International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 05 | May - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Hand Gestures Recognition Based Virtual Mouse and Keyboard

Mosam Shambharkar[1], Sanket Chore[2], Prince Barsagade[3], Piyush Chavhan[4], Prof. D. R. Uike[5]
(Department of Information Technology, Government College of Engineering, Amravati - 444601, Maharashtra, India)

Abstract -

Nowadays computer vision has reached its pinnacle, where a computer can identify its owner using a simple program of image processing. In this era of development, people are using this vision in many aspects of daily life, like Face Recognition, Color detection, Autopilot car, etc. In this project, computer vision i.e., a camera is used in creating a virtual mouse and keyboard using hand gestures recognition. The camera of the computer will read the image of different gestures performed by a person's hand and according to the gestures the computer's mouse cursor will move and even perform the different functions using different gestures. Similarly, the keyboard functions may be used with some other gestures. It will act as a virtual mouse and keyboard with no wire or external devices. The only hardware aspect of the project is a web-cam and the coding is done on python using Anaconda platform. This project makes use of the state-of-art Machine Learning and Computer Vision algorithms to recognize hand gestures, which works smoothly without any additional hardware requirements.

Keywords- Keyboard, Machine Learning, MediaPipe, Mouse, OpenCV.

1. INTRODUCTION:

Virtual Mouse and Keyboard derived from Gestures of Hand makes human interaction with computer simple by only making use of Hand Gestures. The computer requires almost no direct contact. All of the input and output operations can be virtually controlled by using static as well as dynamic hand gestures. While using wireless devices such as a wireless mouse or keyboard, it requires a dongle to connect to the PC, and it also requires a battery to power the mouse as well as keyboard for operating. But in this paper, the user uses his/her built-in camera or a webcam and hand gestures to control the computer mouse and keyboard operations. It has been considered that Hand gestures recognition is an important development technology in industry 4.0 in Human-Computer-Interactions (HCI). They provide the computers the ability to capture the hand gestures and execute the assigned commands without even touching devices such as a mouse or a keyboard physically.

For the development of this application of the AI virtual mouse & keyboard system, Python programming language is used. In the proposed system, the model makes the use of the MediaPipe Framework developed by Google for the tracking of the hands and fingertip. It consists of two modules: 1. Virtual Mouse and 2. Virtual Keyboard. The virtual mouse is for controlling the mouse's functions such as scrolling, right or left click etc., and the virtual keyboard is for controlling the keyboard's functions which includes typing from the keyboard's keys.

1.1. Problem Description and Overview. The proposed Hand Gestures based virtual mouse and keyboard system can be used to overcome problems in the real world such as when there is no power backup for wireless mouse and keyboard for operating and situations where there is no space for a physical mouse or keyboard to use, it can also be used by the persons who do not have prior knowledge of controlling a mouse or keyboard. Also, amidst of the COVID-19 situation, it is not safe to use the devices which are used by public in offices or internet-cafes by touching them because it may result in a possible situation of spread of the virus by touching them, and a person can easily control his/her computer within a greater range wirelessly without any powered backup device, i.e., only using hands, so the proposed Hand Gestures based virtual mouse and keyboard system can be used to overcome these problems since hand gestures detection is used to control the computer's mouse and keyboard functions by using a webcam or a built-in camera which requires no physical touch so far.

1.2. Objective. The main objective of the proposed Hand Gestures based virtual mouse and keyboard system is to develop an alternative to the regular and traditional mouse and keyboard system to perform and control the mouse and keyboard functions, and this can be achieved with the help of a web camera that captures the hand gestures and then processes these frames to perform the particular mouse and keyboard functions.

2. RELATED WORK:

2.1. Virtual Mouse. There are some related works carried out on virtual mouse using hand gesture detection by wearing a glove in the hand and also using color tips in the hands for gesture recognition, but they are no more accurate in mouse functions. The recognition is not so accurate because of wearing gloves; also, the gloves are also not suited for some users, and in some cases, the recognition is not so accurate

because of the failure of detection of color tips. Some efforts have been made for camera-based detection of the hand gesture interface.

