An Efficient Gesture Based Control Of Operating System

Ms. M.S. Sruthi[1], V. Arun[2], E. Jijin suresh[3], E. Jithin suresh[4]
[1,2,3,4]Department of computer science and engineering
Sri Krishna College Of Technology
[1]sruthims041@gmail.com, [2] arunstark85@gmail.com,[3] jijinsuresh6@gmail.com,
[4]jithinsuresh7060@gmail.com

Abstract

Since the invention of computers, methods for establishing interaction between humans and computers have been looked after. In this the mechanical mouse played a major role in establishing HCI (Human Computer Interaction). However, there are certain drawbacks in using a mechanical mouse such as lack of providing a long-time support, requiring external connections like wires, draining of powers and periodic replacement of batteries. The proposed system provides a path for humans to interact with the operating system very efficiently. It only requires an inbuilt camera or an external camera feed as input. Frames containing the gestures are collected using the camera and the event is predicted with respect the gesture provided. This project promotes human- computer interaction where cursor movement and different mouse events are controlled using hand gestures shown in the real-time camera. Using this the user can control the movement of the cursor. Mouse events such as right click, left click and double click can be executed. Thus, removing all the cons produced by the mechanical mouse.

Keywords: *HCI (Human Computer Interaction), Gesture based recognition, Gesture fluidity*

1. Introduction

Gesture-based conversation between humans has become a habitat. The conversational signs such as giving thumps up or handshake helps us to realize that gesture-based communication is the most straight forward form of communication. Even deaf people can understand what the person is trying to convey through hand gestures, and they communicate accordingly. In this proposed system we will get deep inside gesture-based control of the operating system. The operating system is mainly controlled using a mechanical mouse. The proposed system will control the operating system using hand gestures. Cursor events like right click, left click drag and drop, and many other events are controlled using hand gestures in this project. Few gadgets like the mechanical mouse and wireless mouse have their disadvantages so if the user can't afford a mouse are tired of using a mechanical mouse, they can start to control the mouse through hand gestures shown in the frames collected from the inbuilt or external camera. To build the proposed system python programming language is used to

empower the gesture-based control of the operating system. In python, there are some packages built for satisfying the requirements such as mediapipe, autogui, and autopy. Mediapipe is the main package used which helps in reading and recognizing the human hands and other packages like autopy and autogui are used for performing the cursor events such as right or left click, drag and drop function, and many other operations which are done by the cursor in the operating system. The gesture-based control of the cursor in the operating system is expected to control the operating system even more interactively and comfortably than the mechanical mouse. Furthermore, the proposed system could assist the operating system where the users could interact using the hand gesture shown in the inbuilt or external camera without using any other special gadgets. As the proposed system's main pathway to get input is through the camera the user doesn't need to spend money on any other gadgets, as in this generation laptops are provided with inbuilt camera this method is cost-effective. The process of controlling an operating system or a machine using hand gestures is a very interesting and effective approach in HCI (Human-computer interaction).

1.1. Problem Description & Overview

Traditional methods of operating systems heavily rely on mice as the primary input device, which may not be suitable for all users. Mouse require precise hand movements and dexterity, posing challenges for individuals with motor impairments or conditions like arthritis. Moreover, using a physical mouse can be inconvenient in situations where space is limited, such as on public transportation or while traveling. There is a need for an alternative control mechanism that is more accessible, adaptable, and efficient for a wide range of users.

1.2. Applications in Real World

In this generation laptops are provided with built in cameras, so this method helps in saving space as it mainly involves camera. It is also cost effective as it doesn't require any external connections or devices. The process of interacting with the operating system using gesture based is an effective approach for Human Computer Interaction. Gesture based controls are often seen to be used for sign language detection and for smart homes such as adjusting temperature and managing home securities.

1.3. Objective

The objective is to improve the HCI (Human-Computer Interaction). It is used to eliminate the disadvantages provided by the mechanical mouse and to develop and alternative method for controlling the cursor events. So, the proposed method uses web camera or built in camera in order to collect frames, track the hand landmarks, counts the fingers, calculates the distance and implements the cursor events.

