IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Virtual Gesture Fusion (VGF): A Comprehensive review of Human – Computer Interaction through Voice Assistants and Gesture Recognition

Abhishek Dadhich[1], Himanshi Khandelwal[2], Himanshu Jhalani[3], Ayushi Mangal[4]
[1]Asst. Prof., Computer Department, Poornima Institute of Engineering and Technology, Jaipur
[2, 3, 4]Computer Department, Poornima Institute of Engineering and Technology Jaipur

***Abstract*-** VoiceGesture Fusion (VGF) is like teaching computers to understand both our voices and hand movements so we can control them better. In our day-to-day activities, wireless gadgets are increasingly prevalent, with the computer mouse being a significant advancement in human-computer interaction. Even now, Bluetooth and wireless mice remain popular tools. However, it's important to note that these wireless devices still require hardware, such as batteries for power and a dongle for connecting to the computer. This study looks at how VGF works, the problems it faces, and what might be improved in the future. By combining how we talk and move, VGF helps us interact with devices like phones or computers in easier ways. It's useful for things like games, virtual reality, healthcare, and smart homes. The system will utilize Python and OpenCV, along with other technologies. Users will view the camera feed on the system's screen for adjustments. Development of this system will involve the Python packages NumPy and mouse. Additionally, we'll employ tools such as Python, MediaPipe (a Google project), and OpenCV. MediaPipe is well-known for its efficiency and ability to offer swift solutions for AI projects. In order to facilitate more natural human-computer contact, we propose in this work to integrate a voice assistant, chatbot, and hand gesture recognition system to control the virtual mouse.

***Keywords*-** Python, OpenCV, Hand Tracing, Human-Computer Interaction, Deep Learning, MediaPipe, NLP

1. Introduction-

This research introduces a novel AI virtual mouse system that operates based on hand gestures and hand tip detection, enabling users to interact with computers through a combination of computer vision and voice commands. The primary aim of this proposed system is to enhance human-computer interaction by utilizing gestures, text input, and speech recognition [1]. In hand gesture detection, rather than using a conventional mouse, tasks such as controlling the mouse pointer and scrolling are achieved using a webcam or the computer's built-in camera. Human-computer interaction employs computer vision to recognize hand gestures and fingertip positions. Once frames are captured and processed, the webcam identifies different hand movements and fingertip gestures, enabling the execution of corresponding mouse functions [2]. As virtual assistant technologies drive artificial intelligence (AI) platforms, encompassing machine learning, speech recognition, and natural language processing, substantial data volumes are essential for their operation. Advanced algorithms embedded in AI programming facilitate learning from user data, thereby improving the system's ability to predict user needs during interactions with a virtual assistant [3]. A project named Virtual Voice Assistant leverages natural language processing and machine learning to enable device operation through voice commands. Utilized technologies include RESTful APIs, various Python libraries, TensorFlow, and Keras. The latest trend in conversational services revolves around chatbots, virtual entities equipped with interactive text capabilities enabling communication with humans. Chatbots are predominantly built using a conversational dialog engine, typically coded in Python, capable of responding based on analysis of numerous recorded conversations. Presently, chatbots are developed in various languages such as French, Spanish, and English. Encouraging users to adapt to VGF, ensuring synchronization between verbal and hand gestures, and streamlining the process pose significant challenges. Additionally, it's crucial to prioritize user privacy when employing VGF technology. The study concludes by proposing intriguing avenues for future research, such as introducing innovative interaction methods, creating situation-aware systems, and exploring new interaction strategies [6].

1.1. Problem Description:

The proposed AI virtual mouse system offers solutions to practical challenges, such as restricted space for using a physical mouse or individuals experiencing difficulty due to hand impairments. Furthermore, amidst the COVID-19 pandemic, concerns about touch-based device usage for fear of virus transmission have emerged. The AI virtual mouse addresses these concerns by utilizing hand gestures and tip detection via a webcam or integrated camera to operate PC mouse functions, providing a touch-free alternative [4].

1.2. Objective: The main objective of the proposed AI virtual mouse system is to offer an alternative to the traditional mouse setup for executing and managing mouse functions. This involves utilizing a webcam to capture hand gestures and hand tips, which are subsequently analyzed to execute designated mouse actions such as left and right clicks, as well as scrolling.

