RESEARCH ARTICLE OPEN ACCESS

AI Virtual Mouse System Using Hand Gestures and Voice Assistant

Likitha R
Department of Computer science and Engineering,UVCE Bengaluru,India

Kumaraswamy S
Department of Computer science and Engineering,UVCE Bengaluru,India

Revathi B
Department of Computer science and Engineering,RRIT Bengaluru,India

ABSTRACT—
A great example of HCI progress is the mouse. The present cordless mouse or Bluetooth mouse still requires devices such as a battery for energy and a card to connect it to the computer; hence, it is not completely device-free. The suggested AI virtual mouse system employs a digicam or an internal camera to capture hand motions, then uses object recognition hand tip detection and a voice assistant to improve the system's precision, therefore solving the aforementioned problem. A machine-learning algorithm forms the basis of the system. Digital hand gestures can be used to control a computer in place of a physical mouse, performing tasks such as left- and right-clicking, navigating, and pointing. For hand recognition, the software employs deep learning, and for voice assistance, it makes use of the Sapi 5 engine, which executes tasks using python modules. The proposed method implemented with basic mouse functions and also brightness, volume control which also handle fluctuations due to noise would eliminate the requirement for human involvement and computer control equipment in the fight against the spread of COVID-19.

KEYWORDS—HCI, Voice assistant, Sapi 5, Python, mediapipe;

Date of Submission: 11-12-2022 Date of Acceptance: 27-12-2022

I. INTRODUCTION

This template As Bluetooth and other wireless technologies continue to develop, AR and other devices we use on a regular basis will become more compact and portable. This research proposes an AI based virtual mouse system that uses voice assistant and hand gestures to enable mouse-like movements on a computer. The suggested system's main purpose is to enable the user to perform mouse pointer capabilities and scroll operations by using camera of laptop i.e. inbuilt camera or using web camera instead of using traditional mouse. Use of computer vision for detecting hand gestures and hand tips in HCI [1]. Using either web camera or a built-in camera, the AI virtual mouse system tracks the movement of a user's fingertip to perform mouse cursor movements including clicking, scrolling, pointing and perform brightness and volume control. By listening for specific keywords and filtering out nearby noise, voice assistants can provide relevant information in response to user requests. A voice assistant is a type of assistant that is based on digital technology and uses human voice using algorithm and synthesis to pay attention to specific voice commands and provide relevant information or perform specific functions. Some assistants are sketched up independently for each unique device applications like, while others, utilising as a mouse , like voice assistants, totally a software system primarily builds on, may be and ready to incorporate into all devices.

II. LITERATURE SURVEY

A. Gesture with Hands

There are several types of hand gestures that serve different purposes, as described by Ying and Thomas (2001). Gestures used for controlling navigation and pointing in a virtual environment or other display control applications are known as "controlling gestures." One use of gesture control is the virtual mouse interface (Tsang et al., 2005), which may be used in place of a traditional mouse by letting users move the cursor with their hands. Human connection isn't complete without the use of conversational gestures, such as gesturing to emphasise a point. In the context of tele-operation (Hasegawa et al., 1995) and virtual assembly, manipulative gesture is a means of interacting with virtual objects.

B. Neural Networks

The Neural networks are a powerful tool for modelling complicated relationships between input and output using non-linear statistical data. As is evident in Figure 2.4, there are two primary types of neural network architecture: feed-forward and recurrent. Recurrent neural networks are able to identify time-varying patterns because, unlike feed-forward networks, they transfer input from later processing to earlier stage. (Samir, 2000) The neural network's superiority in noise immunity makes it a popular choice for use in pattern recognition categorization. The back propagation learning algorithm is a well-known technique for training a feed forward neural network. Delta Rule and Perceptron are two further examples of learning algorithms that may be used to train a neural network. In a 2009 study (Alsmadi et al.), Temporal hand gesture recognition is a common use of neural networks.