2. Research survey

Control of cursor events using gestures are implemented using various methods. Control of cursor events using gesture recognition can be based on hardware or computer vision.

D.L. Quam proposed a paper "Gesture Recognition with Data Glove" where the hand gestures are identified using a Data Gloves. It is one of the early hardware-based system proposed. The user must wear a Data Glove in order to control the cursor event [9]. Even though this method provided a higher accuracy control it was difficult to use for some gestures.

In 2000, Changbo Hu and Lichen Lang proposed a paper "Virtual Mouse-Inputting Device by Hand Gesture Tracking and Recognition" where tracking is approached using condensation algorithm and active shape model. They combined multi-model templates in order to increase the performance of tracking and HMM (Hidden Markov Models) for recognition [1].

One such method proposed by Alper Aksac in which skin detection and distance classifier are used in identifying the gestures for controlling the cursor events [2]. This method is effective, but it doesn't support dynamic gestures and it only have a limited set of recognized hand gestures.

In 2014, Farooq J and Ali M B proposed a paper "Real Time Hand Gesture Recognition For Computer Interaction" where Fingertip detection methods such as Convex-Hull, K- curvature, Curvature of perimeter are used in detecting the gesture and implementing the cursor event [3]. Even though this method is effective it has a certain limitation such as Convex-Hull method is not robust i.e.., considers knuckles as fingertips and dynamic gestures can't be used.

In 2015, Tsung-Han Tsai and Chih-Chi Huang proposed a paper "Embedded virtual mouse system by using hand gestures" in which skin detection and motion detection are used in identifying the cursor event [4]. Even though this method gives an higher accuracy it can only identify limited cursor events.

In 2019, Kabid Hassan Shibly and Samrat Kumar Dey proposed a paper "Design and development of Hand Gesture Based Virtual Mouse" where it is based on identifying gestures and colour detection. In this the fingertip is captured and tracked with a webcam which bears a colour cap, and the colour and movements are tracked resulting in movement of cursor. It also uses Hue saturation value in order to identify the colour taps in the fingertips [5].

Tsung Han Tsai proposed a paper "Design of hand gesture recognition system for human-computer interaction" where the proposed system uses skin and motion detection for capturing the region-of-interest from the provided frame. For identifying the centroid of an object connected component labelling algorithm is used. In order to identify the exact area of the gesture provided Convex-Hull method is used where the arm area is removed. This provides a real time demonstration system [6].

Shriram S and Nagaraj B proposed a paper "Deep Learning-Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread". This method was proposed in order to avoid spreading covid-19. This AI virtual mouse uses transformational algorithm which converts the co-ordinates of fingertips from the webcam to computer window for controlling the mouse events. When the hands are detected, a rectangular box is drawn with respect to the window in the webcam where the mouse movement is implemented using the cursor [7].

In 2021, Tanzil Shahria and Samiul Haque Sunny proposed a paper "Mapping and Localization in 3D Space for Vision-Based Robot Manipulation" where a robot is used to locate any object in the 3D space. It uses a set of RGB camera, depth sensor and vision processor that allows to get the coordinates of the objects present in 3D space in terms of the camera in order to map and localize the object [8].

Nirvika Rajendra, Aishwarya R and Roopa Shree N K proposed a method where the cursor events are controlled by object detection based on Image processing. The user would hold a green coloured object which is placed in the viewing area of the camera and mouse events are controlled using it [10].

