



Gestural Interface for Networked Kinesthetic Operations (G.I.N.K.O.)

Prathamesh Tupe¹, Sakshi Pawar², Atharva Pagale³, Neenad Jadhav⁴,

Prof. Suchitra Deokate⁵

Student, Department of Computer Engineering, Dhole Patil College of Engineering Pune, Maharashtra, India¹⁻⁴

Professor, Department of Computer Engineering, Dhole Patil College of Engineering Pune, Maharashtra, India⁵

Abstract: In this research, a novel system that uses machine learning-recognized hand gestures to ease file transfer and computer control between two nearby PCs is displayed. By the use of gestures like "catch" and "throw," users may transfer files between computers and handle fundamental PC functions like media control, file browsing, and cursor movement. Bluetooth PAN (Personal Area Network) is used to establish communication between devices, allowing easy data sharing without the need for cable connections or difficult file browsing. The system uses a trained machine learning model for classification, OpenCV for image processing, and a standard webcam for gesture detection. Within an 8–10 m range, experimental results showcase reliable file transfer capability and accurate gesture detection. The goal of this initiative is to enhance computer-human interaction.

Keywords: Machine Learning, Gesture Recognition, File Transfer, Bluetooth PAN, Human-Computer Interaction, Computer Vision.

I. INTRODUCTION

Over time, human-computer interaction has changed dramatically. Natural interaction technique like voice commands is now taking the place of traditional devices like the mouse, keyboard, and touchpad. The goal of these technique is to get technology closer to human behavior and make it easier to use

In this project, we have created a system that controls a PC and wirelessly transfers data to other PCs in around it using hand gestures as input signals. Microsoft's HoloLens Hawaii model, which allows users to "catch" and "throw" virtual objects, served as the model for the concept. Using a webcam and Bluetooth PAN connection, we apply a similar idea to real-world file sharing across laptops and PCs, except this project does not need any manual interaction and the transfer of data can be executed through and with any of the operation systems.

A machine learning model that has been trained on a dataset of different hand gestures is used to recognize the gesture. The CNN model classifies the gesture into specified categories after the image is preprocessed using OpenCV. The system executes related operations, like file transfer, slide navigation, and cursor control, based on the detected motion.

In addition to making computer use easier, this method offers a touchless interface, which is particularly helpful in sanitary settings, classrooms, and presentations. Additionally, file sharing via Bluetooth PAN is convenient, secure and localized without the need for internet or Wi-Fi connectivity.

II. METHODOLOGY

Two-Phase File Transfer & Hand Gesture Control System

Phase 1: Hand Gesture Control of PC Overview:

concept is to create a dependable system that enables users to use hand gestures to control fundamental PC operations, such as cursor movement, clicks, scrolling, and simple shortcuts.

1.1: Design the Vocabulary for Gestures

Cursor Movement: Index finger extended in an open hand. The cursor is controlled by the tip of the index finger.

Left Click: Use your thumb and index finger to pinch quickly.

Right Click: Use your thumb and middle finger to pinch quickly.

Drag & Hold: A prolonged pinch between the thumb, index finger and middle finger which will be used to drag the file.

Scroll: Up/down motion of index finger as well as middle finger To move back or forth, use index as well as middle finger to swipe left or right. Minimizing the Tab: use four fingers excluding thumb swiping to the left to minimize the tab

Closing Tab: use thumb flick down gesture to close any tab



1.2: Configuring the Technology Stack Python is the language chosen for quick prototyping.

Libraries:

- i. OpenCV: To record webcam footage.
- ii. MediaPipe Hands: To continuously detect and track 21 significant landmarks on the hand.
- iii. PyAutoGUI: To convert motions into keyboard and mouse events that are shared by the entire system.

1.3: Develop the Core Gesture Engine

Hand Tracking: there is script that constantly retrieves the (x, y, z) coordinates of each of the 21 hand landmarks using MediaPipe.

Cursor Control: PyAutoGUI moves the mouse cursor after mapping the normalized coordinates of the INDEX_FINGER_TIP to your screen resolution.

Classification of Gestures:

Determine the distance between THUMB_TIP and INDEX_FINGER_TIP in order to perform pinch detection.