2. Related Work

1) Quam's hardware-based system, as referenced in [9], conducted a gesture recognition experiment utilizing a human hand to control the DataGlove. This experiment focused on examining twenty-two gestures classified into three categories. The first category involved solely finger flexion movements, while the second category combined hand orientation and finger flexion exercises

2) Chen-Chiung Hsieh et. Al. proposed a real-time hand gesture recognition system in their work titled "Motion History Image is Used by a Real-Time Hand Gesture Recognition System" [10]. This study introduced a technique for detecting hand movements based on motion history images and utilized a skin color model based on facial features.

3) Vantukala Vishnu et. Al. developed a virtual mouse control system utilizing Hand Gesture Recognition and Colored Finger Tips technology, as described in [11]. This technology enables cursor control without the need for sensors or direct physical contact by tracking and distinguishing colorful fingertips The virtual mouse serves various functions such as scrolling, single or double-clicking on the left, and other actions, with different configurations of colored caps employed for different operations.

4) According to B. H. Juang, as mentioned in [12], L. R. Rabiner's "An Introduction to Hidden Markov Models" highlights the significance of Hidden Markov Models (HMMs) in real-time dynamic gesture detection processes. HMMs are particularly effective in static environments and are considered practical for this purpose.. The hand gesture laptop utilized in this research, enabling various activities such as volume control and media playback management.

5) Amit M. Potdar et. Al. proposed a RADAR-based object detection system utilizing an ultrasonic sensor, as outlined in [13]. This project involved the development of a system for detecting objects using RADAR principles but employing ultrasonic sensors as a cost-effective alternative to traditional RADAR systems, which can be expensive and complex to operate.

3. Proposed System

VGF revolutionizes human-computer interaction by integrating voice commands and hand gestures, minimizing direct physical contact with the computer. Through the utilization of voice assistants and a combination of static and dynamic hand gestures, all input and output operations can be efficiently handled. This is facilitated by CNN-like models executed by MediaPipe, operating on the Pybind11 framework [5].

The VGF system consists of two modules: one focusing on hand recognition using MediaPipe Hand recognition, particularly employing gloves with a uniform color, while the other integrates a voice assistant and a chatbot. Presently, it is designed to function on Windows operating systems.

The coordinated operation of gesture-controlled virtual mouse and voice assistant characterizes the planned VGF system. Its development involved Python modules, Media Pipe, NLP, and OpenCV [7]. The following steps outline the functioning of the system.

Step 1: User input is received either through speech or gestures.

Step 2: If the input is in the form of gestures, the gesture recognition process is activated.

Step 3: Using OpenCV and Media Pipe, the code identifies and maps the hand's coordinates, referred to as landmarks. Each motion corresponds to specific hand landmarks, aiding in determining the hand's position.

Step 4: Based on the recognized motion, the system executes the intended action.

Step 5: Upon receiving a speech command, the system evaluates whether it constitutes a gesture command. If affirmative, it triggers the gesture recognition process, repeating steps 3 and 4.

Step 6: Alongside processing gesture commands, the voice assistant interprets user requests as input and responds accordingly

Voice-Gesture Fusion (VGF) is divided into two phases.

3.1 Gesture Recognition:

a. Workflow:

Through the integration of the PyAutogui module and a web camera, various mouse activities can be executed via hand gestures with assistance from a voice assistant. To enable real-time hand tracking, the initial step involves employing OpenCV, which activates the webcam to capture and monitor hand movements. Subsequently, the hand tracking module identifies landmarks on the hands and maps them accordingly [3]. Finally, PyAutogui facilitates the execution of virtual mouse operations. The hand gestures and accompanying assistance are driven by self-sufficient "Computer Vision" and "AI ML" algorithms, operating independently without requiring external hardware.