3. Methodology

The methodology employed in this research focuses on developing a gesture-based control system for the operating system. The system architecture includes a hardware setup consisting of an inbuilt or external camera for capturing video frames of hand gestures, along with a software setup utilizing Python programming language and libraries such as mediapipe, OpenCV, autopy, and pyautogui. The hand gesture recognition process involves video processing using the mediapipe library to detect human hands accurately. OpenCV is then utilized to extract relevant features from the hand region in the video frames, including finger positions, palm orientation, and hand movements. Machine learning techniques, such as decision trees or neural networks, are applied for gesture classification based on the extracted features. The recognized hand gestures are mapped to specific cursor events, such as left-click, right-click, drag and drop, or scrolling. Cursor control is achieved through the execution of cursor events using the autopy and pyautogui libraries. The developed system undergoes iterative testing and refinement, evaluating its accuracy and responsiveness with various hand gestures in different scenarios, including common cursor actions and complex gestures. Refinement involves adjusting parameters, enhancing gesture recognition algorithms, and improving the overall user experience. By following this methodology, our aim is to create an effective and intuitive gesture-based control system that enhances Human-Computer Interaction in operating systems. The workflow of the software created is explained and depicted in Fig. 1.

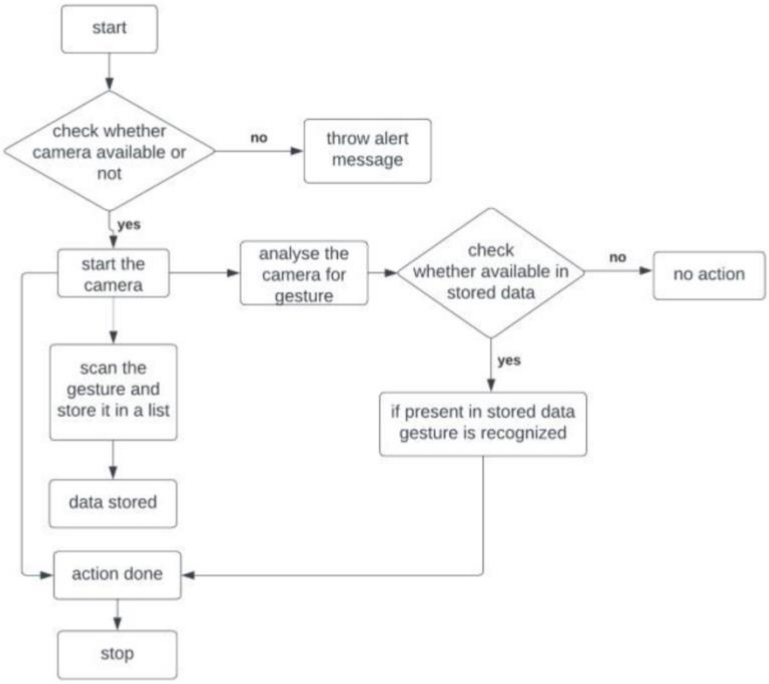

start
check whether camera available or not
no
throw alert message
yes
start the camera
analyse the camera for gesture
check whether available in stored data
no
no action
yes
scan the gesture and store it in a list
if present in stored data gesture is recognized
data stored
action done
stop

Fig. 1. Flow chart of proposed system.

3.1. Camera

The proposed virtual mouse system is based on the frames that have been captured by the integrated camera in a laptop or PC. At first the program starts running the camera to capture images. The system would capture the video frames in real-time using the webcam.

3.2. Capturing

By using an infinite loop, the frames are captured till program termination. Then the captured frames are processed into RGB colour space.

3.3. Hand Tracking

A loop is started to continuously read the frames for processing hand tracking in each frame. Image is processed in RGB format and returns the hand landmarks. The landmarks are visualized on the frame and the frame rate is calculated. It utilizes a pre-trained neural network for detecting the hand landmarks which includes fingers, palms and wrists. By tracking the movement of the landmarks, gestures and the finger positions can be recognized. From that the events are identified and implemented. In order to make the application function, mediapipe is utilized, and mediapipe acts as the main algorithm to deal with the data acquired using the camera. Mediapipe is used to work with the data collected from the built-in camera in the systems. The data gathered by the system's built-in camera was used by mediapipe to operate. It primarily uses the information gathered from the built-in camera to do image processing. Data must be provided in the application as mediapipe requests it. With the aid of the opencv library, the data is extracted from the video. With opencv, we may manipulate the pixel values directly, access picture characteristics, choose a specific region from an image, split images, and more. Hence, in order to pick a region to focus on, gather data from it, and feed it information, we need information from our hands, specifically the locations of each finger and other parts of our hands. It is a Python GUI automation library. It contains options for managing keyboard and mouse functionality. Additionally, it does other tasks like alert display, colour detection, and more. But operating the mouse is our primary necessity. We can therefore obtain the mouse's functionality using autopy, set gesture values to it, and use it. For instance, if a user does a right click, the user's input gesture will be assigned to that functionality in accordance, and if the user repeats that gesture, the assigned functionality is performed.