G. Sahu et al. built a system for controlling mouse pointer using webcam which control volume of media player, PowerPoint slides and can make or end a call. They used RGB color tapes to recognise user's finger.

In 2019, K. Hassan et al. presented a system to design and develop a hand gesture based virtual mouse. They captured different gestures via webcam and performed mouse functions according to the gestures. This system achieved 78% to 90% accuracy. The system does not work efficiently in the complex or rough background

2.2 Virtual Keyboard. V. Saraswasti et al. introduced a system for disabled people entitled *Eye Gaze System to Operate Virtual Keyboard*. First it captures the user's face and gets the position of eye gaze which is used as reference point in the later stages. HaarCascade method was used to extract features of face, and Integral Projection method was used to get the position of the eye movement. Based on their experiment, the ratio between the duration of normal writing and duration of typing using their system for two words is 1:13.

In 2018, Jagannathan MJ et al. presented a finger recognition and gesture based augmented keyboard system. The system was developed using OpenCV libraries and Python. Palm detection is used for typing on the augmented keyboard. Virtual Keyboard performs based on the movement of the finger.

As we can see from the reviewed literature and related work, previous systems includes either virtual keyboard or virtual mouse. Those systems can't fully eliminate the need of mouse and keyboard completely. This work aims to build an interactive computer system which can be operated without any physical mouse and keyboard.

3. METHODOLOGY:

For the purpose of detection of hand gestures and hand tracking, the MediaPipe framework is used, and OpenCV library is used for computer vision. The algorithm makes use of the machine learning concepts to track and recognize the hand gestures and hand fingertip. Other libraires used are PyAutoGUI and PynPut which are used for implementation of Mouse and Keyboard functions with hand gestures.

3.1 MediaPipe: MediaPipe is a framework which is used for applying in a machine learning pipeline, and it is an opensource framework of Google. The MediaPipe framework is useful for cross platform development since the framework is built using the time series data. The MediaPipe framework is multimodal, where this framework can be applied to various audios and videos.

3.1.1. MediaPipe Hands: MediaPipe Hands is a high-fidelity hand and finger tracking solution. It employs machine learning (ML) to infer 21 3D landmarks of a hand from just a single frame.

MEDIAPIPE HAND SOLUTION DETAILS:
Hand tracking neural network pipelines: *Lite and Full*, to predict 2D and 3D hand landmarks on an image / video sequence. Both pipelines consist of:

- Hand detector model, which locates hand region
- Hand tracking model, which predict 2D keypoints, 3D world keypoints, handedness on a cropped area around hand
- MediaPipe graph, with hand-tracking logic (Figure 1).

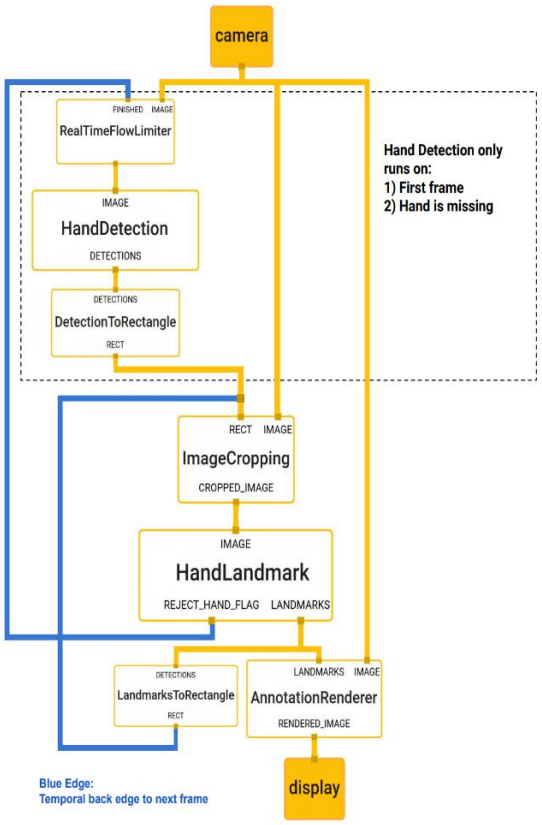


Figure 1: MediaPipe hand recognition graph.

