INTERNATIONAL JOURNAL OF
RECENT TRENDS IN MULTIDISCIPLINARY RESEARCH
Special Issue, March-April 2023, Vol 3 (02), 87-90.
International conference on Innovation towards Sustainable Development Goals-**ICISDG'23**
ISSN No: 2583 0368

Performance Evaluation of Virtual Mouse Using AI

Naman Tiwari[1], Piyush Tiwari[2], Akruti Sonawane[3], Rupali Satpute[4]
[1,2,3]*Electronics and Telecommunication, K J Somaiya Institute of Technology, India.*
[4] *Professor, Electronics and Telecommunication, K J Somaiya Institute of Technology, India.*

Article Type: Research

OPENACCESS
Article Citation:
Naman Tiwari[1], Piyush Tiwari[2], Akruti Sonawane[3], Rupali Satpute[4], "Performance Evaluation of Virtual Mouse Using AI", International Journal of Recent Trends In Multidisciplinary Research, March-April 2023, Vol 3(02), 87-90.

Check for updates

https://www.doi.org/10.59256/ijrtmr.20230402c19

Abstract: This paper proposes a way of controlling the position of the mouse cursor with the help of a fingertip without using any electronic device. We can perform the operations like clicking and dragging objects easily with help of different hand gestures. The only necessary input device for the suggested system is a webcam. The software will need to be implemented using Python and OpenCV. The output of the webcam will be shown on the computer's screen so that the user can fine-tune it. NumPy, Autopy, and Media pipe are the Python requirements that will be utilised to create this system.
Key Word: Open Cv, Autopy, Mediapipe Numpy; Calibrated, Gesture.

1. Introduction

The paper demonstrates the functioning of an AI virtual mouse system that accepts hand gestures as input via an external camera or a webcam attached to a laptop. With the aid of this detection of finger movement for improved mouse operations in a computer by employing computer vision. The suggested system's primary goal is to use an integrated camera to perform mouse actions such single clicks, double clicks, right clicks, drags, scrolls, drops, and selects. The term "human-computer interaction," or HCI, refers to the use of hand motions and fingertips for touch detection. Not only do we need to charge the battery cell to power it at some time while using a wireless or Bluetooth device, like a mouse. In this system, all computer mouse activities are controlled by user hand gestures. The webcam records motions through the image, stores them temporarily, analyses the frames, and may identify predetermined hand gesture operations. The most popular programming language for this kind of virtual mouse system is Python. Contract computer visions also use the OpenCV libraries. For tracking fingertips in real-time, it uses well-known Python tools as MediaPipe and Autopy. Additionally, the Pyput and PyAutoGUI packages were used to monitor user fingertips as they moved around the screen to perform actions like drag, scroll, and left and right clicks. Our goal is to virtually alter our lives in order to make them simpler and less time-consuming.

2. Methodology

A. Media pipe

It is a cross-platform framework developed by Google that is mostly employed for the construction of multimodal pipelines in machine learning. Since the MediaPipe framework is built using statistical data, it is most helpful for cross-platform development work. The MediaPipe framework is multimodal, which means it can be used for motion tracking, object detection, face detection, face mesh, iris scanner position detection, hand detection, hair segmentation, and objection detection.The ideal option for developers is the MediaPipe framework, which has been used to create a variety of cross-platform systems and applications as well as to build, evaluate, and develop systems performance in the form of graphs (Android, IOS, web, edges devices). Three reliabilities are provided by the MediaPipe package: performance, evaluation, and and security—are provided by the MediaPipe package.

B. Autopy

It is a Python cross-platform GUI automation module. This module in the system under consideration maintains track of each finger. Autopy can detect which finger is up and which is down by tracking the fingertip. Giving the system an input in the form of 0 and 1 causes this procedure to take place. The media pipe module accepts the output from this module, processes it, and outputs the right data. OpenCV derived this output.

C. Open CV

A package for computer vision called OpenCV has image-processing tools for object detection. Real-time computer vision applications can be created by utilizing the OpenCV library for the Python programming language. Processing of images and videos as well as analytical techniques like face and object detection use the Open CV library.

D. Importing the necessary Library:

Install and import all required libraries in your choice IDE first. OpenCV, NumPy, media pipe, and the autopy framework are to be installed. After successfully install all the library in PyCharm, our next is to import all that library into the code section so that we can make use this library in our system.

E. Initializing the capturing device:

The next task is to initialize the connected image capture device. We are making use of system primary camera as cap = cv2.VideoCapture(0). if any secondary image capturing device connected that is not used by this program.

