Virtual Mouse Using OpenCV

Shreyansh Singh
Galgotias University
Greater Noida, India
shreyansh_singh.scsebtech@galgotiasu
niversity.edu.in

Abstract — Vision-based automatic hand gesture recognition has been a greatly effective exploration point as of late with persuading applications such as human-computer interaction (HCI), robot control, and gesture-based communication translation. Attacking the matter in its generality requires elaborate algorithms requiring intensive computer resources. Something that drives everyone for the task may be a humanoid navigation predicament, in which we have an advantage in managing a robot by supporting pose symbols given through a body. Due to real-time operational requirements, we have an interest in a computationally efficient algorithm. Hand Gesture Recognition performs a fundamental position in human-computer intercommunications. As we can see that there are so many new Technological advancements happening such as biometric authentication which we can see frequently in our smartphones, similarly hand gesture recognition is a modern way of human-computer interaction i.e., we will control our system by showing our hands ahead of webcam and hand gesture recognition is often useful for all kinds of people. Within this paper, a description of a computer vision-based mouse is given that can be controlled and command the cursor of a computer or an automated system using hands without any physical contact with the system using libraries such as OpenCV, Mediapipe, AutoPy, NumPy. In this system, there's no got to have a cable connection between the pc and mouse nor a wireless transmitter-receiver pair because the mouse movements are transferred to the computer by the camera. The concept of a vision-based finger mouse is first proposed by Quek et al. in which the user controls the cursor by moving his fingers within the three-dimensional space. In our mouse system, the approach is simpler than in the sense that we place specific reference points on the mouse and the mouse moves on a two-dimensional surface. We follow evidence features and the position of the cursor is updated correspondingly. This approach is computationally more efficient than finding the fingertip in a cluttered background and tracking it. In addition to the above features, our mouse has well-defined regions corresponding to buttons to implement clicking. To click a switch the user solely covers a button area to some measure by a person's finger or a pointer. Through this idea, the paper is impersonated. This paper provides a depth interpretation of the methodologies and algorithms for virtual mouse and colour detection.

1. INTRODUCTION

In human-computer interactions, hand gesture recognition plays a really vital role. As we'll see that there are numerous new Technological advancements happening like biometric authentication which we'll see frequently in our smartphones, similarly hand gesture recognition may be a modern way of human-computer interaction i.e., we'll control our system by showing our hands before webcam and hand gesture recognition is usually useful for all types of individuals. This approach is computationally more efficient than finding the fingertip during a cluttered background and tracking it. additionally, to the above features, our mouse has well-defined regions like buttons to implement clicking. To click a button the user simply covers a button region a couple of times together with his or her finger or a pointer. it's a program that applies picture handling, recovers important information and carries out it to the mouse interface of the PC as indicated by predefined notions. The code consists on Python3.7. It utilizes the cross-stage picture handling module OpenCV and executes the mouse activities utilizing the Python-explicit library PyAuto GUI. Video catches by the webcam are handled and just the three hued fingertips are removed. Their focus is decided by utilizing the technique for minutes and counting on their relative positions it's concluded that what activity is to be performed. As we'll see that there are numerous new Technological advancements happening like biometric authentication which we'll see frequently in our smartphones, similarly hand gesture recognition may be a modern way of human-computer interaction i.e., we'll control our system by showing our hands before webcam and hand gesture recognition is usually useful for all types of individuals. This approach is computationally more efficient than finding the fingertip during a cluttered background and tracking it. additionally, to the above features, our mouse has well-defined regions like buttons to implement clicking. To click a button the user simply covers a button region a couple of times together with his or her finger or a pointer. it's a program that applies picture handling, recovers important information and carries out it to the mouse interface of the PC as indicated by predefined notions. The code consists on Python3.7. It utilizes the cross-stage picture handling module OpenCV and executes the mouse activities utilizing the Python-explicit library PyAuto GUI. Video catches by the webcam are handled and just the three hued fingertips are removed. Their focus is decided by utilizing the technique for minutes and counting on their relative positions it's concluded that what activity is to be performed.

