International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 05 | May - 2023 Impact Factor: 8.176 ISSN: 2582-3930

VIRTUAL MOUSE IMPLIMENTATION USING OPEN COMPUTER VISION

N. Srujan[1], MD. Afreed[2,] K. Rathnakar[3], K. Harshith Reddy[4], PL Pradhan[5]

[1,2,3,4]*B.Tech. Student, Department of Computer Science and Engineering,*

srujannagapuri121@gmail.com , mohdafreed06@gmail.com,

rathnakark6866@gmail.com , harshithreddy3008@gmail.com

pradhan.cse@nmrec.edu.in

[5]*Assistant Professor, Department of Computer Science and Engineering,*
Nalla Malla Reddy Engineering College, Hyderabad, India

Abstract— we are going to be taken care of virtual mouse on Human Computer Interaction (HCI). A virtual mouse is a device used to air-browse the functions of a system such as computer, laptop or a smart-pad with the functions corresponding to a mouse and a marker. It can be used in class-rooms for teaching without the board and chalk. This can also be used in conferences as a smart device to work as a mouse. A person is just required to make the corresponding gestures imitating a mouse or a marker in front of an Infra-red (IR) camera using this hardware device which is modeled as a pen and this avoids the use of wired mouse, chalks and boards. A virtual marker is already implemented by using software to process the images captured to get the coordinates. Based on the idea of virtual marker we are developing virtual mouse that can control the system by showing out in front of web camera and hand gesture recognition can be useful for all kinds of people.

Keywords—Human computer interaction, Hand gesture recognition, Open computer vision, Finger tracking, Image processing.

1. INTRODUCTION

A Computer Mouse is an input device that helps to point and to interact with whatever that is being pointed. There are so many types of mouse in the current trend, there's the mechanical mouse that consists of a single rubber ball which can rotate in any direction and the movement of the pointer is determined by the motion of that rubber ball. Later the mechanical mouse is replaced by the Optical Mouse.

No Matter how much the accuracy of the mouse increases but there will always be limitations of the mouse as the mouse is a hardware input device and there can be some problems like mouse click not functioning properly ad etc., as the mouseis a hardware device like any other physical object even the mouse will have a durability time within which is functional and after its durability time we have to change the mouse. such as speech recognition. Speech Recognition is used forrecognition and translation of the spoken language into text. Thus, Speech Recognition can replace keyboards in the future, Similarly Eye Tracking. which is used to control the mouse pointer with the help of our eye. Eye Tracking can replace mouse in the future.

Gestures can be in any form like hand image or pixel image or any human given pose that require less computational difficulty or power for making the devices required for the recognitions to make work. Different techniques are being proposed by the companies for gaining necessary information/data for recognition handmade gestures recognition models. Some models work with special devices such as data glove devices and color caps to develop a complex information about gesture provided by theuser/human.

2. EXISTING METHODOLOGY

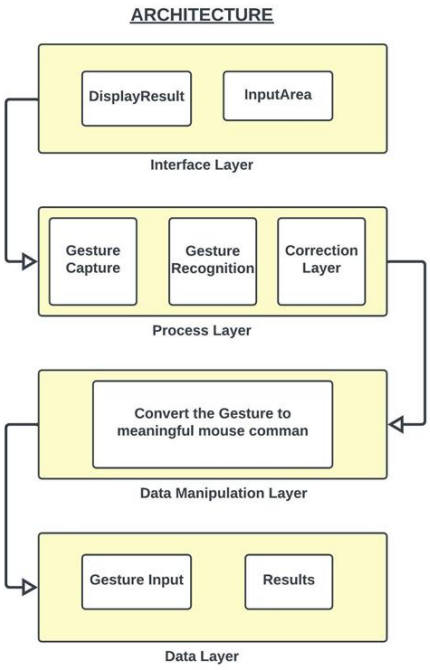
The existing system as physical mouse which is wired, wireless and cordless. There are some virtual mouse systems which controls the system by hand gestures. Here the virtual mouse is controlled with fingers by using color caps to fingers. So the existing system is

not user friendly and not secure. By using the idea of existing system we are developing proposed system which is more secure and user friendly. The existing system contains a mouse that may be wireless or cordless to control the cursor, be aware that we may use hand gestures to monitor the system. The existing visual mouse control system incorporates simple mouse functionality using colored tips for web camera capture, which is why colored fingers act as a web camera color sensor such as red, green, blue to monitor the system, while it can perform basic mouse functions such as drag, drag, drag up, scroll down, left-click with the right-click using a finger touch without any colored finger because the skin color detection system is more flexible than the existing system. In an existing system use vertical hand recognition such as fingerprint identification, hand shape, number of fingers in a clearly defined action, making the system very difficult to understand and difficult to use.

