

SYSTEM FUNCTIONALITY CONTROL BY AIR MOUSE

Chitineni Yashwanth¹ Akula Abheshekh²

Student of GITAM Institute of Technology and Management ¹

Student of CMR College Engineering of Technology²

¹Computer Science and Engineering ²Electronics and Communication Engineering

Abstract – This paper puts forward the usage of a camera-based hand detection system which uses live stream from a webcam. This system allows the user to perform basic mouse operations like left-click, right-click, scroll up, scroll down and drag and also feature additional operations like volume adjustment, brightness adjustment, tab switching and desktop switching and even capturing screenshots in your laptop/desktop. This system is cost efficient as it only uses a built in webcam to capture or track the user's hands having 3D landmarks in two dimensions. The library we use in our proposed system is Mediapipe and it is implemented using Python and OpenCV. Keeping in mind the importance of easy and effortless communication, we have implemented this system. Each operation has its own input gesture and the corresponding output is shown on the screen.

Key words – Python, OpenCV, Mediapipe, Hand gestures, image processing, highGUI, Human Machine Interface.

1. INTRODUCTION

Communication is very important in one's life but not everyone can communicate verbally. So the next efficient and simple way is using hand gestures where even deaf and dumb people can understand. In this system we use an in-built webcam in a laptop or an external webcam mounted on top of the desktop to capture the hand gestures having 3D landmarks. The process starts with detection of landmarks, determining the position of each landmark using the X,Y coordinates, performing operations based on the hand gestures.

The potential to recognise the shape and movement of the user's hands can be an important component in enhancing the user experience across a variety of technological platforms. For example, it can become the base for sign language understanding and hand gesture control. Hand gestures come naturally to people but to a computer it is a challenging task to perceive it and perform tasks. Hands usually have certain obstructions like shakes which affect the precision. Hand Gesture recognition is related majorly to image processing and machine learning. It is an important research topic because in some situations we require sign languages for communication[1].

Since hand gesture recognition is a natural ability to us, it is a difficult task for a computer to perceive as hands often overlap on each other[2].

This paper allows users to experience a hand gesture recognition tool effortlessly with minimal cost and equipment. It can perform actions like volume, brightness control, tab and desktop switching.

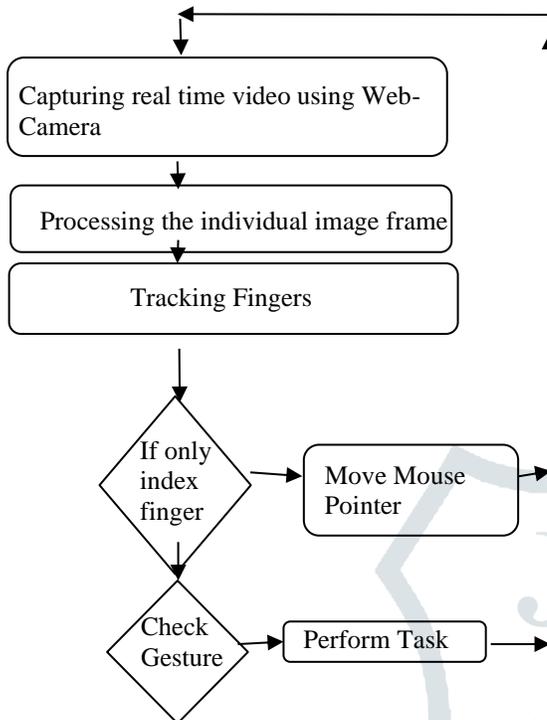
2. EXISTING SYSTEM

In recent times, due to the pacing development of various software technologies, a need for good interaction between humans and machines has become an issue. The use of wired or wireless mouse is not the most comfortable method for many users[3]. So the use of hand gesture recognition has been introduced. The existing system consists of a simple hand gesture recognition system performing simple operations like mouse dragging, left and right clicks and simple file transfers. Though there are many systems which make use of the hand recognition, they are only used for simple operations as mentioned. The further development of hand recognition and virtual mouse pointer has not been used.

3. PROPOSED SYSTEM

Our system is a low cost hand recognition system with only a webcam needed to mount on a desktop. We can perform many tasks from simple ones like mouse cursor controlling, left and right-click to some complex tasks like controlling volume and brightness and complex task like capturing screenshots. Our system operates on the cropped image region and returns 3D hand keypoints. Using the 3D keypoints we determine the coordinate values of them and further using them the system determines the gesture and performs the corresponding operation. This system is implemented using Python language with the help of Mediapipe and OpenCV. We use the Python programming language because it is simple, easy to understand and robust. Python code is flexible, responsive and well suitable for creating Hand recognition systems. Mediapipe library assists us to create functions suitable for our proposed system. This system can be further modified by implementing more hand gestures to perform more complex operations, we are just restricted by our imagination.

4. Flowchart



5. Methodology

The methodology is discussed as below.

5.1 Camera Settings

The runtime operations are managed by the in-built or mounted webcam of laptop or desktop. A video capture object is necessary to capture a video. Its parameter is a numerical value which is its device index. This device index is used to specify the camera number. Usually we use the built-in webcam so we pass 0 as parameter. Then we capture the feed and finally release the capture.

5.2 Capturing frames

It is mandatory for the camera to run for the entire program so we used a for loop to capture frames at each instance. We capture the live feed stream from webcam, frame by frame. Then we process each captured frame which is in RGB color space.

5.3 Display the frame

The imshow() may be a function of HighGui and it's required to call the waitKey regularly. The processing of the event loop of the imshow() function is finished by calling waitKey. The function waitKey() waits for key events for a "delay" which is 5ms in this case. Windows events like resizing, input events, etc. are processed by HighGui.

