorithms or exploring new approaches such as deep learning models.

V. FUTURE SCOPE

Our project aims to build the base for future implementations of gesture control. Our desire is to make this framework integrable with other systems so it can run alongside any device or platform. Taking an example of the Samsung's 'Open Palm to take a Photo' being pre-integrated within the Samsung camera interface, we aim to have the base framework (which is this project) be added as a part of the basic PC environment.

The addition of this framework to the existing and developing systems will facilitate gesture control for simple tasks, making each device significantly easier to operate for those who find it difficult to operate the existing, majorly mouse and keyboard-controlled systems.

REFERENCES

[1] "The Distribution of Users' Computer Skills: Worse Than You Think" Jakob Nielsen [Online]. Available: https://www.nngroup.com/articles/computer-skill-levels/

[2] "Hand Gesture Recognition System Using Fuzzy Algorithm and RDBMS for Post PC" Jung-Hyun Kim, Dong-Gyu Kim, Jeong-Hoon Shin, Sang-Won Lee & Kwang-Seok Hong Conference Paper. [Online]. Available: https://link.springer.com/chapter/10.1007/11540007_21

[3] "Hand gesture recognition system based in computer vision and machine learning: applications on human-machine interaction Dissertation" Paulo Trigueiros.[Online].Available: https://typeset.io/authors/paulo-trigueiros-rgjmd83kae

[4] "Hand Gesture Recognition Using OpenCV and Python" V. Harini, V. Prahelika, I. Sneka & P. Adlene Ebenezer

[5] "Real-Time Hand Gesture Recognition Using TensorFlow and OpenCV" Pratik Thakur, Rupali Borade, Shivani Jadhav, Ronak Gaikwad, Jayashree Pansare, Shubhangi Khade Department of Computer Engineering MES College Of Engineering, Pune. [Online]. Available: https://www.jetir.org/papers/JETIR2205616.pdf

[6] "Recent methods in vision-based hand gesture recognition" Haitham Badi. [Online]. Available: Recent methods in vision-based hand gesture recognition International Journal of Data Science and Analytics (springer.com)

[7] Andrea Mongardi, Fabio Rossi, Andrea Prestia, Paolo Motto Ros, Massimo Ruo Roch, Maurizio Martina, Danilo Demarchi "Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband." IEEE Transactions on Biomedical Circuits and Systems, Vol 16, Issue 6, December 2022, DOI: 10.1109/TBCAS.2022.3211424

[8] https://typeset.io/search?q=gesture%20recognition