It is a pinch if the distance is less than a certain threshold, like 30 pixels.

A "hold" is a prolonged pinch; a rapid pinch is a click.

Swipe Detection: To identify swipes, monitor the index finger and middle finger landmark's speed and direction.

Phase 2: "Catch and Throw" File Transfer System

Overview: Aim is to put in place a user-friendly system for transferring data between Bluetooth-enabled PC's that allows users to "catch" files and "throw" them to another computer using natural hand motions.

2.1: Establish the Vocabulary for File Transfer Gestures The catch gesture is a basic catch action made after selecting a file. Throw Gesture: Throw gesture is a basic throw action.

Target Selection: The target PC is determined by the throw gesture. Cancel Transfer: To cancel the transfer user just have to open and close the palm 2 times after the catch gesture is performed.

2.2: Configuring the Technology Stack Language: Python (in line with Phase 1) Libraries:

For gesture recognition, OpenCV & MediaPipe Hands: (Already from Phase 1)

For Bluetooth connectivity, PyBluez or socket programming.

For UI feedback and transfer animations, use Pygame or Tkinter.

2.3: Engine used for File Transfer

2.3.1. Configuring Bluetooth Connectivity Connect PCs via Bluetooth PAN (Personal Area Network).

Put in place automation for device pairing and discovery.

2.4: User Interface

2.4.1. Visual Input File Caught: The highlighted area surrounding the selected file glows

Animated "energy" effect surrounding the cursor, ready to throw Transfer Progress: During the transfer, a buffering animation with the text sending will be displayed on the screen Finish: Animation of the thumbs up with the text file sent will be displayed

2.4.2. System for File Preview thumbnails for videos and pictures: document icons and essential information While choosing, the file type it's size will be shown. Phase 2 Deliverable A complete file-transfer system that allows users to: grab over files to visually "capture" them.

Using a throw motion, "Throw" files to a different Bluetooth-enabled computer. successfully move different kinds of files between machines Obtain verification before receiving the files from another PC.

Obtain verification that the transfers were successful. Combining Phase 1 with Integration To access files, Phase 1's cursor control is used.

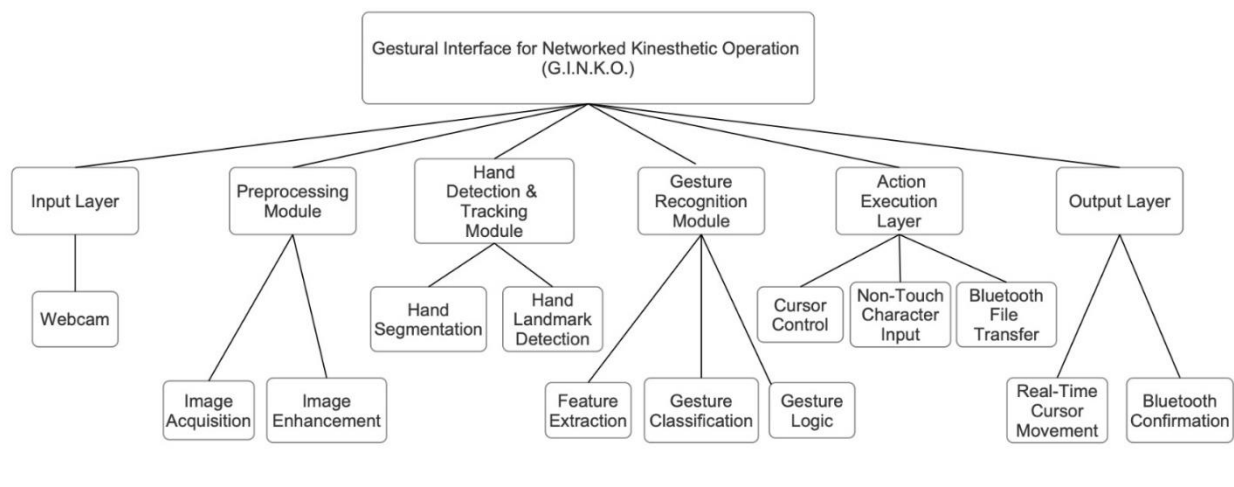
The same hand tracking pipeline is used for both transmission and control purposes.