The Hand models used are as follows:

- Palm detection model: TFLite model (lite), TFLite model (full), TF.js model

- Hand landmark model: TFLite model (lite), TFLite model (full), TF.js model

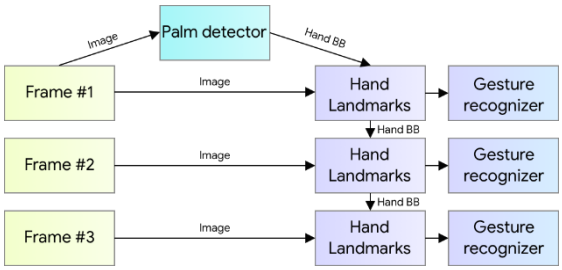


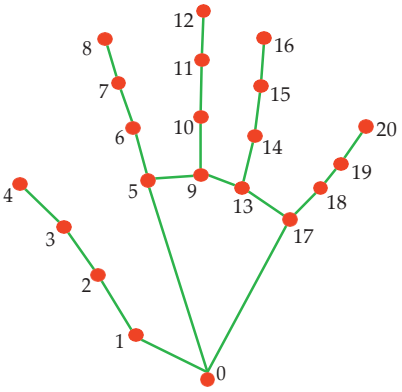
Figure 1 (b): Hand perception pipeline overview.

MediaPipe Hands utilizes an ML pipeline consisting of multiple models working together: *A palm detection model* that operates on the full image and returns an oriented hand bounding box as in Figure 1(b). *A hand landmark model* that operates on the cropped image region defined by the palm detector and returns high-fidelity 3D hand keypoints as in Figure 2.

SOLUTION SPECIFICATIONS:
Model Type - Convolutional Neural Network
Model Architecture - Two step neural network pipeline with single-shot detector and following regression model running on the cropped region.
Inputs - A video stream or an image of arbitrary size. Channels order: RGB with values in [0.0, 1.0].
Output(s) -
List of detected hands, each containing

- 21 3-dimensional screen landmarks
- A float scalar represents the handedness
- probability of the predicted hand.
- 21 3-dimensional metric scale world landmarks. Predictions are based on the GHUM hand model.

As show in Figure 3, Landmark screen z-value and 3D metric x, y, z coordinate values estimate is provided using synthetic data, obtained via the GHUM model (articulated 3D human shape model) fitted to 2D point projections.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

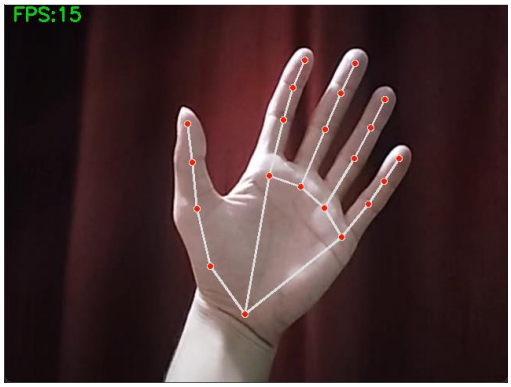


Figure 2: Co-ordinates or landmarks in the hand.

Furthermore, the non-maximum suppression works significantly better on small objects such as palms or fists. A model of hand landmark consists of locating 21 joint or knuckle co-ordinates in the hand region, as shown in Figure 2.