2. RELATED WORK

We studied several reports and papers which gave us a few extra ideas about our project. [1] The first paper was based on the way to control the position of the cursor with the bare hands without using any electronic device. While some actions similar to clicking and dragging things are going to be completed with various hand gesticulations. The proposed system will only require a webcam as a data input device. The software's which will be required to implement the proposed system are OpenCV and python. The output of the camera is going to be displayed on the system's screen so that it is often further calibrated by the user. The python dependencies which will be used for implementing this technique are NumPy, math, wx and mouse. [2] The second report is based on the detailed explanation of the algorithms and methodologies for the colour detection and virtual mouse published in 2019 by Kollipara Sai Varun, I. Puneeth, T. Prem Jacob. In human-computer interactions, hand gesture recognition plays a vital role. As we will see that numerous new Technological advancements are happening like biometric identification which we will see frequently in our smartphones, similarly hand gesture recognition is a modern way of human-computer interaction i.e., we will control our system by showing our hands ahead of webcam and hand gesture recognition is often useful for all kinds of people. [3] The third report is based on a fingertip-gesture-based interface that allows humans to easily interact with computers by hand published in 2021 by Dinh-Son Tran & Ngoc-Huynh Ho & Hyung-Jeong Yang1 & Soo-Hyung Kim & Guee Sang Lee. A most challenging part for human-computer interactions is the real-time fingertip-gesture which is based on the interface which continues to be challenging for human-computer communications, acknowledgements to sensor noise, increasing light levels, and accordingly the complexity of tracing a fingertip beyond a spread of problems. Utilizing fingertip tracking being a virtual mouse could likewise obtain a well-liked technique of communicating with computers without a mouse device. In this work, the authors introduced a singular virtual mouse system using RGB-D pictures and fingertip apprehension. The hand area of the case and therefore the core of the palm is originally extorted using in-depth skeleton-joint knowledge pictures from version 2 of Microsoft Kinect Sensor then transformed within a binary image. Then, the outlines of the hands are extricated and characterised by a border-tracing algorithm. The algorithm is used to distinguish the fingertip spot, backed by the hand-contour coordinates is KCosine. Subsequently, the RGB pictures are managed by a mouse pointer using the screen to locate the fingertip. The system tracks fingertips in real-time at 30 FPS on a private computer employing one CPU and Kinect V2. The experimental results showed a high accuracy level; the system can work well in real-world environments with one CPU. [4] The fourth report is assigned with a computer vision-based mouse, which may command and control the cursor of a computer or a programmed system operating a camera promulgated in 2017 by Aykut ERDEM, Erkut ERDEM, Yasemin Yardimci, Volkan ATALAY, A. Enis ÇETİN. To relocate the cursor on the system screen, the user utterly relocates the mouse-shaped virtual device placed on a facade within the viewing range of the camera. The video delivered by the camera is measured using computer vision systems and accordingly, the system initiatives the cursor according to mouse movements. The computer vision-based mouse has regions like buttons for clicking. To click a button the user simply covers one among these regions with his/her finger. [5] This paper recommended an expeditious algorithm for automatically diagnosing a restricted collection of gesticulations from hand copies for a robot controller application published at 2014 by Asanterabi Malima, Erol Özgür, and Müjdat Çetin. Hand gesture recognition may be a challenging problem in its general form. We contemplate a difficult and quick collection of standard rules and a moderately structured background and acquire an easy, yet powerful, method for movement recognition. Our strategy includes measures for segmenting the hand area, establishing the fingers, and ultimately distributing the indication. The algorithm is variant to the scale, translation and rotation of the hand. We illustrate the effectiveness of the procedure on the real comparison.

3. PROPOSED MODEL

In this project, the utilization of the hand as a virtual mouse can do all that a mouse manages without contacting the system. The webcam of the framework had been used to recognize my hands. It will then, at that point, make a jumping box around hands and spotlight on two fingers: The front finger and the centre finger. The front finger will go about as a cursor and moving it around, we will be moving the cursor around. Presently, to effectively click utilizing hand following, it is recognizing the distance between the front finger and the centre finger. On the off chance that they are combined, then, at that point, it will play out a tick. There are four modules Introduction and data collection model: To read the relevant literature and check the feasibility of the project. Start working on the front end and implementation of codes to detect the efficiency and accuracy of the product in public point of view. Codes are available on GitHub. Modules that are should have been introduced for it to work appropriately are OpenCV, Mediapipe and Autopy.