C. Voice Assistant

There have been several recent developments and advances in the area of voice-based assistants. The widespread use of related technologies in consumer electronics like fitness bands and smartwatches, speakers, mobile phones, Bluetooth earbuds, desktop computers, and laptops televisions, and so on has been a major factor in the industry's meteoric rise. Today, voice assistants are becoming increasingly commonplace in the smart gadgets being introduced to the market. The data created nowadays is massive, so if we want our assistant to be able to handle it and provide better results, we'll need to put machine learning into it and teach our gadgets to be effective with the tasks for which they were designed. Technologies such as the Internet of Things, natural language processing, and big data access management stand on par with machine learning in terms of significance. Voice-activated assistants can make our lives much simpler. All you have to do is give the system voice commands, and it will take care of everything for you, from translating your speech to text to extracting the keywords to running searches based on those keywords. The unique direct modelling technique for speech recognition proposed in the paper Speech recognition using flat models by Patrick Nguyen and others simplifies the measurement of sentence consistency. This method has been dubbed the Flat Direct Model (FDM). They deviated from the standard Markov model and created a nonsequential model. By taking this method, they were able to tackle an important issue with specifying features. The absolute rate of phrase errors was reduced by 3% thanks to the template-based features [2].

The potential of IPAs that employ Natural Language Processing and cutting-edge computing technologies for learning is investigated further in the paper On the track of AI: Learning with Intelligent Personal Assistant by Nil Goksel and coauthors. In essence, they have examined Intelligent Personal Assitants operational framework within the context of AI [4].

III. PROPOSED SYSTEM

A high-quality finger and hand tracking system, MediaPipe Hands. Machine learning (ML) is used to determine 2D and 3D landmarks of a hand from a single image. Whereas the present state-of-the-art methods often require robust PC settings for inference, scalable and our solution delivers real-time performance on a mobile phone to many hands. It is our sincere desire that making this hand perception capabilities available to the broader research and development community would inspire the development of novel use cases, leading to the creation of novel applications and the discovery of novel research topics.

A. Palm Detection Model

In this a single shot detector model for hand identification and position like detection using face mesh is tailored for mobile real-time usage. mediapipe lite model and full model must recognise hands in a wide range of sizes, with a huge scale span (20x) relative to the picture frame, and in occluded and self-occluded states, making hand detection a challenging problem. Because of the lack of high contrast patterns in the hand region (as is present in the face, for example in the eye and mouth area), reliable visual detection of hands is more challenging. However, precise hand localisation is aided by other information, such as the, torso and arm, or human traits.

mediapipe method is interdisciplinary in nature and uses a variety of techniques to address the aforementioned issues. Training a palm detector instead of a hand detector is the first step since it is far simpler to estimate the bounding boxes of rigid objects like palms and fists than it is to identify hands with movable fingers. For social and self-scenarios, such handovers, the non-maximum suppression method is especially useful due to the small size of the palms involved. Modeling palms using square anchor boxes (connections in ML language) and ignoring other aspect ratios can further minimise the number of anchors by a factor of 4-5. In addition, even for little objects, a feature extractor based on a codec pair is used in order to grasp the whole scene context (similar to the Retina Net approach). Last but not least, because to the sizeable scale variance, palm model minimised attentional drift during training to keep a large number of hooks.

By combining these methods, mediapipe model are able to increase palm identification accuracy to 95.7% on average. To put things in perspective, a baseline of just 86.22% is achieved when using a standard cross entropy loss and no decoder.

B. Hand Landmark Model

Mediapipe approach overcomes these problems by employing a variety of Once the palm has been recognised over the whole image, or direct coordinate prediction, or mediapipe hand landmark model uses regression, to pinpoint 2D and 3D hand-knuckle positions inside the observed hand areas. The model is able to develop a consistent internal hand posture representation and is resilient even when only a portion of a hand is seen or when the hand is partially obscured by the model's own body.

medipipe have personally labelled over 30,000 photos from the actual world with 21 3D coordinates to use as ground truth data. mediapipe additionally render a high-quality synthetic hand model over different backdrops and map it to the associated 3D coordinates to better cover the available hand positions and give extra oversight on the nature of hand geometry.

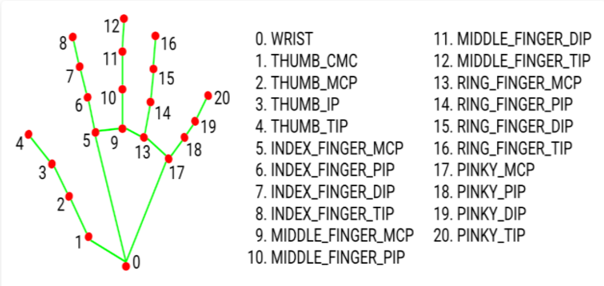


Fig. 1. Hand Landmark Model

Fig. 2.