Landmark Key Point	x	y	z
0. WRIST	0.6840033531188965	1.0263876914978027	2.010356894288634e-07
1. THUMB_CMC	0.6130445003509521	0.9741635918617249	0.03094823844730854
2. THUMB_MCP	0.5557330846786499	0.8811166286468506	0.0455927699804306
3. THUMB_IP	0.5219229459762573	0.7894793748855591	0.0584392212331295
4. THUMB_TIP	0.5180284380912781	0.7064498662948608	-0.06933904439210892
5. INDEX_FINGER_MCP	0.5668647193908691	0.6334170508384705	-0.02117644414305687
6. INDEX_FINGER_PIP	0.5768647193908691	0.6534170508384705	-0.03117644414305687
7. INDEX_FINGER_DIP	0.5706912279129028	0.5962913036346436	0.04923054948449135
8. INDEX_FINGER_TIP	0.5715159177780151	0.5457372665405273	-0.06257902830839157
9. MIDDLE_FINGER_MCP	0.6479760408401489	0.7473073601722717	-0.013104413636028767
10. MIDDLE_FINGER_PIP	0.6382952332496643	0.6300567984580994	0.026784468442201614
11. MIDDLE_FINGER_DIP	0.6369984745979309	0.5659630298614502	-0.041496854275465 01
12. MIDDLE_FINGER_TIP	0.6405328512191772	0.5123766660690308	-0.053147152066230774
13. RING_FINGER_MCP	0.7000306844711304	0.7579972743988037	-0.02059961110353 47
14. RING_FINGER_PIP	0.6969378590583801	0.6438716650009155	-0.03746974468231201
15. RING_FINGER_DIP	0.6995339393615723	0.5830267071723938	-0.045679654926061 63
16. RING_FINGER_TIP	0.7026668190956116	0.5309850573539734	-0.05189122259616852
17. PINKY_MCP	0.7517656087875366	0.7871302962303162	-0.031399969011545 18
18. PINKY_PIP	0.7558741569519043	0.6990792751312256	-0.046594571322202 68
19. PINKY_DIP	0.7589550018310547	0.648797869682312	0.048333074897527695
20. PINKY_TIP	0.7612610459327698	0.6052000522613525	-0.049021389335393906

Figure 3: Key points of hand landmark corresponding in coordinate (x,y,z) in MediaPipe for one hand gesture.

3.2 OpenCV: OpenCV is a computer vision library which contains image-processing algorithms for object detection. OpenCV is a library of python programming language, and real-time computer vision applications can be developed by using the computer vision library. The OpenCV library is used in image and video processing and also analysis such as face detection and object detection.

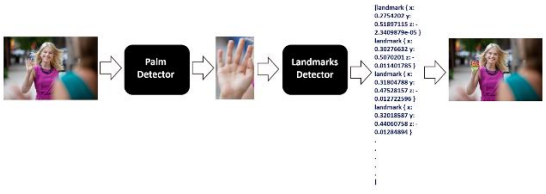


Figure 4: Palm and Landmarks detector and processing.

3.3 PyAutoGUI: PyAutoGUI lets your Python scripts control the mouse and keyboard to automate interactions with other applications. The API is designed to be simple. PyAutoGUI has several features: Moving the mouse and clicking in the windows of other applications. Sending keystrokes to applications (for example, to fill out forms). Take screenshots, and given an image (for example, of a button or checkbox), and find it on the screen. Locate an application's window, and move, resize, maximize, minimize, or close it (Windows-only, currently). Display alert and message boxes.

3.4. *PynPut:* This library allows you to control and monitor input devices. It contains sub packages for each type of input device supported. pynput.mouse Contains classes for controlling and monitoring a mouse or trackpad. pynput.keyboard Contains classes for controlling and monitoring the keyboard.

4. RESULTS:

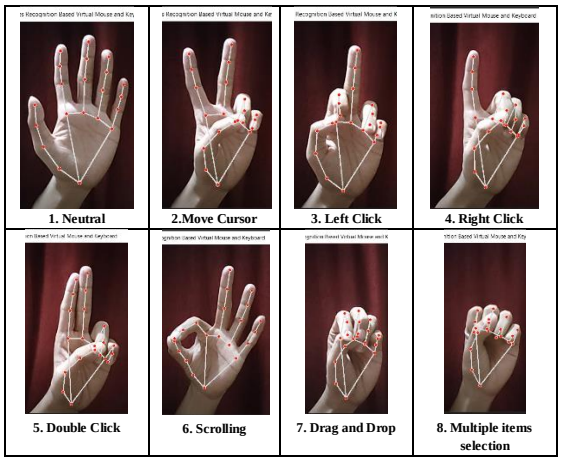


Figure 5: Hand Gestures for virtual Mouse Controller.

As you can see from Figure 5, the hand gestures for controlling Mouse consists of total eight gestures for controlling the cursor of the mouse and its functions such as left click, right click, double click, etc.