4. Gesture recognition

As the built system works according to the user gesture input, we can take a set of mocked data as user input let the data be:

$$[{}^{1}[1,1,0,0,0],\ {}^{2}[0,1,0,0,0],\ {}^{3}[1,1,1,0,0],\ {}^{4}[0,0,0,0]]$$

4.1. Distancecalculation

The distance is calculated between two specific landmarks on the hand. It takes landmark indices, image and drawing parameters as input. Inside the method the x and y coordinates of the specified landmarks are extracted from a list of landmarks. For this we use the Euclidean distance formula:

$$\text{Distance} = \sqrt{((x2 - x1)^2 + (y2 - y1)^2)} \qquad (1)$$

This formula calculates the distance between (x1, x2) and (y1, y2). This calculated distance is returned along with the modified image and the coordinates of the two landmarks.

4.2. Mousemovement^{1}

The proposed system uses raising of index and thumbs finger for mouse movement. The hands in the captured frames are detected and tracked and the resulting image with annotated landmarks is stored. Then the positions of specific landmarks are identified, and values are returned. The state of the fingers is determined and are moved into a list. So, while showing the specific sign for the mouse movement it is checked with the stored data and gets implemented.

4.3. Right click^{2}

For right click the system uses raising of the index finger in order to perform the right click operation. The same procedure is followed, the landmarks are stored, and the positions are identified then the state is determined and moved into a list. While showing the gesture the data is checked with the stored data and the event is implemented.

4.4. Left click^{3}

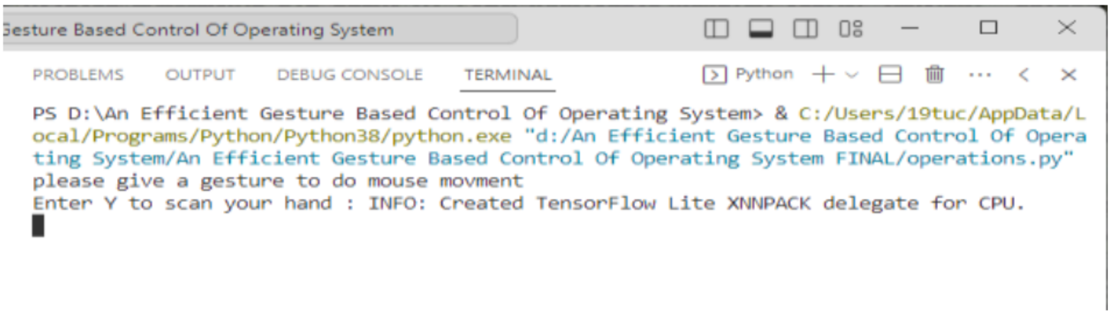
In order to implement left click the index, middle and thumbs are raised. It also consists of the same previous procedures, when the gestures are shown it is checked with the data stored and the event gets implemented.

4.5. Doubleclick[4]

In order to implement double click all the fingers should be closed. It involves the same process and steps. When the gesture is shown it checks with the stored data and implements the cursor event.