C. System Structure For Voice Assistant

- The suggested concept for an efficient method of constructing a Personal voice assistant makes use of a Speech Recognition library with several in-built capabilities, allowing the assistant to comprehend the command provided by the user and replying to the user in voice, using Text to Speech operations. Once an assistant has recorded a user's voice instruction, underlying speech-to-text techniques can be used.
- Conceptual Design
- One component of the system architecture is a microphone for capturing speech patterns.
- The second is the transcription of audio files.
- Comparing the input to a set of rules that have already been established.
- Resulting in the expected outcome.

D. Simple Procedure

The primary process of a voice assistant is depicted in the diagram below. The process of converting spoken words into writing is called "speech recognition." The computer then uses the character set of the command to locate and run the relevant script. However, that is not the only layer of intricacy. No matter how much time you put in, there is still another component that greatly affects whether or not a product is seen.

The voice recognition equipment is easily distracted by background noise. A possible explanation for this is because people have trouble telling the difference between your speech and background noises like a dog barking or a helicopter passing overhead

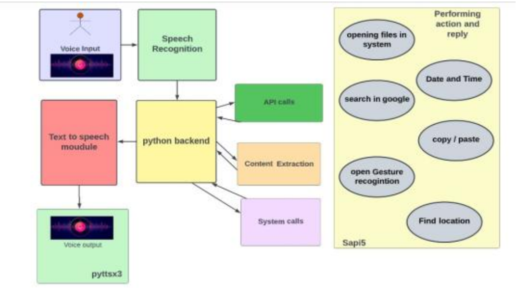


Fig. 3. Basic Work flow of voice assistant

E. Process Flow Diagram

In the form of Siri, Google Voice, and Bixby, voice assistants are already built into our mobile devices. Sales of smart speakers like the Amazon Echo and Google Home are increasing at the same rate as sales of smart phones were a decade ago, according to a recent NPR research, which found that around one in six Americans currently has one. However, you may feel as though the voice revolution is still far off at work. One obstacle is the rise of open offices, since no one wants to be "that guy" who won't stop yelling at his computer. There are 3 separate parts to this Assistant. The very first is the capability of the assistant to recognise the user's voice and act on it. Second, processing the user's input by determining its meaning and applying it appropriately. Finally, the assistant provides the user with the outcome in real time by voice. The assistant will begin by collecting data from the user. The assistant will take the user's analogue voice input and transform it into digital text

IV. METHODOLOGY

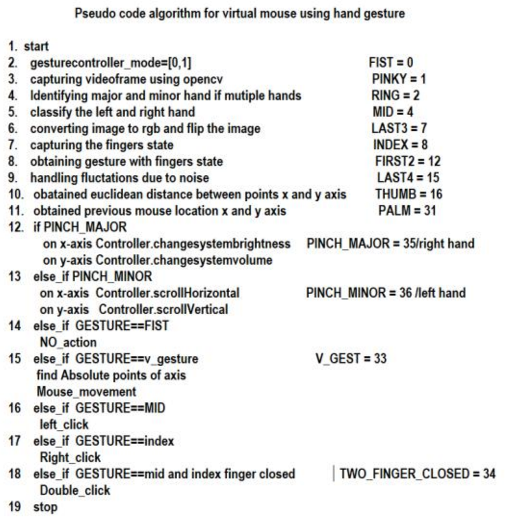
The suggested artificial intelligence virtual mouse system utilizes data from camera images on a personal computer or laptop. As can be seen in Figure 4, the Python computer vision package OpenCV is used to build a video capture object, which in turn triggers the web camera to begin recording. The frames are taken by a webcam and sent to a computerized artificial intelligence

A. Video stream for hand

Handles camera using digicam, object obtained from opencv which captures the video stream and assign the height and width in pixels for obtained frame, also used flip image and converting from BGR to RGB image

B. Hand detection

Setting major hand or minor hand using points from mediapipe framework, obtaining maximum hands is 2 ,min detection confidence is 50% and min tracking confidence is 50 % and getting status of finger.