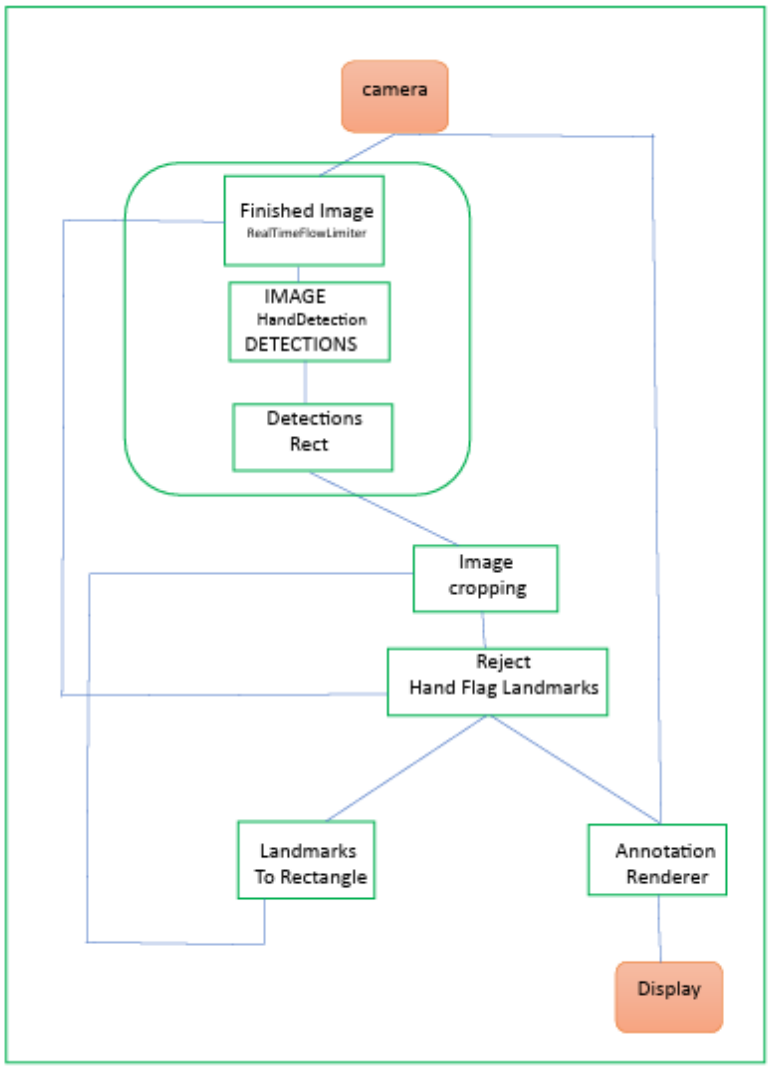


Fig.1 Mediapipe Hand Recognition Graph

b. Methodology:

1. HAND TRACKING - The Hand Tracking model relies on two core backend modules: palm detection and hand landmarks. Palm detection isolates and identifies the specific area within an image that contains a palm. Subsequently, Hand Landmarks detects 21 unique landmarks on the cropped image [4].

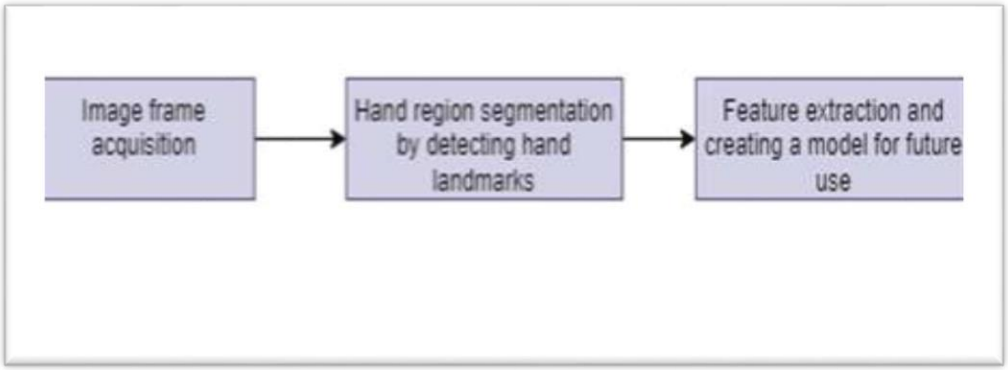


Fig. 2 Hand Tracking Model

2. FINGER COUNTER - : The Finger Counter module imports the hand tracking model, designed with easily adaptable and reusable components for enhanced efficiency during integration and utilization. Moreover, Finger Counter imports a database comprising six hand images demonstrating counts from one to five, along with an image of a fist [5].

