

Gesture Controlled Virtual Mouse with Voice Automation

Prithvi J, S Shree Lakshmi, Suraj Nair and Sohan R Kumar

Department of Computer Science And Engineering
B.M.S. College of Engineering Bengaluru, Karnataka, India

Ms. Sunayana S

Department of Computer Science and Engineering Visveswaraya Technological University,
Belgaum Bengaluru, Karnataka, India

Abstract— This research paper proposes a Gesture Controlled Virtual Mouse system that enables human-computer interaction using hand gestures and voice commands. The system requires no direct contact with the computer and allows for virtual control of all input/output operations. The system employs state-of-the-art Machine Learning and Computer Vision algorithms to recognize static and dynamic hand gestures and voice commands, without the need for additional hardware. The system comprises two modules, one that works directly on hands using MediaPipe Hand detection and another that uses gloves of any uniform color. The system leverages models such as Convolutional Neural Networks implemented by MediaPipe running on top of pybind11. The paper discusses the system's architecture, algorithmic approach to gesture recognition, and implementation of both modules in detail. The proposed system presents a natural and user-friendly alternative to traditional input methods and can have potential applications in healthcare and education. The paper's findings will be of interest to researchers and practitioners in the field of Human-Computer Interaction.

Index Terms— Gesture Control, Virtual Mouse, Human-Computer Interaction, Hand Gestures, Voice Commands, Machine Learning, Computer Vision, MediaPipe, Convolutional Neural Networks, Pybind11, Healthcare, Education.

I. INTRODUCTION

The field of Human-Computer Interaction has seen significant advancements with the introduction of innovative technologies. Traditional input methods such as keyboards, mice, and touchscreens have become more sophisticated, but still require direct contact with the computer, limiting the scope of interaction. Gesture-based interaction has emerged as an alternative approach to traditional methods, and the Gesture Controlled Virtual Mouse is an innovative technology that enables intuitive interaction between humans and computers. This research paper presents a comprehensive study of the Gesture Controlled Virtual Mouse, which leverages state-of-the-art Machine Learning and Computer Vision algorithms to enable users to control input/output operations using hand gestures and voice commands without the need for direct contact.

The Gesture Controlled Virtual Mouse is designed using the latest technology and is capable of recognizing both static and dynamic hand gestures in addition to voice commands, making the interaction more natural and user-

friendly. The system does not require any additional hardware, and the implementation of the system is based on models such as the Convolutional Neural Network (CNN) implemented by MediaPipe running on top of pybind11. The system comprises two modules, one of which operates directly on hands using MediaPipe hand detection, while the other module uses gloves of any uniform color. The system currently supports the Windows platform.

This research paper presents a detailed analysis of the Gesture Controlled Virtual Mouse, covering the system's architecture, algorithmic approach to gesture recognition, and implementation of both modules. The paper also discusses the advantages of the Gesture Controlled Virtual Mouse over traditional input methods, such as the increased naturalness and user-friendliness of the interaction. The findings presented in this paper will contribute to the growing field of Human-Computer Interaction and will be useful for researchers, developers, and anyone interested in the latest advances in gesture-based interaction technology.

II. PROBLEM STATEMENT

With the emergence of ubiquitous computing, traditional methods of user interaction involving the keyboard, mouse, and pen are no longer adequate. The limitations of these devices restrict the range of instructions that can be executed. Direct usage of hand gestures and voice commands have the potential to serve as input devices for more natural and intuitive interaction, enabling users to perform everyday tasks with ease. Such methods can offer a more extensive instruction set and eliminate the need for direct physical contact with the computer, further enhancing the user's experience.

III. LITERATURE SURVEY

A. Background

Gesture-based mouse control using computer vision has been a topic of interest for researchers for a long time. Various methods have been proposed for gesture recognition, but in this paper, the authors have proposed a new method based on color detection and masking. This system is implemented in Python programming language using the OpenCV library, which is a popular computer vision library. The proposed system is a virtual mouse that will work only based on webcam captured frames and tracking colored fingertips.

The objective of this paper is to develop and implement an alternative system to control a mouse cursor. The alternative method is hand gesture recognition using a webcam and a color detection method. The ultimate outcome of this paper is to develop a system that recognizes hand gestures and controls the mouse cursor using the color detection method of any computer.