5. Result & Evaluation

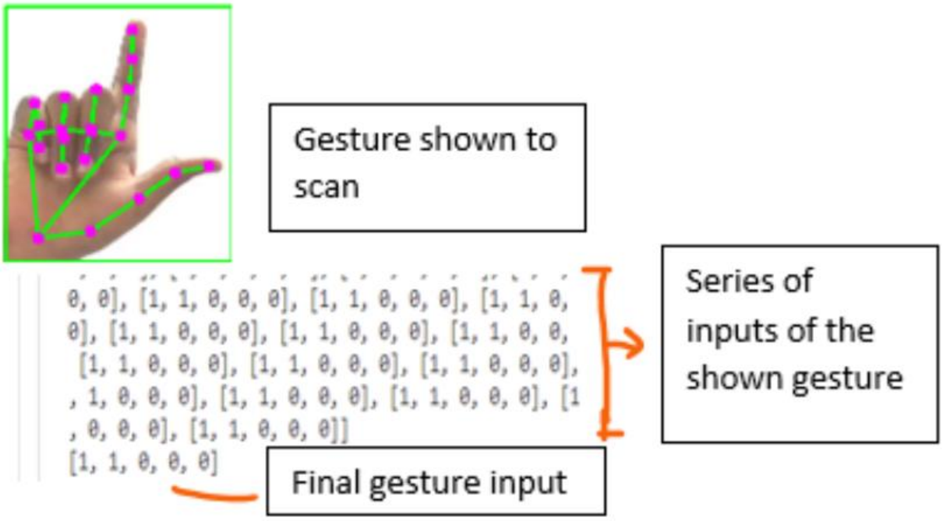
The proposed paper focuses on improving HCI (Human Computer Interaction). The main mouse events implemented are mouse movement, right click, left click and double click. At first, we start to run our application. when the application starts, it will start to ask input for the mouse features in the bellow image it asks gesture input for mouse movement.



```
PS D:\An Efficient Gesture Based Control Of Operating System> & C:/Users/19tuc/AppData/L
ocal/Programs/Python/Python38/python.exe "d:/An Efficient Gesture Based Control Of Opera
ting System/An Efficient Gesture Based Control Of Operating System FINAL/operations.py"
please give a gesture to do mouse movment
Enter Y to scan your hand : INFO: Created TensorFlow Lite XNNPACK delegate for CPU.
█
```

Fig. 2. gesture input console image.

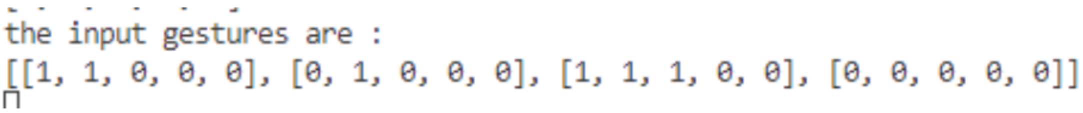
So the user can type y(yes) when he is ready and start scanning his gesture in front of the camera the application starts to scan, recognize the hands gives some series outputs and take a mean of the output and the application sets that gesture for that mouse operation in the below image it shows a series of output and its final result gesture too.



```
0, 0], [1, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1, 1, 0,
0], [1, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1, 1, 0, 0,
 [1, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1, 1, 0, 0, 0],
, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1, 1, 0, 0, 0], [1
, 0, 0, 0], [1, 1, 0, 0, 0]]
[1, 1, 0, 0, 0]
```

Fig. 3. Gesture scanned and determined the input.

Gesture shown are scanned and converted into list which has 1's and 0's in it. And this list is stored in a group of lists. Now these lists of gesture input are iterated and each gesture in the list is assigned to the operations of the mouse like right click, left click, drag and drop operations.



```
the input gestures are :
[[1, 1, 0, 0, 0], [0, 1, 0, 0, 0], [1, 1, 1, 0, 0], [0, 0, 0, 0, 0]]
```

Fig. 4. All gestures grouped into 2d list.

So according to the above results the application will iterate through these inputs and the application will start to assign this list of gesture according to the mouse operations. The shown gesture for the above results is explained in the below images (Fig. 5., Fig. 6., Fig. 7., Fig. 8.).