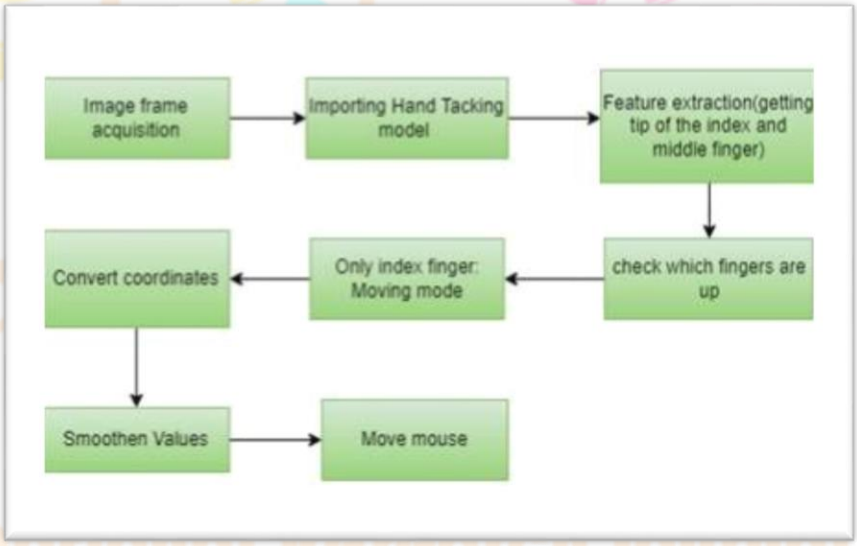


Fig. 3 Finger Counter Model

3. GESTURE VOLUME CONTROL - This model will also import Hand tracking modules. Detecting index finger and thumb to control the volume. Also detecting the pinky finger to fix the volume. Hence using 21 hand landmark features extracting the landmarks which are required and programming them accordingly to give the expected output is the aim here [8].

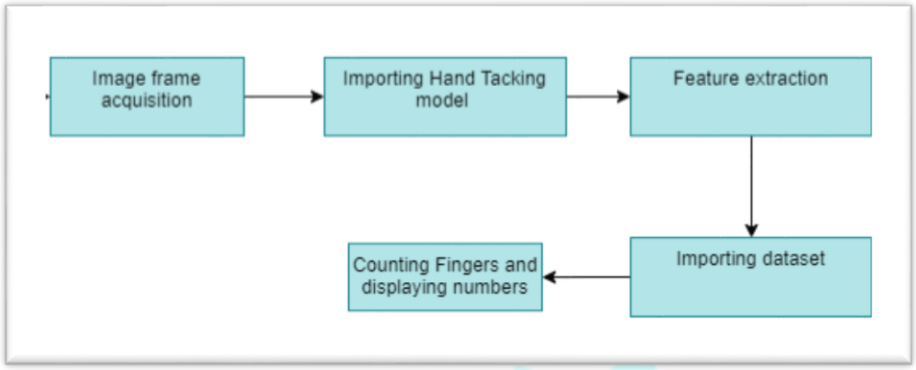


Fig.4 Gesture Volume Control Model

c. Hand gesture tracking algorithms:

1. OpenCV: Short for Open-Source Computer Vision, OpenCV is a C++ library designed for computer vision tasks. It is freely available and compatible with multiple platforms. OpenCV offers real-time GPU acceleration capabilities, which are utilized in applications such as face and gesture recognition systems, as well as 2D and 3D feature toolkits [1].

2. Autopy: Autopy is a Python module for GUI automation that operates seamlessly across different platforms. It tracks the movement of fingertips and provides binary output (0 or 1) to indicate whether a finger is lifted or lowered. This output is integrated with OpenCV, which also generates the appropriate image frame [2].

3. **MediaPipe:** Developed by Google and released as an open-source tool, MediaPipe serves as a versatile solution for building multimodal pipelines in machine learning applications. Utilizing vast amounts of prior data inputs, the MediaPipe Hands Model is trained to accurately detect hands, enabling precise hand identification. This technology can be leveraged for tasks such as hand gesture control and interpreting sign language. Moreover, MediaPipe facilitates augmented reality experiences by enabling the overlay of digital content onto real-world environments [3].

3.2 Voice Assistant:

a. Workflow:

To achieve its objectives, the Voice Assistant integrates several modules: speech recognition, natural language processing, and speech synthesis. It responds to user commands by repeatedly analyzing the user's voice input. Based on these commands, the assistant generates text responses that meet the user's expectations [5]. In today's landscape, various voice assistants are available, including the IBM Watson app, the Cortana desktop app, the Amazon Alexa Wall Clock app (for individual devices), and Google Assistant (for multiple devices). Each assistant offers unique features within its application, enabling them to handle diverse situations efficiently [6]. These assistants effectively address many user concerns.