The system works on the frames captured by the webcam on the computer machine or built-in camera on a laptop. By creating the video capture object, the system will capture video using the webcam in real-time. The camera should be positioned in a way so that it can see the user's hands in the right positions.

B. Literature Survey

In the previously proposed system by Kabid Hassan Shibly's "Design and Development of Hand Gesture Based Virtual Mouse" research paper published in ICASERT (2019), color detection is done by detecting color pixels of fingertips with color caps from the frames that were captured by the webcam. This is the initial and fundamental step of the proposed system. The outcome of this step will be a grayscale image, where the intensity of the pixels differs from the color cap to the rest of the frame, and the color cap area will be highlighted. Then, rectangle bounding boxes (masks) will be created around the color cap, and the color cap will be tracked. The gesture will be detected from the tracking of these color caps. At first, the center of two detected color objects is calculated, which is done by the coordinates of the center of the detected rectangle. To create a line between two coordinates, the built-in OpenCV function is used, and to detect the midpoint equation, a given formula is used. This midpoint is the tracker for the mouse pointer, and the mouse pointer will track this midpoint. In this system, the coordinates from camera captured frames resolution are converted to screen resolution. A predefined location for the mouse is set, so that when the mouse pointer reaches that position, the mouse started to work, and this may be called an open gesture. This allows the user to control the mouse pointer.

The previous system uses close gestures for clicking events. When the rectangle bounding boxes come closer to another rectangle, the bounding box is created with the edge of the tracking bounding boxes. When the newly created bounding box becomes 20 percent of its creation time size, the system performs the left button click, and it can be clicked. By holding this position more than 5 seconds, the user can perform a double-click. And for the right button click, again the open gesture is used. To perform the right button click, a single finger is good enough. The system will detect one fingertip color cap, then it performs a right button click. To scroll with this system, the user needs to use the open gesture movement with three fingers with color caps. If the users use their three fingers together and change its position to downwards, it will perform scrolling down. Similarly, if its position is changed to upwards, it will perform scrolling up. When three fingers move up or down, the color caps get a new position and new coordinates. By

the time all three color caps get new coordinates, it performs scrolls. If their y coordinate values decrease, it will perform scrolling down, and if the values increase, it will perform scrolling up. In conclusion, the proposed system has shown a new method for gesture-based mouse control using computer vision. The system uses color detection and masking to recognize hand gestures and control the mouse cursor.

IV. PROPOSED SYSTEM

A. Overview

The proposed Gesture Controlled Virtual Mouse system also includes a third module that leverages voice automation for wireless mouse assistance. This module allows users to perform mouse operations such as clicking, scrolling, and dragging, by simply giving voice commands. This feature is especially helpful for users who are unable to use hand gestures due to physical limitations.

The voice automation module is implemented using state-of-the-art speech recognition algorithms that enable the system to accurately recognize the user's voice commands. The module is designed to work seamlessly with the other two modules of the system, allowing users to switch between hand gestures and voice commands effortlessly.

This module also adds a layer of convenience by allowing users to perform mouse operations from a distance, without the need for any direct contact with the computer. This makes it a useful tool for presentations, demonstrations, and other scenarios where the user needs to interact with the computer without being physically close to it.

Overall, the Gesture Controlled Virtual Mouse system is an innovative and user-friendly solution that simplifies human-computer interaction. With its advanced machine learning and computer vision algorithms, it offers a reliable and efficient way for users to control their computers using hand gestures, voice commands, or a combination of both.

B. Convolutional Neural Networks (MediaPipe running on top of pybind11)

The convolutional neural network (CNN) implemented by MediaPipe is based on deep learning algorithms that use a series of convolutional layers to extract features from images. The basic algorithm for CNNs can be summarized as follows:

1. Input layer: Accepts the input image and performs preprocessing such as normalization.
2. Convolution layer: Applies convolution operation to the input image using multiple filters to extract relevant features. The output of this layer is called a feature map.
3. Activation function: Introduces non-linearity to the feature maps.
4. Pooling layer: Reduces the spatial dimensions of the feature maps to reduce computational complexity.
5. Repeat steps 2-4 for multiple layers.
6. Flatten layer: Converts the feature maps into a vector to feed them into the fully connected layer.
7. Fully connected layer: Performs the classification

task by applying weights and biases to the input vector.