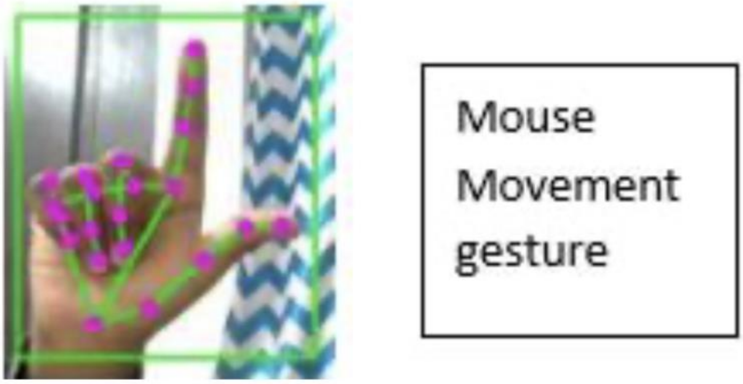


Fig. 5. Gesture given by a user to the application.

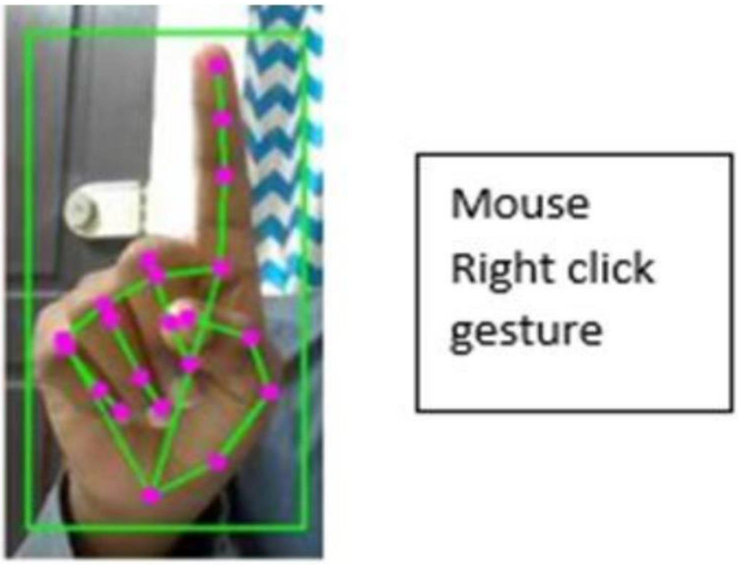


Fig. 6. Mouse right click event gesture.

In the above image this gesture is set for right click operation.

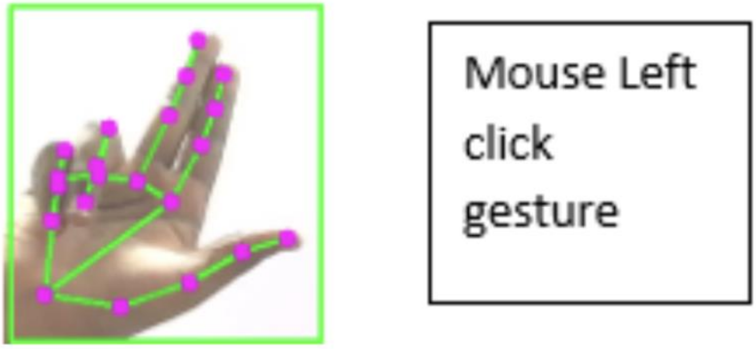


Fig. 7. Mouse left click event gesture.

In the above image the gesture set to the left click operation of the mouse.

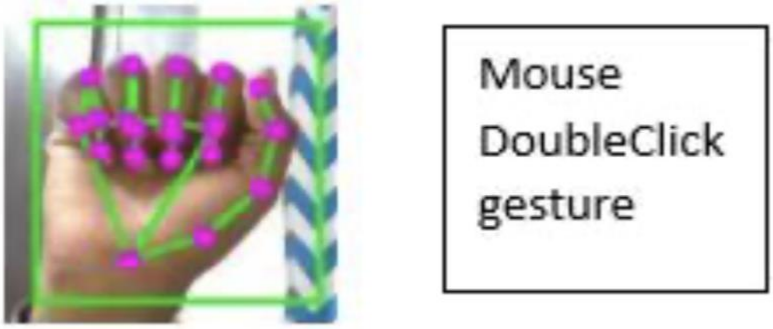
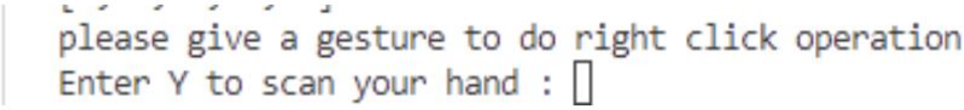


Fig. 8. Mouse Double click event gesture.