Integrated user interface for complex file transfers and basic control

This approach enables a smooth transition from simple PC control to intuitive file transfer using the same gesture-based interaction paradigm.



III. SYSTEM ARCHITECTURE



1. Input: Live Video Feed from Webcam
2. Hand Landmark Detection: Using MediaPipe Hands library to identify 21 key points on the hand.
3. Gesture Processing & Logic:
 - Track the Index Finger Tip for cursor movement.
 - Calculate the distance between Index Finger Tip and Thumb Tip to detect a "click" gesture.
4. Cursor Control: Map the finger coordinates to screen coordinates and control the system cursor using pyautogui or similar.
5. Output: Real-time Cursor Movement and Click Actions.

IV. RESULT

The final result was the successful implementation of a gesture recognition system that used hand movements to control basic PC functions; Bluetooth PAN connectivity enabled wireless file transfer between nearby PCs; the "Catch and Throw" gesture method worked well for user-friendly media and file sharing; real time gesture detection with low latency using a webcam/sensor input; and effortless connectivity and data transfer with decent transfer speed and range.

V. CONCLUSION

The research presented here describes an intelligent and interactive system that uses machine learning to integrate Bluetooth PAN file transfer with gesture-based PC control. The potential of human-computer interaction beyond standard input devices is demonstrated by the combination of CNN for gesture categorization and Bluetooth connectivity for data transfer. The system enables a futuristic, touchless computer experience by successfully substituting basic hand movements for physical tasks like clicking, dragging, or connecting connections. It is affordable, simple to set up, and adaptive to accessibility, presentations, and home automation needs.

VI. FUTURE SCOPE

- Integration with AI/ML models for more accurate and adaptive gesture recognition.
- Implementation on mobile devices and smart TVs for cross-platform control.
- Addition of voice and gesture hybrid control for enhanced accessibility.

REFERENCES

- [1]. K. Y. Chen, D. Ashbrook, M. Goel, S. H. Lee, and S. Patel, "AirLink: Sharing Files Between Multiple Devices Using In-Air Gestures," in Proc. of ACM UIST, St. Andrews, UK, 2014, pp. 565–575. <https://mynkgoel.github.io/pdfs/airlink.pdf>



- [2]. N. Fadel and E. I. Abdul Kareem, "Computer Vision Techniques for Hand Gesture Recognition: Survey," Int. J. of Advanced Computer Science and Applications (IJACSA), vol. 14, no. 5, pp. 45–52, May 2023. DOI: 10.14569/IJACSA.2023.0140507
- [3]. A. Singh, R. Kumar, and S. Kaur, "Real-Time Hand Gesture Recognition System Using CNN and OpenCV," in 2021 IEEE Int. Conf. on Intelligent Systems and Computation (ICISC), pp. 256–260, 2021. DOI: 10.1109/ICISC52236.2021.9570284
- [4]. S. B. Patil and V. P. Deshmukh, "Human-Computer Interaction Using Hand Gestures: A Machine Learning Approach," International Research Journal of Engineering and Technology (IRJET), vol. 8, no. 6, pp. 1410–1414, June 2021. <https://www.irjet.net>
- [5]. J. Wang, Z. Liu, and J. Wu, "Gesture-Based Communication Between Multiple Devices Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 116 512–116 523, 2021. DOI: 10.1109/ACCESS.2021.3106523
- [6]. M. A. Rahman, M. N. Hassan, and T. Rahman, "A Vision-Based Hand Gesture Recognition System for Human–Computer Interaction," IEEE Trans. on Human–Machine Systems, vol. 50, no. 5, pp. 456–467, Oct. 2020. DOI: 10.1109/THMS.2020.2975632
- [7]. S. K. Ghosh and A. Dutta, "Machine Learning-Based Dynamic Gesture Detection for Smart Device Control," in Proc. IEEE ICCCNT, pp. 1–5, 2022. DOI: 10.1109/ICCCNT56476.2022.9988451
- [8]. A. Chaudhary, S. Singh, and R. Mehra, "An Efficient File Transfer Mechanism Using Gesture Recognition and ML Techniques," IJRASET, vol. 11, no. 7, pp. 312–318, July 2023. <https://www.ijraset.com>