8. Output layer: Produces the final output.

Here's a pseudocode implementation of a simple CNN algorithm:

Algorithm 1: Convolutional Neural Network Algorithm

```

Input: Input image  $I$ 
Output: Output feature map  $O$ 
1: Initialize: Set stride  $S$  and filter size  $K$ ; Calculate:
   Output size  $O_s = (I_s - K) / S + 1$ ;
2: for each filter  $F_i$  do
3:   for each output channel  $c$  do
4:     for each pixel in  $O_c$  do
5:       Calculate: Starting pixel  $ps = pixel_i * S$ ;
       Calculate: Ending pixel  $pe = ps + K$ ;
       Extract:  $K \times K$  region  $R$  from  $I_c$  starting
       from  $ps$ ; Convolve: Element-wise
       multiply  $R$  with  $F_i$ ; Sum: Add up all
       the elements in the resulting matrix;
       Assign: Result to corresponding pixel in
        $O_c$ ;
6:     end
7:   end
8: end
    
```

V. WORK DONE AND RESULTS ANALYSIS

A. Gesture-Controlled Mouse

- 1) *Neutral Gesture:* Neutral Gesture. Used to halt/stop execution of current gesture.
- 2) *Move Cursor:* Cursor is assigned to the midpoint of index and middle fingertips. This gesture moves the cursor to the desired location. Speed of the cursor movement is proportional to the speed of hand.
- 3) *Left Click:* Gesture for single left click

- 4) *Right Click:* Gesture for single right click
 - 5) *Double Click:* Gesture for double click
 - 6) *Scrolling:* Dynamic Gestures for horizontal and vertical scroll. The speed of scroll is proportional to the distance moved by pinch gesture from start point. Vertical and Horizontal scrolls are controlled by vertical and horizontal pinch movements respectively.
 - 7) *Drag and Drop:* Gesture for drag and drop functionality. Can be used to move/tranfer files from one directory to other.
 - 8) *Multiple Item Selection:* Gesture to select multiple items
 - 9) *Volume Control:* Dynamic Gestures for Volume control. The rate of increase/decrease of volume is proportional to the distance moved by pinch gesture from start point.
 - 10) *Brightness Control:* Dynamic Gestures for Brightness control. The rate of increase/decrease of brightness is proportional to the distance moved by pinch gesture from start point.
- B. Voice Automated Mouse
- 1) *Launch / Stop Gesture Recognition:* article graphicx Echo Launch Gesture Recognition Turns on webcam for hand gesture recognition. Echo Stop Gesture Recognition Turns off webcam and stops gesture recognition. (Termination of Gesture controller can also be done via pressing Enter key in webcam window)

- 2) *Google Search:* Echo search (text you wish to search) Opens a new tab on Chrome Browser if it is running, else opens a new window. Searches the given text on Google.
- 3) *Find a Location on Google Maps:* Echo Find a Location Will ask the user for the location to be searched. (Location you wish to find) Will find the required location on Google Maps in a new Chrome tab.
- 4) *File Navigation:* Echo list files / Echo list Will list the files and respective file numbers in your Current Directory (by default C:) Echo open (file number) Opens the file / directory corresponding to specified file number. Echo go back / Echo back Changes the Current Directory to Parent Directory and lists the files.

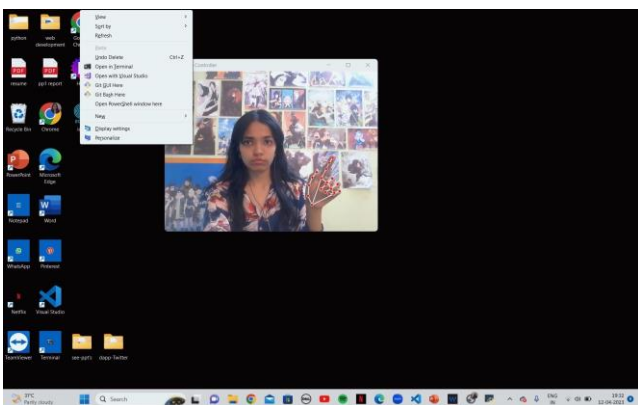


Fig. 1. Virtual Mouse



Fig. 2. Voice Assistant- ECHO

5) *Current Date and Time*: Echo what is today's date / Echo date Echo what is the time / Echo time Returns the current date and time.