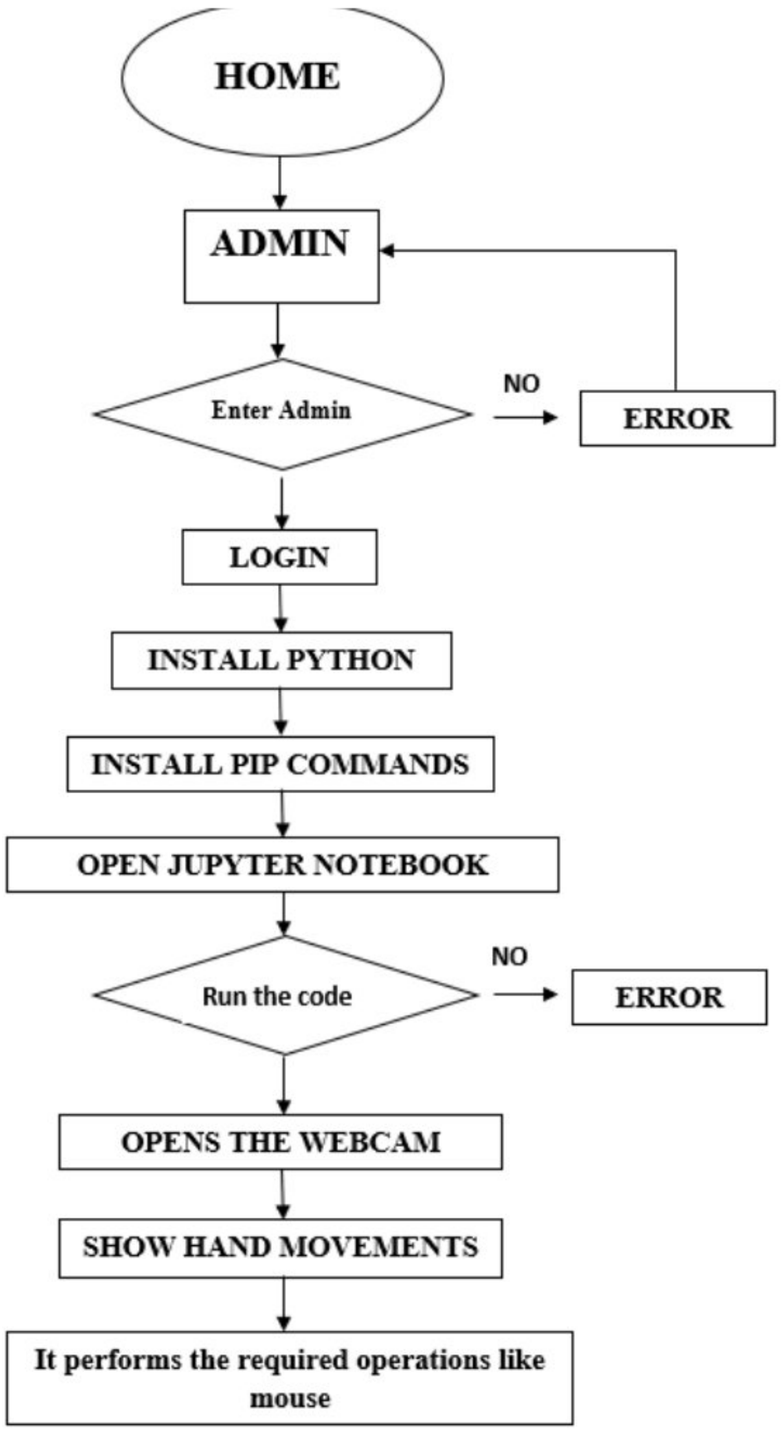

Fig 1: Architecture Diagram

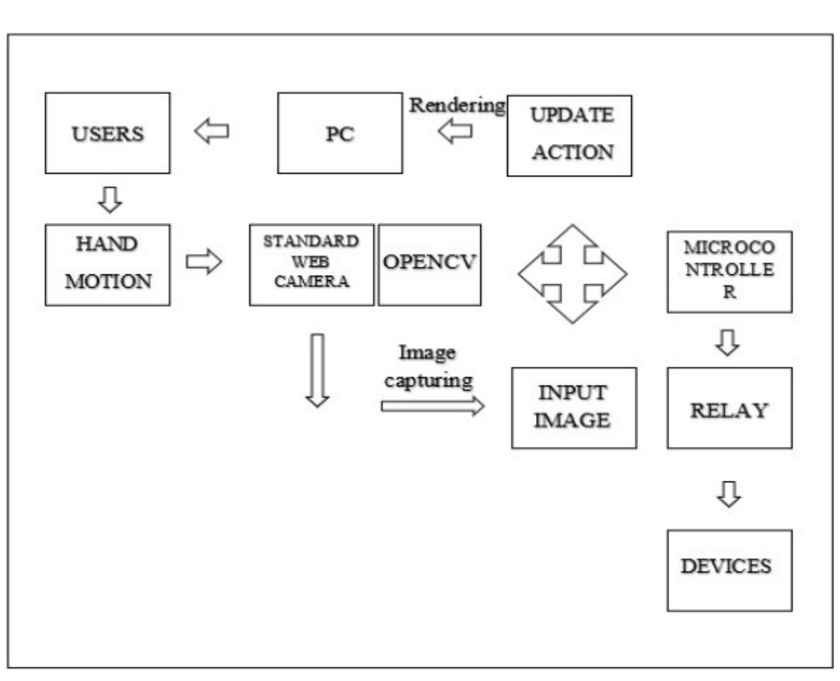

Fig 2: System Architecture

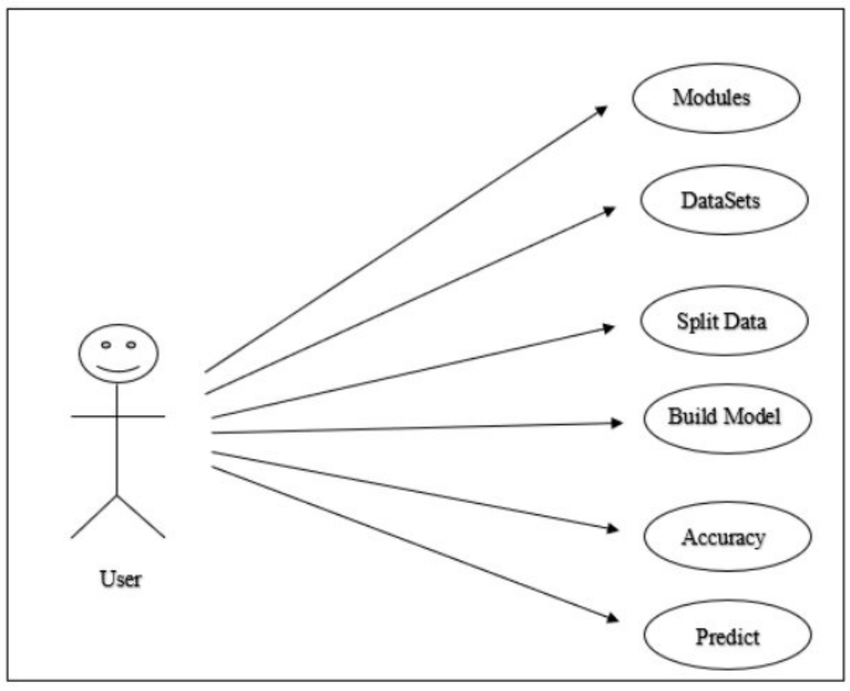

Fig 3: USE Case diagram

4. DATA COLLECTION

An input prompt will be raised, press 's' and hit enter to begin saving pictures for the foundation. After pressing 's', it will catch 200 pictures of the foundation. The presentation window will show up and begin catching the pictures, so escape the edge and permit the camera to catch the foundation. The program will naturally close. Presently you could check by perusing if the dataset is made or not. The picture dataset will be made in a similar registry where the python program is put away. Four catalogues will be made by the name designated to them. The folders have been made effectively, presently check to assume the pictures have been caught and saved. The picture size won't be as old as we're seeing during the catching system. We have diminished the picture size so when it is utilized in an AI venture to prepare the model it requires some investment.