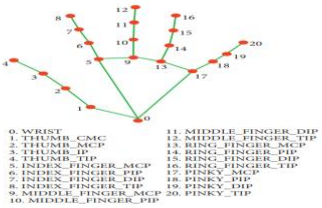
F. Capturing the frames and processing:

The proposed AI virtual mouse system uses primary camera for capturing the frames. The video frames are processed and converted BGR to RGB format to find the hand in video frame. This color conversion is done by the following code: check.

```
img = cap.read()
imgRGB = cv2.cvtcolor(img. cv2.COLOR_BGR2RGB) lmlist = handLandmarks(imgRGB)
 if len(lmlist) !=0:
 finger = fingers(lmlist)
 x1, y1 = lmlist[0][1:]
 x2, y2 = lmlist[12][1:]
 x3 = numpy.interp(x1, (75, 640 - 75), (0, wScr))
 y3 = numpy.interp(y1, (75, 480 - 75), (0, hScr))
```

G. Initializing media pipe and capture window:

After the Initialization of media pipe is done, which is responsible for tracking the hand gesture and other actions. Then we are setting the minimum confidence for hand detection and minimum confidence for hand tracking by main Hand=initHand.Hands(min_detection_confidence=0.8, min_tracking_confidence=0.8). This shows that the major hand has an 80% success rate while performing the activity. The media pipe is then used to draw the axis line across the palm and centre of the fingers. that keeps track of finger motion and connects to neighbouring fingers



3. Result

Our next objective is to specify the whole hand land mark capture after the hand axis has been verified. The media pipe is responsible for tracking these 21 joints or identifying points. If no landmark is located, the empty default value is printed. That indicates that nothing needs to be done. If not, it returns the X, Y, and Z axis coordinate value. The hand landmark region does not contain the thumb finger. By drawing a graph line between each finger, we can create a loop between 21 points and return the value of the tip point. The lines are also visible when we fold the finger down or down, just like the matching graph line. The distance between the fingers doesn't matter at this point. The distance between fingers can increase to 5 cm. If not, it returns to the starting place of discovering. We transform the decimal coordinate relative to the index based on each image after determining the height, width, and center of the hand landmark. A green rectangular box enclosing the hand where we can operate the mouse displayed after capturing all of the hand features.

There were various problems with the clicking and dragging method of selecting text. It will be possible for the user to easily manage keyword functionality and human-computer interface if some barriers are removed and new ways are developed (HCI). We investigate the system's performance in difficult circumstances, at the maximum user-camera distances, and in low-light conditions processing blurry images and detecting one person's hand at a time. This study uses a single CPU and doesn't need an extra GPU system for support.

4. Discussion

The scope of this project is to develop AI based Virtual mouse that will help us to operate without touching any devices or screens. In this digital era we are adjusting our living while being in pandemic, a touchless mouse controller will be useful to prevent the risk of spreading such viral virus infections, this Virtual mouse could be used. There were some difficulties in execution of clicking and dragging to select text. Some of the limitations will be overcome, new method are developed so that it would be easy for the user to handle the keyword functionalities along with the Human computer Interaction HCI. With further research we investigate that the system in complicated cases, maximum distant between user and camera, low light condition, blury image processing, and can detect one person hand at a time. This studies uses only single CPU and does not require any external support GPU system. The accuracy is low for "Right Click" as this is the hardest gesture for the computer to understand. The accuracy for right click is low because the gesture used for performing the particular mouse function is harder. Also, the accuracy is very good and high for all the other gestures. Compared to previous approaches for virtual mouse, our model worked very well with 99% accuracy. A comparison between the existing models and the proposed AI virtual mouse model in terms of accuracy. It is evident that the proposed AI virtual mouse has performed very well in terms of accuracy when compared to the other virtual mouse models. The novelty of the proposed model is that it can perform most of the mouse functions such as left click, right click, scroll up, scroll down, and mouse cursor movement using finger tip detection, and also, the model is helpful in controlling the PC like a physical mouse but in the virtual mode.

5. Conclusion

The main objective of the AI Virtual Mouse system is to eliminate the need for a hardware mouse by allowing users to control the cursor with hand movements. The suggested device uses a built-in camera or webcam to use computer vision to recognize hand movements and fingertips. The specific mouse function is carried out by these frames. From the findings, we concluded that the new AI Virtual mouse is much more effective than the current model and prevents most of its shortcomings. Therefore, real-time applications are possible with AI Virtual Mouse. It has similar restrictions, such as a little loss of accuracy in right-click mouse functionality. The technology is suggested for home automation, allowing for the control of tubes with gestures.

References

[1] 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, July 2010.

[2] 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.

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

[4] Google, MP, https://ai.googleblog.com/2019/08/on-devicereal-time-hand-tracking-with.html

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

[6] Google, MP, https://ai.googleblog.com/2019/08/on-devicereal-timehand-tracking-with.html.

[7] Tsung-Han Tsai, Chih-Chi Huang and Kung-Long Zhang, "Embedded Virtual Mouse System by Using Hand Gesture Recognition", 2015

[8] Zhou Ren, Junsong Yuan, Jingjing Meng, andZhengyou Zhang, "Robust Part-Based Hand Gesture Recognition Using Kinect Sensor", pp. 1110, vol15, August 2013

[9] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek, "A brief introduction to OpenCV", July 2012

[10] Viraj Mali, Ankit Gorasi, Meghna Patil, Pavan Wawage, "Home Automation and Security using arduino micro-controller", pp 215, March 2016