3. PROPOSED METHODOLOGY

The system contains a mouse that may be wired and wireless to control the cursor. Even though there are some virtual mouse which is controlled by using hand gestures. The functionality of existing virtual mouse is done by using colored caps for user fingers for web camera capture. Now proposed system is that we will not use any wired mouse. We are developing a virtual mouse to control the system. In proposed system the used should not require to use any color caps to fingers in order to control the system. Without using the colored caps to fingers for web camera capture we are developing this project. It is more secure, usable, accurate and user friendly

4. SYSTEM ARCHITECTURE



5. ADVANTAGES

- More accurate
- No space required
- Ease of use

6. DISADVANTAGES

- Usually not as secure as mouse
- Not very user friendly
- Requires more maintenance

7. MODULES

➢ **Image Processing Module:** This module would involve capturing an image or video feed of the user's hand or fingers using a camera, and then processing that image to detect the user's finger movements and gestures.

- **Recognition Module:**
  This module would interpret the data from the image processing module and identify specific finger gestures, such as clicking, scrolling, or dragging.
- **Mouse Emulation Module:**
  This module would translate the output from the gesture recognition module into commands that emulate the behaviour of a computer mouse, such as moving the cursor or clicking on buttons.
- **Communication Module:**
  This module would establish a connection between the finger tracking system and the computer or device being controlled, such as through Bluetooth or USB.
- **Calibration Module:**
  This module would allow the user to calibrate the finger tracking system for their own hand size and finger movements, in order to achieve more accurate and reliable control of the computer mouse.

8. SYSTEM IMPLEMENTATION

The system working conditions are based on Anaconda Environment interface design, with OpenCV, wx, Numpy libraries and some of the sub packages of these libraries. Camera resolution is 1920*1080 and with fps of 40 Default user. The model is designed for the recognition and further working is done by the commands given to the system and howthe user wants the gestures to make recognize. At the same time, the mouse pointer movement will be captured and made operate without any human interaction. The model working and implementation is shown in the figures.

```
while True:
    ret, img=cam.read()
    img=cv2.resize(img,(340,220))

    #convert BGR to HSV
    imgHSV= cv2.cvtColor(img,cv2.COLOR_BGR2HSV)
    # create the Mask
    mask=cv2.inRange(imgHSV,lowerBound,upperBound)
    #morphology
    maskOpen=cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernelOpen)
    maskClose=cv2.morphologyEx(maskOpen,cv2.MORPH_CLOSE,kernelClose)

    maskFinal=maskClose
    conts,h=cv2.findContours(maskFinal.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)

    cv2.drawContours(img,conts,-1,(255,0,0),3)
    for i in range(len(conts)):
        x,y,w,h=cv2.boundingRect(conts[i])
        cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255), 2)
        cv2.cv.PutText(cv2.cv.fromarray(img), str(i+1),(x,y+h),font,(0,255,255))
    cv2.imshow("maskClose",maskClose)
    cv2.imshow("maskOpen",maskOpen)
    cv2.imshow("mask",mask)
    cv2.imshow("cam",img)
    cv2.waitKey(10)
```

Fig. 1. This is the part of the code that is used for the color detection and highlighting the wanted color.

A. Model Understanding

The main thing we need to identify are the applications the model is going to develop so the development of the mouse movement without using the system mouse, the model is developed using computer vision where the color variation isused for the mouse identification and movement . We cansee the fig 1 to identify the color variation / color detection that helps with the main part of our problem.

```
lowerBound=np.array([33,80,40])
upperBound=np.array([102,255,255])
```

Fig. 2. Limits of Green color in HSV color identification (Hue, Saturation,and Value).

```
mask=cv2.inRange(imgHSV,lowerBound,upperBound)
```

Fig. 3. For adding the limits to the mask, so that it can identify the greenfrom the camera and highlights the input from the user.