In the Fig. 8. the double click event gesture input is given by the user and the application sets that gesture to that operation of the gesture. In this application the user will be notified that for which operation the user is giving input so he/she doesn't need to get confused while giving input to the application the notification is shown in the bellow image.



```
please give a gesture to do right click operation
Enter Y to scan your hand : 
```

Fig. 9. message that mentions about for which operation we are going to set a gesture.

According to the above image this is how the user will be notified about which operation he is going to give input. The testing is done in a plain background. It is also tested in a complex/non-plain background such as bright colored t- shirts/regular shirts, complex patterns of background. These are measured based on accuracy produced which is shown in the below table.

TABLE 1. Evaluation of cursor events

Evaluation		
Event	***Plain bg accuracy (%)***	***Complex bg accuracy (%)***
Cursor events	63	80

From the above table we can infer that the provided method operates efficiently in complex backgrounds than in plain backgrounds. In order to improve the accuracy in plain background, background subtraction techniques can be applied for extracting the hand region from the plain background. The accuracy may also be higher when used in other OS.

6. Future work and conclusion

The proposed system can be further refined in the future as this project main aim is to increase the efficiency to control the operating system and other electronic devices. So, as the package mediapipe supports other actions like body gesture, full hand gesture, eye contact gestures and facial gesture this can make the communication between human and computer more interactive.

The main objective of the virtual mouse system is to control the mouse functions by using hand gestures instead of using a physical mouse. The proposed system can be achieved by using a webcam or a built-in camera that detects the hand gestures and hand tips and processes these frames to perform the particular mouse functions. Based on the analysed results this system can be used for developing HCI (Human Computer Interaction).

7. Acknowledge

We would thank our institution and our faculty guide for extending their fullest cooperation and thankful guidance in completing the project work.

8. References

8.1. Conference Proceedings

[1] *Hu, C., Liang, L., Ma, S., & Lu, H. (2000, October). Virtual Mouse— inputting device by hand gesture tracking and recognition. In International Conference on Multimodal Interfaces (pp. 88-95). Springer, Berlin, Heidelberg.*

[2] *Aksaç, A., Öztürk, O., & Özyer, T. (2011, December). Real-time multiobjective hand posture/gesture recognition by using distance classifiers and finite state machine for virtual mouse operations. In 2011 7th International Conference on Electrical and Electronics Engineering (ELECO) (pp. II-457). IEEE.*

[3] *Farooq, J., & Ali, M. B. (2014, April). Real time hand gesture recognition for computer interaction. In 2014 International Conference on Robotics and Emerging Allied Technologies in Engineering (iCREATE) (pp. 73-77). IEEE.*

[4] *Tsai, T. H., Huang, C. C., & Zhang, K. L. (2015, June). Embedded virtual mouse system by using hand gesture recognition. In 2015 IEEE International Conference on Consumer Electronics-Taiwan (pp. 352353). IEEE.*

[5] *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.*

[6] *Tsai, T.-H., Huang, C.-C., & Zhang, K.-L. (2019). Design of hand gesture recognition system for human-computer interaction. Multimedia Tools and Applications.*

[7] Shahria MT, Sunny MS, Zarif MI, Ahamed SI, Rahman MH. Mapping and Localization in 3D Space for Vision-based Robot Manipulation.

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

8.2. Journal Article

[9] Shriram S, Nagaraj B, Jaya J, Shankar S, Ajay P. Deep learning-based real-time AI virtual mouse system using computer vision to avoid COVID-19 spread. Journal of healthcare engineering. 2021 Oct 25;2021.

[10] Rajendra, Nirvika, et al. "Cursor Movement by Object Detection Based on Image Processing." Journal homepage: www.ijrpr.com ISSN 2582 (2021): 7421.