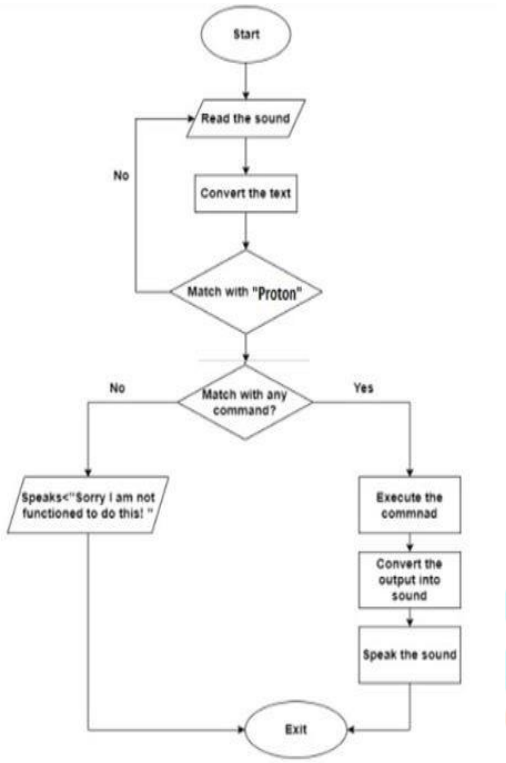


Fig 5 : Flowchart of voice assistant

b. Methodology:

1. **PyAutoGUI:** PyAutoGUI is a Python module designed to manage keyboard inputs, mouse clicks, and cursor movements. It finds applications in automation, game development, and GUI testing tasks [9].
2. **Pyttsx3:** This Python library utilizes Text-To-Speech engines to convert text into speech. Easily installable via pip, it is commonly employed in speech-enabled applications, language learning tools, and assistive technology solutions [10].

The VGF System combines two phases into a unified system. It determines if the voice assistant is active and responds to user commands accordingly [11]. When instructed to initiate gesture recognition, it proceeds to recognize gestures and execute corresponding actions.