Pseudo code algorithm for virtual mouse using hand gesture

```
start
 gesturecontroller_mode=[0,1]                                    FIST = 0
 capturing videoframe using opencv                               PINKY = 1
 Identifying major and minor hand if mutiple hands               RING = 2
 classify the left and right hand                                MID = 4
 converting image to rgb and flip the image                      LAST3 = 7
 capturing the fingers state                                     INDEX = 8
 obtaining gesture with fingers state                            FIRST2 = 12
 handling fluctations due to noise                               LAST4 = 15
 obatained euclidean distance between points x and y axis        THUMB = 16
 obtained previous mouse location x and y axis                   PALM = 31
 if PINCH_MAJOR
    on x-axis Controller.changesystembrightness     PINCH_MAJOR = 35/right hand
    on y-axis Controller.changesystemvolume
 else_if PINCH_MINOR
    on x-axis  Controller.scrollHorizontal          PINCH_MINOR = 36 /left hand
    on y-axis   Controller.scrollVertical
 else_if  GESTURE==FIST
    NO_action
 else_if  GESTURE==v_gesture                         V_GEST = 33
    find Absolute points of axis
    Mouse_movement
 else_if  GESTURE==MID
    left_click
 else_if  GESTURE==index
    Right_click
 else_if  GESTURE==mid and index finger closed      | TWO_FINGER_CLOSED = 34
    Double_click
 stop
```

C. Hand Gesture

Points for each finger are fist = 0, pinky finger = 1, ring finger = 2, mid finger = 4, last3 finger is open = 7, index = 8, first2 finger is open= 12, last4 finger = 15, thumb is open= 16 , palm finger = 3 and extra mappings are v_gest = 33 v_gesture if index and mid finger in v shape, two_finger_closed = 34 if index and mid finger is join, pinch_major = 35 if right hand with thumb and index finger is join with finger tip ,pinch_minor = 36 if left hand with thumb and index finger is join with fingertip.

Obtain current finger state 1 if finger is open else 0.set 'finger' by computing ratio of distance between finger tip,middle knuckle, base knuckle. points for each finger [[8,5,0],[12,9,0],[16,13,0],[20,17,0]] from index to pinky for thumb is zero. Calculating Euclidean distance between 'point' for finding state of finger and ratio.

$$d = \sqrt{[(x2 - x1)^2 + (y2 - y1)^2]}$$

D. Handling Fluctations due to noise

Representing gesture corresponding to Enum 'Gest'.sets 'frame_count', 'original_gesture', 'previous_gesture', handles fluctations due to noise.

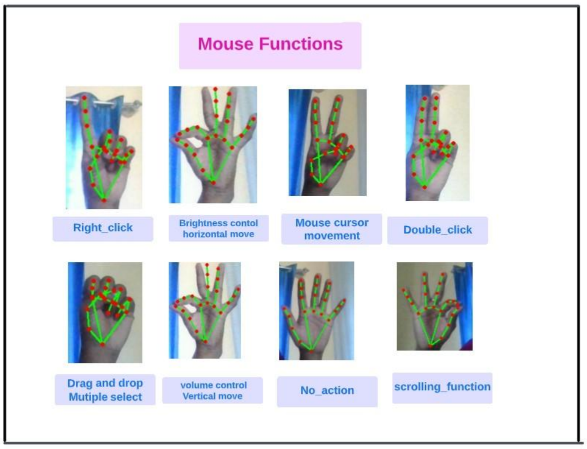
E. To control Position of Pointer

Using the mediapipe package of Python, if either the index finger (with tip Id = 8) or the middle finger (with tip Id = 4) is raised, the mouse cursor is caused to move around the window of the computer.

The detected and tracked hands are collected here, with each hand being represented by a list of 21 landmarks. The x and y coordinates are scaled to [0.0, 1.0] and (x, y, and z coordinates). It is based on the image's width and height. With the wrist as the starting point, and the lower the value, the depth of a landmark is represented by z and the closer the landmark is to the camera. Z's magnitude is about comparable to that of x.

A trove of detected/tracked hands, with each one represented by a list of 21 landmarks in global coordinates. The x, y, and z components of each landmark are real-world three-dimensional coordinates in meters, with the hand's approximate geometric center serving as the point of origin.

Handedness data from identified and followed hands (i.e. is it a left or right hand). An individual's score and label make up their hand. label can have the values "Left" or "Right" in a string. score is the expected handedness probability, and it is greater than or equal to 0.5 in every case (and the opposite handedness has an estimated probability of 1 - score).