5. ALGORITHM

Step 1: To start with, we need to bring every one of the necessary modules into the program console. We just need two modules, one is the "OpenCV" and the other is the "os" module. Opencv is utilized to catch and deliver the picture utilizing the PC camera and the os module is utilized to make an index.

Step 2: Make Camera Object: As we need to make our picture dataset, we really want the camera, and OpenCV assists us with making camera protests that can be utilized later for different activities.

Step 3: Make Label Folders: Now, we want to make organizers for each mark for separation. Utilize the underneath given code for making these envelopes, you could add however many marks as you need. We have given our mark names as per the game stone, paper, scissors. We are setting up a dataset that could characterize the picture as though it is a stone or paper or scissor or simply a foundation.

Step 4: The last advance to catch pictures: This is the last and most significant stage of the program. Inline remarks have been composed to make it clearer. Here we need to catch pictures and store those pictures as indicated by the

mark organizer. Perusing the code completely we have referenced every easily overlooked detail here.
Step 5: We are utilizing jupyter scratch pad to run this program, you could utilize any python mediator. To start with, go to the cell menu and snap on "Run All" this will run every one of the cells accessible in one stroke.

6. FEASIBILITY

As already demonstrated in the literature review, the work on virtual mouse had been already done using various modules and techniques. (Reference 1 to 5).
• Solving the current real-world issue, segments in problem creation, usage of the real basic objects will be done.
• The framework should be assessed according to the specialized perspective first. The evaluation of this possibility should be founded on a framework plan of the framework necessity in the terms of input, result, projects and strategies. Having recognized a layout framework, the examination should proceed to recommend the sort of hardware, required technique creating the framework, of running the framework whenever it has been planned.
• The developing arrangement needs to be maintained by value including profit. Guidelines to guarantee that purpose is focused toward the outline, which will perform greatest, yield at the most immediate. Unitedness of the circumstances, which influence the expansion of a new system, is the value it would expect. The following are a number of the important financial questions asked during the preliminary investigation: the prices conduct a full system investigation. The cost of the hardware and software. The benefits are in the form of reduced costs or fewer costly errors.
• Functional plausibility is a proportion of how well a proposed framework addresses the problems and makes the most of the chances distinguished during extension definition and how it satisfies the prerequisites recognized in the necessities examination period of the system development. Our proposed framework beats every one of the issues identified with the present complex system and fulfils all the extensions as characterized.
• The project would be valuable since it fulfils the targets when created and installed. All conduct viewpoints are thought about cautiously and infer that the venture is behaviorally attainable.

7. IMPLEMENTATION AND FUTURE ENHANCEMENT

Gathering erudition, monitoring devices (like a webcam) operating correctly or not, Acquisition strip or finger fabric, which should be appropriate to the fingers, Import packages like NumPy, OpenCV, pynput. mouse, tkinter, Implement the Open Gesture Operation and Fine Tuning.

8. CONCLUSION

This model can finish up by utilizing the subjects of computer vision like open CV can frame masks
that can variate tones by utilizing shading variation techniques and the advancement of mouse movement by utilizing specific bundles like 'mouse' which will be utilized for the development of mouse by using the directions that are connected to the detected color. This can give ease utilization of frameworks and many different applications. The advancement of these methods and models
are truly tremendous. The shading recognition model can be developed assuming we need to recognize a specific colour out of a hued photograph. Furthermore, the mouse development can be created in such a manner it can carry on like a genuine mouse that will help us for utilizing the framework without contacting the framework's console or mouse. The improvement can be in such a manner it tends to prepare on CNN's that will help for a superior performed model. The Models can be created in various ways by utilizing some most recent bundles like 'pyautoGUI' that will assist us with providing orders which will recognize include and fill a few roles on the framework. So, if any different shading is recognized it can fill the extraordinary role or on the other hand assuming contribution from the client is distinguished it will open a particular organizer easily without playing out any activities, a basic signal can do the work. So the open CV is helping the clients with various available types of models that will make ease life.

9. REFRENCES

[1] Guoli Wang, (2010). Optical Mouse SensorBased 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.