5.4 Mouse Movement

Through live stream via a web camera, the index fingertip is detected in each frame using a media pipe and variation in its position is transformed in screen resolution scale and cursor on the screen is moved.

5.5 Clicking

Tips of the thumb finger and middle finger are monitored and as these fingers move too close to the index finger the click operation is done, the thumb is used to perform the left click operation and the middle finger for the right click.

5.6 Drag

To implement the dragging, we introduce the variable 'dragFlag'. It will be set to "True" if it was first detected. So if dragFlag is set, then it drags the object and releases it when the open gesture is detected.

5.7 SnapShot / ScreenShot

"Thumb" and "Middle" fingers are tracked to and if only these fingers are found, a snap of the screen is taken.

5.8 Scroll

Here "Index" and "Little" fingers are tracked for scrolling, and "Thumb" is used to change the direction of the scroll.

5.9 Volume Control & Brightness Control

A separate flag called "volumeFlag" & "brightnessFlag" is used; it is set to "True" if certain gestures and a pinching operation is used to control the value/percentage.

5.10 Window Switch (3 Finger Swipe)

3 finger gesture is used to represent a similar windows operation called "3 Finger Swipe" which is used for quick change of window.

5.11 Desktop Switch (4 Finger Swipe)

Similar to 3 finger gesture, 4 finger gesture is used to represent a similar windows operation called "4 Finger Swipe" which is used to quick change of desktop on OS versions Windows 8 and above

6. RESULT AND EVALUATION

The main purpose of this system is to assist users who are unable to use their hands in a proper way and who seek to improvise human machine interaction.

As a result of frame rate, the coordinates of landmarks the system predicts the movement of the hand or simply the hand gesture and performs the operation.



FIG 1: CLICK

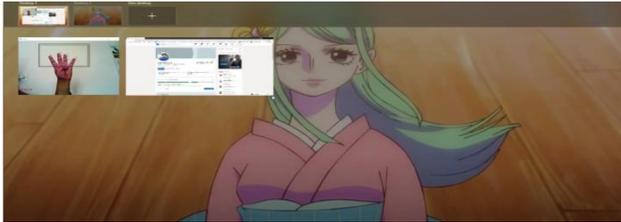


FIG 2: DESKTOP CHANGE



FIG 3: TAB SWITCH

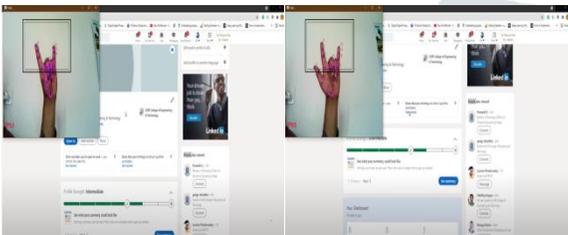


FIG 4: SCROLL UP AND DOWN



FIG 5: RIGHT CLICK



FIG 6: VOLUME AND BRIGHTNESS CONTROL

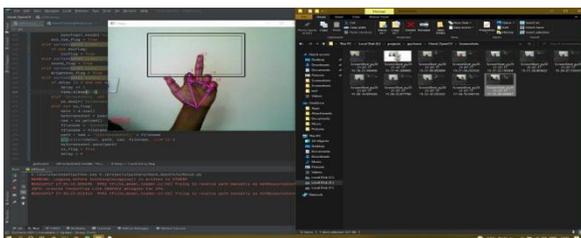


FIG 7: SCREENSHOT

7. CONCLUSION

Gestures play a vital role in creating a Human Machine Interface (HMI). With improved communication between human and machine, various operations can be performed hassle free. Hand gesture recognition can be used to enhance the computer's ability to understand human body language and also reduces the effort of giving inputs manually using mouse and keyboard.

Our proposed work minimises the use of manual operations as mentioned above and also reduces the effort of remembering shortcuts. Hand gesture recognition systems have applications in a wide range of fields such as Augmented reality, Artificial intelligence etc. In layman terms our system uses 21 3D landmark coordinates to predict the hand gesture and perform the operation.

The major extension to this work can be done to make the system able to work at a much more complex background and compatible with different light conditions[4]. This system will also include the execution of several other gestures to perform different computer operations[5].

8. REFERENCES

- [1]Harini V., Prahelika V., Sneka I., Adlene Ebenezer P. (2020) Hand Gesture Recognition Using OpenCv and Python. In: Smys S., Iliyasa A.M., Bestak R., Shi F. (eds) New Trends in Computational Vision and Bio-inspired Computing. ICCVBIC 2018. Springer, Cham. https://doi.org/10.1007/978-3-030-41862-5_174
- [3]VIRTUAL MOUSE USING HAND GESTURE Abhilash S S1, Lisho Thomas2, Naveen Wilson3, Chaithanya C4International Research Journal of Engineering and Technology (IRJET)Volume: 05 Issue: 04 | Apr-2018 pp 3903 to 3906
- [2,4]Hand Gesture Recognition System Using Camera Viraj Shinde, Tushar Bacchav, Jitendra Pawar, Mangesh Sanap international Journal of Engineering Research & Technology (IJERT) Vol. 3 Issue 1, January - 2014 pp- 1628 to 1633
- [5] Ahmad Puad Ismail, Farah Athirah Abd Aziz, Nazirah Mohamat Kasim, & Kamarulazhar Daud (2021). Hand gesture recognition on python and opencv. *IOP Conference Series: Materials Science and Engineering*, 1045(1), 012043.