4. Results:

A. Gesture Recognition:

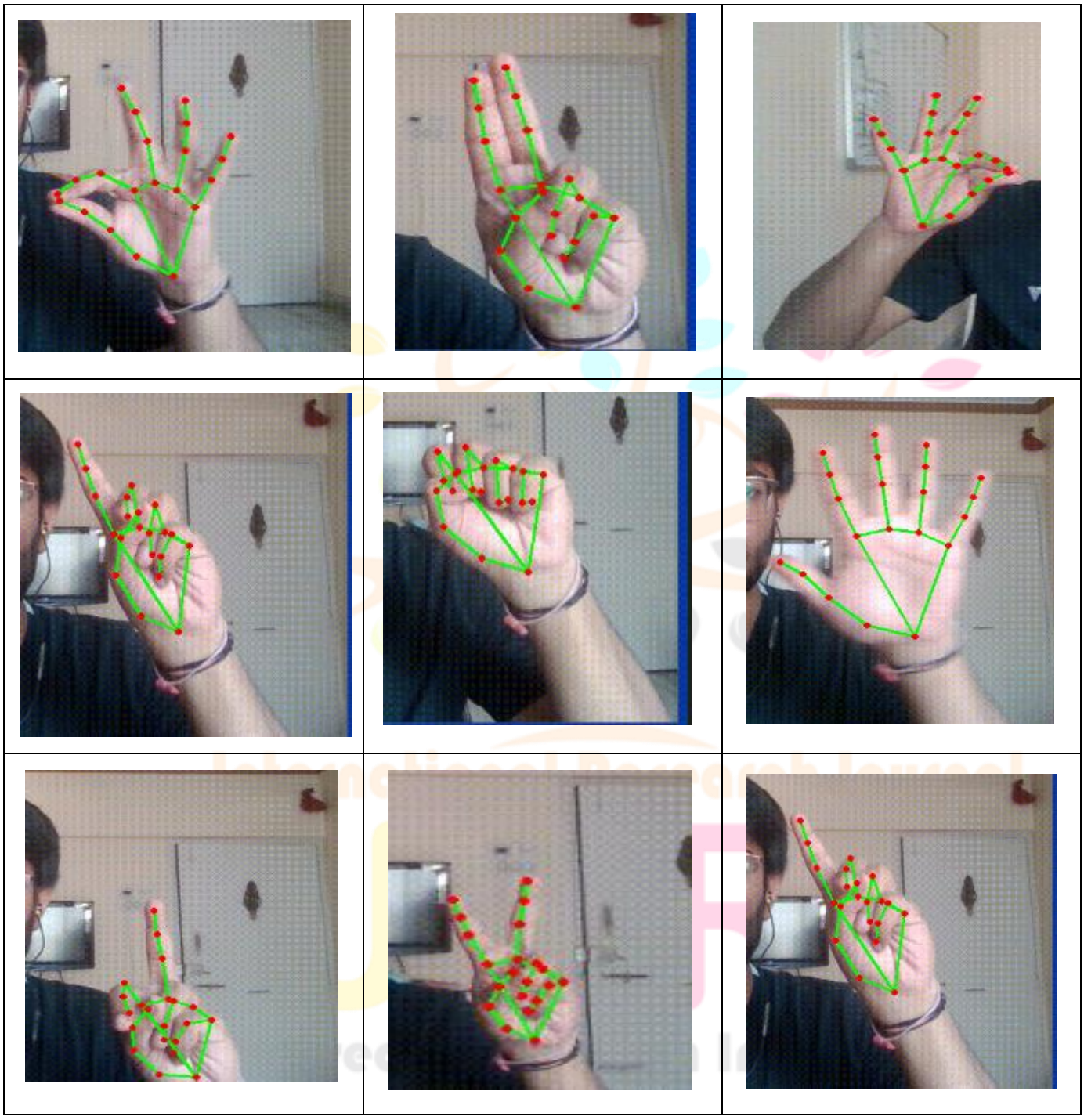
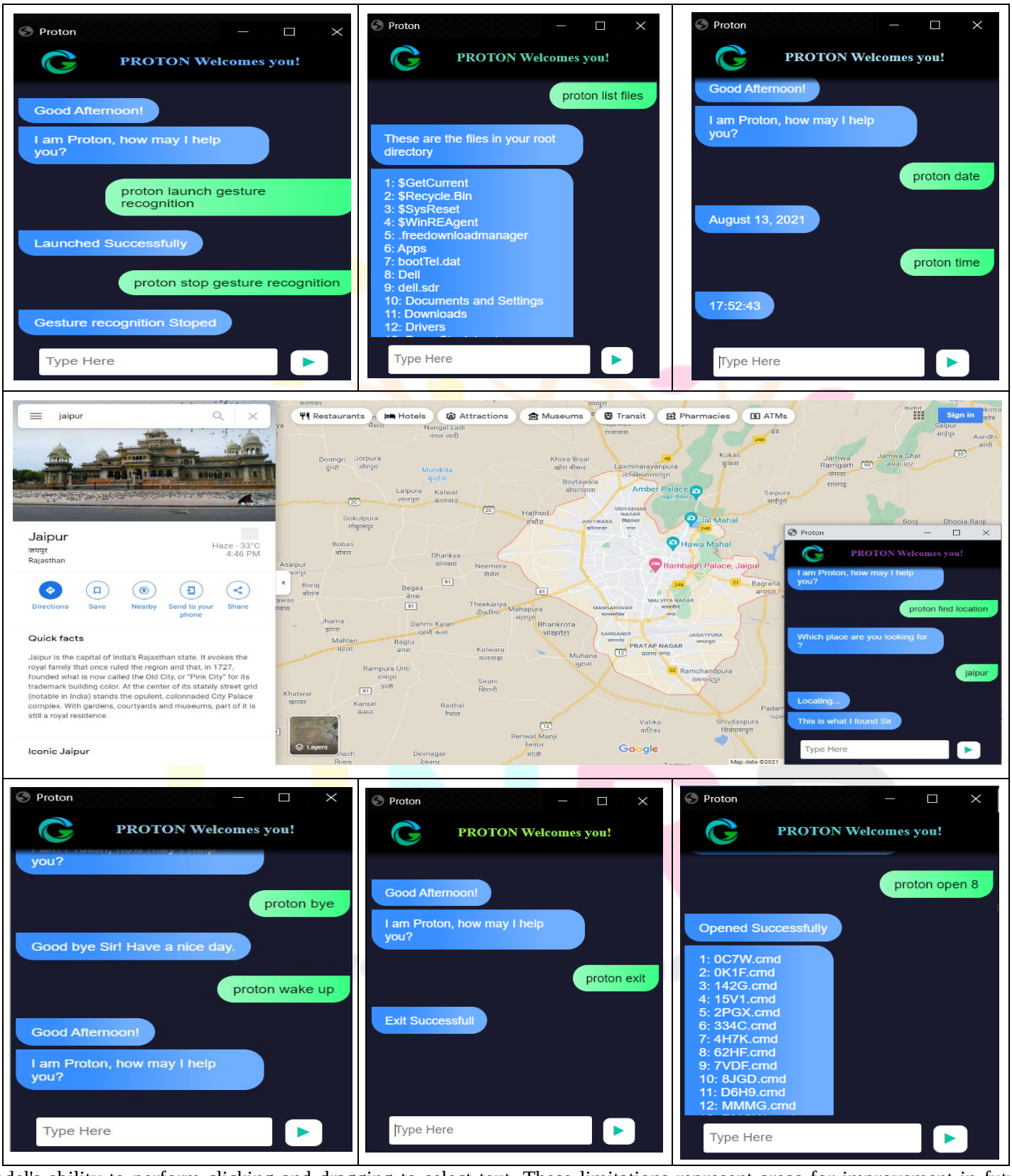