Mouse Functions
Right_click
Brightness contol horizontal move
Mouse cursor movement
Double_click
Drag and drop Mutiple select
volume control Vertical move
No_action
scrolling_function

F. To control mouse function with Gestures

Execute mouse function with particular detected gestures.

Obtaining x and y axis mouse coordinate and execute mouse function. If V shape gesture is detected with index and mid finger then mouse movement from obtaining absolute vale from x and y axis. If Fist gesture recognition, then drag and drop and multiple select function will be handled.

If Pinch gesture is detected through major hand, on x-axis Control change system brightness in fig 5, on y-axis Control change system volume in fig 6. If pinch gesture is detected through Minor hand, on x-axis 'Control scroll Horizontal', on y-axis 'Control scroll Vertical'. Obtaining pinch level with distance and position and assign threshold.

If mid finger is open with 4 finger tip id, then left click,
Index finger is open with finger tip id then right click in fig 7, two finger closed then double click.

All function can be trigger and execute through pyautogui
Which helps to interact with os and execute mouse functions.

G. Methodology used by voice assistant

Headings, From the get-go, we utilize sapi5 and pyttsx3 to provide our software the ability to interact with the system voice. The pyttsx3 library is a Python implementation of text-to-speech technology. It's compatible with Python 2 and 3, unlike many other libraries, and it even functions while you're not online. Windows programmed may take use of voice detection and synthesis thanks to the Speech Application Programming Interface (SAPI), an API created by Microsoft. The program's capabilities are then defined in the main function. The suggested system is expected to be able to do the following.

a) The helper constantly polls the user for feedback and awaits further instructions. The listening time may be adjusted based on the user's needs.
(b) The helper will keep asking the user to repeat the instruction if it doesn't understand it the first time around.
c) Depending on the user's preferences, this assistant can be programmed to speak in a male or female voice.

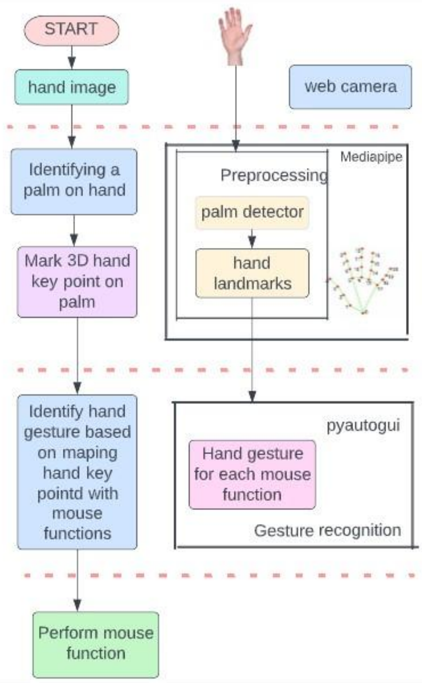

START
hand image
web camera
Identifying a palm on hand
Mediapipe
Preprocessing
palm detector
hand landmarks
Mark 3D hand key point on palm
Identify hand gesture based on maping hand key pointd with mouse functions
pyautogui
Hand gesture for each mouse function
Gesture recognition
Perform mouse function

Fig. 4. Basic Work flow of virtual mouse

V. EVALUATION OF EXPERIMENT

The proposed artificial intelligence virtual mouse technology uses computer vision to better facilitate user interaction with digital environments.

It is difficult to perform comparative evaluation of multiple versions of the AI virtual mouse system due to the dearth of relevant datasets. Fingertip and hand grip identification were tested under varying illumination and camera distance situations. In Table 1 we see experimental data. The findings of the experiment, in which each participant tried the Services online mouse device 15 times under normal lighting conditions, 3 times under dim lighting conditions, 8 times at a great proximity from the camera, and 3 times at a far distance, are presented in Table 1.