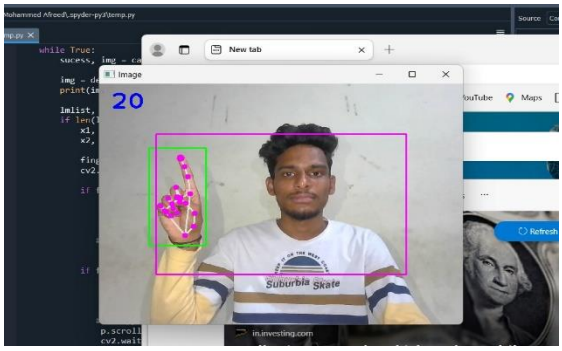
The model creation depends mostly on the steps that are given in the Fig 2 and 3. There we are identifying the color required for the user by using the range of green by using thecolor values in RGB image and highlighting them from HSVimage and converting them into Black and White and displayit in the mask which is system understandable image representation. After recognizing the highlighted color from the Fig 1, it's shown that formation of red boxes around the color.

```
elif(len(conts)==1):   #case for a single rectangle

    if pinchFlag==0:

        pinchFlag=1

        mouse.press(Button.left)    #reducing mutiple clicks

    x,y,w,h=cv2.boundingRect(conts[0])

    cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)

    cx=int(x+w/2)

    cy=int(y+h/2)
    #plotting a big circle with centre as rectangle's centre
    cv2.circle(img,(cx,cy),int((w+h)/4),(0,0,255),2)

    mouseLoc = mLocOld + ((cx, cy) - mLocOld) / DampingFactor

    mouse.position = (sx - (mouseLoc[0] * sx / camx), mouseLoc[1] * sy / camy)

    mLocOld = mouseLoc
cv2.imshow("cam",img)   #Cam OP



cv2.waitKey(5)
```

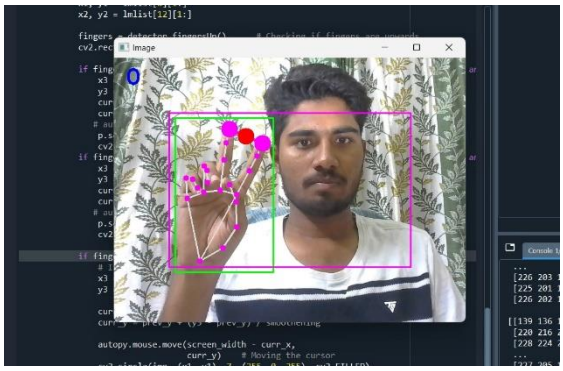
Fig. 4. Code for operating if there is only one figure with highlighted color

9. RESULT ANALYSIS

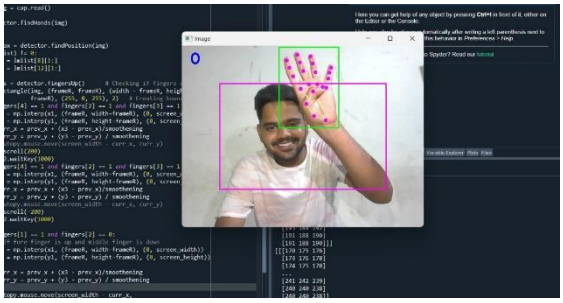
The model which we have developed by using open CV and Numpy. The models which we are developing are color detection and mouse movement based on highlighted color which is given from the user for the movement of the mouse. Let's see how the output is generated.