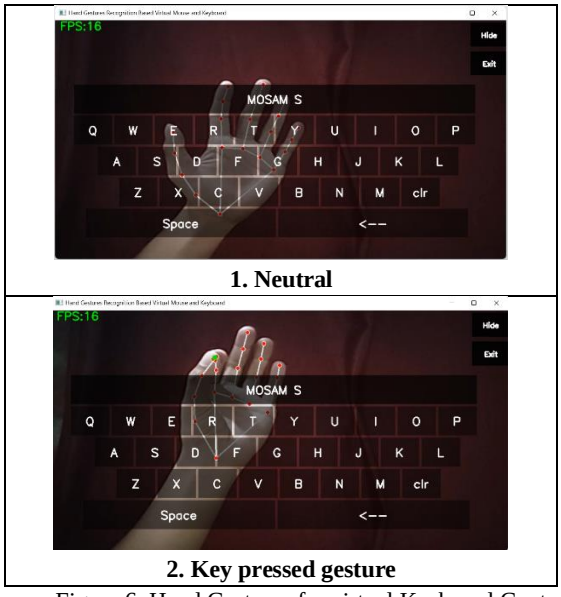


Figure 6: Hand Gestures for virtual Keyboard Controller.

And for the virtual keyboard, a QWERTY on-screen keyboard has been drawn at the window which gives output while the hand gesture of Key Pressed is hovered over the keyboard keys. The output can also be seen at the other applications where the cursor is placed. The green dot while the pinch gesture of the hand shows that the model is working correctly and based on the distance of the thumb and index finger, it is pressing the keyboard's key.

5. CONCLUSION:

The main objective of the Hand gestures based virtual mouse and keyboard system is to control the mouse cursor and keyboard functions by using the hand gestures instead of using a physical mouse or keyboard. The proposed system can be achieved by using a webcam or a built-in camera which detects the hand gestures and hand tip and processes these frames to perform the particular mouse and keyboard functions.

From the results of the model, we can come to a conclusion that the proposed Hand gestures virtual mouse and keyboard system has performed very well and has a greater accuracy compared to the existing models and also the model overcomes most of the limitations of the existing systems. Since the proposed model has greater accuracy, it can be used for real-world applications, and also, it can be used to reduce the spread of COVID-19, since the proposed mouse and keyboard system can be used virtually using hand gestures without using the traditional physical mouse or keyboard.

The model has some limitations such as small decrease in accuracy in scrolling mouse function and some difficulties in clicking and dragging to select the text and pinch gesture for keyboard keys selection, also not all the keyboard's keys are available in the proposed model. Hence, we will work next to overcome these limitations by improving the finger tip detection algorithm to produce more accurate results and more keyboard's keys implementation.

ACKNOWLEDGMENT:

Prof. D. R. Uike, Dept. of Information Technology (I.T.), Government College of Engineering, Amravati, has been a source of support and guidance to the authors throughout this research.

We are also obliged to our Head of Department (Dept. of Information Technology), Prof. A. W. Bhade for giving us this great opportunity. We are grateful for their cooperation during the period of our assignment.

REFERENCES:

[1] MediaPipe-Github, Access 2022.
[2] OpenCV for Image Processing.
[3] Grishchenko I, Bazarevsky V, MediaPipe Holositic-Simultaneous Face, Hand and Pose Prediction on Device, Google Research, USA, 2020, Access 2021.
[4] Applying Hand Gesture Recognition for User Guide Application Using MediaPipe Indriani et. al. 2021
[5] An Interactive Computer System with Gesture-Based Mouse and Keyboard, by Dipankar Gupta et. al. 2020
[6] Design and Development of Hand Gesture Based Virtual Mouse by Kabid Hassab Shibly et. al. 2019
[7] Finger recognition and gesture based augmented keyboard by Jagannathan MJ et. al. 2018.
[8] Sahu, G., Mittal, S.: Controlling mouse pointer using web cam (2016)
[9] Eye gaze system to operate virtual keyboard by Vidya Itsna Saraswati 2016
[10] A Study on Weareable Gesture Interface, Rakesh D. Desale et. al. 201