TABLE I. EXPERIMENTAL RESULT

Hand Tip Gesture	Mouse Function Perform	Success	Failure	Accuracy
Gesture tip id is 33	Mouse movement	100	0	100
Pinch major(x axis)	Brightness control	97	3	97
Pinch major(y axis)	Volume control	100	0	100
Two finger closed tip id 34	Drag and drop	95	5	95
Pinch minor(x axis)	horizontal scroll	100	0	100
Pinch minor(y axis)	vertical scroll	100	0	100
Tip id 8 index finger is up	Left click	100	0	100
Tip id 4 mid finger is up	Right click	100	0	100
Two finger closed tip id 34	Double click	98	0	98
All five finger are up	No action	100	0	100
Total		1893	13	97

TABLE II. EXISTING MODELS

Existing models	Accuracy%
models Virtual mouse system using RGB-D images and fingertip detection [11]	96.13
Palm and finger recognition based [12]	78
Hand gesture-based virtual mouse [13]	78
The proposed AI virtual mouse system	97

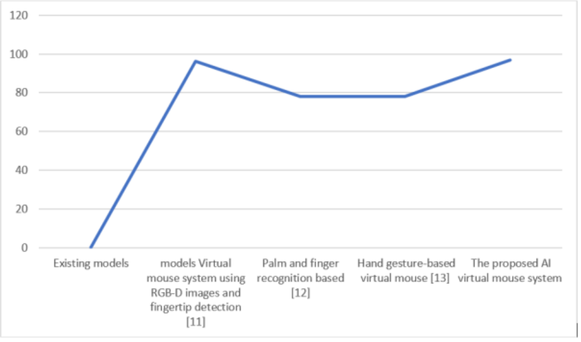


Fig. 5. Graph for Existing model

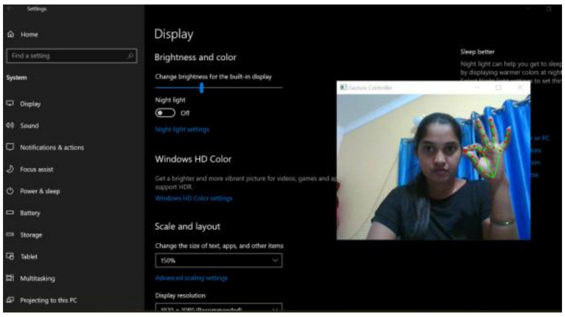


Fig. 6. Snapshot of Brightness control

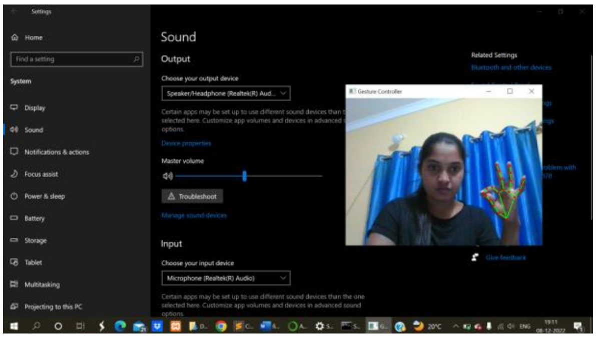


Fig. 7. Snapshot of volume control

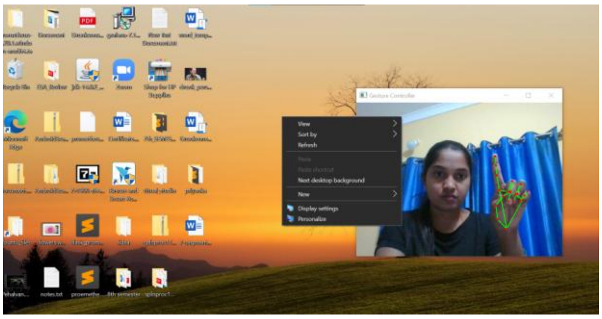
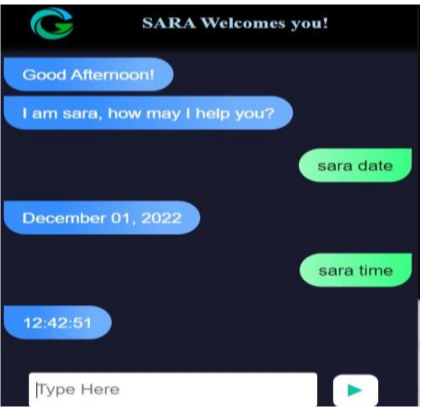


Fig. 8. Snapshot of Right click

A. *Execution of Voice assistant*

When first activated, the helper will wait for instructions from the user. The assistant will record the user's voice command if one is given and then look for the specified keyword inside it. If the assistant was able to identify a relevant keyword, it will execute the requested action and provide its results to the user verbally and in text on the terminal. If the user does not provide proper input, the assistant will wait for it once again. Every one of these features is crucial to the smooth operation of the system as a whole.