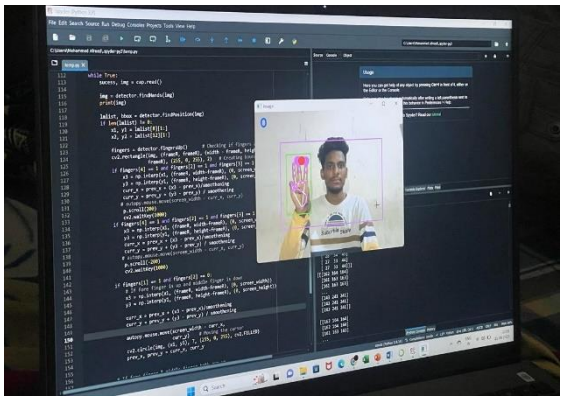
Cursor movement



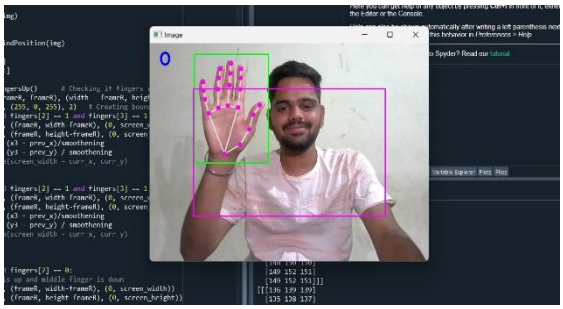
Right click



Left click



Moving downwards



Moving upwards

10.CONCLUSION

This model can conclude by using the topics of computer vision like open CV, it can form masks that can variety colors by using color variation techniques and also development of mouse movement by using certain packages like 'mouse' which will be used for the movement of mouse by using the coordinates that are linked to the detected color.
This can provide ease use of systems and many other applications. So the open CV is helping the users with different accessible forms of models that will make ease life.

10. FUTURE SCOPE

The development of these techniques and models are really vast. The color detection model can be developed if we wantto identify a particular color out of a colored photo. And the mouse movement can be developed in such a way it can act like a real mouse that will help us for using system without even touching the system's keyboard or mouse. The development can be in such a way it can be training on CNN's that will help for a better performed model. The Models can be developed in different ways by using some latest packages like 'pyauto GUI' that will help usto give commands which will identify an input and perform some function on the system. So if any separate color is detected it can perform special function or if an input fromuser is detected it will open any specific folder with ease without performing any actions, a simple gesture can do the job.

11. REFERNCES

[1] Guoli Wang, (2010). Optical Mouse Sensor-Based Laser Spot Tracking for HCI Input, Proceedings of the 2015 Chinese Intelligent Systems Conference: Volume 2, pp.329-340.

[1] Guoli Wang, (2010). Optical Mouse Sensor-Based Laser Spot Tracking for HCI Input, Proceedings of the 2015 Chinese Intelligent Systems Conference: Volume 2, pp.329-340.

[2] Anna De Liddo, Ágnes Sándor, et.al, (2012). Contested Collective Intelligence: Rationale, Technologies, and a Human-Machine Annotation. Computer Supported Cooperative Work (CSCW) Volume 21, Issue 4–5, pp 417–448.

[3] Rashmi Adatkar, Ronak Joshi, et.al, (2017). Virtual Mouse, Imperial Journal of Interdisciplinary Research (IJIR), Vol-3, Issue-4.

[4] Arul. V. H, Dr. Ramalatha Marimuthu, (2014). A Study on Speech Recognition Technology, Journal of Computing Technologies, Volume 3 Issue 7, pp 2278 – 3814.

[5] Aniwat Juhong, T. Treebupachatsakul, et.al, (2018). Smart eye-tracking system. 2018 International Workshop on Advanced Image Technology (IWAIT).

[6] Guojen Wen, Zhiwei Tong, et.al, (2009), Man machine interaction in machining center. International workshop on intelligent systems and applications. pp 1-4.

[7] S.D. Bharkad, et.al. (2017). international conference on computing methodologies and communication, pp 1151-1155.

[8] Litong Fan, Zhongli Wang, Baigen Cail, et.al (2016). A survey on multiple object tracking algorithm. 2016 IEEE International Conference on Information and Automation (ICIA)

[9] Pritpal Singh, B.B.V.L. Deepak, Tanjot Sethi and Meta Dev Prasad Murthy (2015). Real-Time Object Detection and Tracking Using Color Feature and Motion. International Conference on Communication and Signal Processing.

[10] G. Saravanan, G. Yamuna, S. Nandhini (2016). Real time implementation of RGB to HSV/HSI/HSL and its reverse color space models. 2016 International Conference on Communication and Signal Pe-tracking system. 201d applications. pp 1-4.

[11] S.D. Bharkad, et.al. (2017). international conference on computing methodologies and communication, pp 1151-1155.

[12] Litong Fan, Zhongli Wang, Baigen Cail, et.al (2016). A survey on multiple object tracking algorithm. 2016 IEEE International Conference on Information and Automation (ICIA)

[13] Pritpal Singh, B.B.V