Fig 6: Different Gestures

B. Results of Voice Assistant:

5. Future Work: The system is designed to be user-friendly and completely autonomous. Future research aims to enhance algorithms by integrating speed-based robot motion control with segmentation approaches [13]. Some limitations of the proposed AI virtual mouse include a minor decrease in the accuracy of the right-click mouse function and difficulties in the



model's ability to perform clicking and dragging to select text. These limitations represent areas for improvement in future iterations of the AI virtual mouse system.

6. Conclusion: Python and its libraries were utilized to develop the VGF system. This system allows users to engage with it solely through hand gestures and a voice assistant, enabling dynamic functionalities such as online searches, time and date inquiries, and even mimicking physical mouse actions. This eliminates the necessity for a physical mouse, rendering the system completely hands-free. Observations indicate that the VGF system outperforms other comparable systems.

7. References:

[1] Singh, J., Goel, Y., Jain, S., & Yadav, S. (2023). Virtual mouse and assistant: A technological revolution of artificial intelligence. arXiv preprint arXiv:2303.06309.

[2] Engineering, J. O. H. (2023). Retracted: Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread. Journal of healthcare engineering, 2023, 9804176.

[3] Kathar, S., Jagtap, S., Pardeshi, S., Giri, S., & Kapare, S. AI Virtual Mouse.

[4] Guha, J., Kumari, S., & Verma, S. K. (2022). AI Virtual Mouse Using Hand Gesture Recognition. IJRASET, 10(IV)

[5] Teja, A. R., & Nellore, A. P. (2022). GESTURE CONTROLLED AI VIRTUAL MOUSE SYSTEM USING COMPUTER VISION. GESTURE, 12(7), 134-138.

[6] Shibly, K. H., Dey, S. K., Islam, M. A., & Showrav, S. I. (2019, May). Design and development of hand gesture based virtual mouse. In 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT) (pp. 1-5). IEEE.

[7] Khan, I., Kanchan, V., Bharambe, S., Thada, A., & Patil, R. Gesture Controlled Virtual Mouse with Voice Assistant.

[8] Nandwalkar, J., Mandal, M., Khirari, A., & Bhalchim, T. Control Mouse using Hand Gesture and Voice.

[9] Quam, D. L. (1990, May). Gesture recognition with a dataglove. In IEEE Conference on Aerospace and Electronics (pp. 755-760). IEEE.

[10] Hsieh, C. C., Liou, D. H., & Lee, D. (2010, July). A real time hand gesture recognition system using motion history image. In 2010 2nd international conference on signal processing systems (Vol. 2, pp. V2-394). IEEE.

[11] Reddy, V. V., Dhyanchand, T., Krishna, G. V., & Maheshwaram, S. (2020, September). Virtual mouse control using colored finger tips and hand gesture recognition. In 2020 IEEE-HYDCON (pp. 1-5). IEEE.

[12] Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. ieee assp magazine, 3(1), 4-16.

[13] Kulkarni, A. U., Potdar, A. M., Hegde, S., & Baligar, V. P. (2019, July). Radar based object detector using ultrasonic sensor. In 2019 1st International Conference on Advances in Information Technology (ICAIT) (pp. 204-209). IEEE.

[14] Kasar, M., Kavimandan, P., Suryawanshi, T., & Abbad, S. (2024). AI-based real-time hand gesture-controlled virtual mouse. *Australian Journal of Electrical and Electronics Engineering*, 1-10.