SARA Welcomes you!

Good Afternoon!

I am sara, how may I help you?

sara date

December 01, 2022

sara time

12:42:51

s

Fig. 9. GUI for voice assistant

VI. CONCLUSION

The primary goal of the AI virtual mouse system is to allow the user to operate the mouse pointer with hand movements rather than a hardware mouse. The suggested system may be implemented with the use of a webcam or an integrated camera by processing the frames to carry out the necessary mouse actions based on the detected hand motions and hand tip.

In this thesis, we offer a high-level overview of how we implemented a Static Voice enabled personal assistant in Python for the desktop. This voice-enabled personal assistant will be more useful to persons with disabilities and help save time in modern lifestyles.

This assistant does a good job at carrying out some of the duties the user specifies. In addition, this assistant may do a wide variety of tasks, including as text message delivery to the user's mobile device, YouTube automation, and information retrieval from Wikipedia and Google, all in response to a single voice query.

The voice assistant has allowed us to automate several services with a single command. The majority of the user's work, such as online searching, is simplified by this tool. Our goal is to make this tool so capable that it can take the position of human server administrators entirely. The project was constructed utilizing modules from open-source software that have the support of the Anaconda community, so any changes may be implemented quickly. The project's modular design allows for further customization and the installation of new features without impacting the workings of the existing system

REFERENCES

[1]. J. Katona, "A review of human–computer interaction and virtual reality research fields in cognitive InfoCommunications," Applied Sciences, vol. 11, no. 6, p. 2646, 2021

[2]. J. D. L. Quam, "Gesture recognition with a DataGlove," IEEE Conference on Aerospace and Electronics, vol. 2, pp. 755–760, 1990.

[3]. D.-H. Liou, D. Lee, and C.-C. Hsieh, "A real time hand gesture recognition system using motion history image," in Proceedings of the 2010 2nd International Conference on Signal Processing Systems, IEEE, Dalian, China, July 2010.

[4]. S. U. Dudhane, "Cursor control system using hand gesture recognition," IJARCCE, vol. 2, no. 5, 2013. K. P. Vinay, "Cursor control using hand gestures," International Journal of Critical Accounting, vol. 0975–8887, 2016.

[5]. L. Thomas, "Virtual mouse using hand gesture," International Research Journal of Engineering and Technology (IRJET, vol. 5, no. 4, 2018.

[6]. P. Nandhini, J. Jaya, and J. George, "Computer vision system for food quality evaluation—a review," in Proceedings of the 2013 International Conference on Current Trends in Engineering and Technology (ICCTET), pp. 85–87, Coimbatore, India, July 2013.

[7]. J. Jaya and K. Thanushkodi, "Implementation of certain system for medical image diagnosis," European Journal of Scientific Research, vol. 53, no. 4, pp. 561–567, 2011.

[8]. P. Nandhini and J. Jaya, "Image segmentation for food quality evaluation using computer vision system," International Journal of Engineering Research and Applications, vol. 4, no. 2, pp. 1–3, 2014.

[9]. J. Jaya and K. Thanushkodi, "Implementation of classification system for medical images," European Journal of Scientific Research, vol. 53, no. 4, pp. 561–569, 2011.

[10]. J. T. Camillo Lugaresi, "MediaPipe: A Framework for Building Perception Pipelines," 2019, https://arxiv.org/abs/1906.08172

[11]. D.-S. Tran, N.-H. Ho, H.-J. Yang, S.-H. Kim, and G. S. Lee, "Real-time virtual mouse system using RGB-D images and fingertip detection," Multimedia Tools and ApplicationsMultimedia Tools and Applications, vol. 80, no. 7, pp. 10473–10490, 2021.

[12]. A. Haria, A. Subramanian, N. Asokkumar, S. Poddar, and J. S. Nayak, "Hand gesture recognition for human computer interaction," Procedia Computer Science, vol. 115, pp. 367–374, 2017.

[13]. K. H. Shibly, S. Kumar Dey, M. A. Islam, and S. Iftekhar Showrav, "Design and development of hand gesture based virtual mouse," in Proceedings of the 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), pp. 1–5, Dhaka, Bangladesh, May 2019.