6) *Copy and Paste*: Echo Copy Copies the selected text to clipboard. Echo Paste Pastes the copied text.

7) *Sleep / Wake up Echo*: Sleep Echo by Pauses voice command execution till the assistant is woken up. Wake up Echo wake up Resumes voice command execution.

8) *Exit*: Echo Exit Terminates the voice assistant thread. GUI window needs to be closed manually.

VI. CONCLUSIONS

In conclusion, Gesture Controlled Virtual Mouse is an innovative system that revolutionizes the way humans interact with computers. The use of hand gestures and voice commands provides a new level of convenience and ease to users, allowing them to control all I/O operations without any direct contact with the computer. The system utilizes state-of-the-art Machine Learning and Computer Vision algorithms such as CNN implemented by MediaPipe running on top of pybind11 to recognize hand gestures and voice commands accurately and efficiently. The two modules - one for direct hand detection and the other for gloves of any uniform color - cater to different user preferences and provide flexibility in usage. Additionally, the system incorporates a voice automation feature that serves various tasks with great efficiency, accuracy, and ease. With the current implementation of the system on the Windows platform, Gesture Controlled Virtual Mouse presents an exciting prospect for the future of human-computer interaction. It is expected to increase productivity and convenience for users and could potentially have numerous practical applications in industries such as healthcare, gaming, and manufacturing.

ACKNOWLEDGMENT

We would like to thank Miss Sunayana for her valuable comments, suggestions to improve the quality of the paper and for helping us review our work regularly. We would also like to thank the Department of Computer Science and Engineering, B.M.S. College of Engineering

for providing us with opportunity to encourage us to write this paper.

REFERENCES

- [1] Tsang, W.-W. M., Kong-Pang Pun. (2005). A finger-tracking virtual mouse realized in an embedded system. 2005 International Symposium on Intelligent Signal Processing and Communication Systems. doi:10.1109/ispacs.2005.1595526.
- [2] Tsai, T.-H., Huang, C.-C., Zhang, K.-L. (2015). Embedded virtual mouse system by using hand gesture recognition. 2015 IEEE International Conference on Consumer Electronics - Taiwan. doi:10.1109/icce-tw.2015.7216939 10.1109/icce-tw.2015.7216939.
- [3] Roh, M.-C., Huh, S.-J., Lee, S.-W. (2009). A Virtual Mouse interface based on Two-layered Bayesian Network. 2009 Workshop on Applications of Computer Vision (WACV). doi:10.1109/wacv.2009.5403082 10.1109/wacv.2009.5403082.
- [4] Li Wensheng, Deng Chunjian, Lv Yi. (2010). Implementation of virtual mouse based on machine vision. The 2010 International Conference on Apperceiving Computing and Intelligence Analysis Proceeding. doi:10.1109/icacia.2010.5709921 10.1109/icacia.2010.5709921.
- [5] Choi, O., Son, Y.-J., Lim, H., Ahn, S. C. (2018). Co-recognition of multiple fingertips for tabletop human-projector interaction. IEEE Transactions on Multimedia, 1-1. doi:10.1109/tmm.2018.2880608.
- [6] Jyothilakshmi P, Rekha, K. R., Nataraj, K. R. (2015). A framework for human-machine interaction using Depthmap and compactness. 2015 International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT). doi:10.1109/erect.2015.7499060.
- [7] [7]S. Vasanthagokul, K. Vijaya Guru Kamakshi, Gaurab Mudbhari, T. Chithrakumar, "Virtual Mouse to Enhance User Experience and Increase Accessibility", 2022 4th International Conference on Inventive Research in Computing Applications (ICIRCA), pp.1266-1271, 2022, doi:10.1109/ICIRCA54612.2022.9985625.
- [8] Shajideen, S. M. S., Preetha, V. H. (2018). Hand Gestures - Virtual Mouse for Human Computer Interaction. 2018 International Conference on Smart Systems and Inventive Technology (ICS-SIT). doi:10.1109/icssit.2018.8748401.
- [9] Henzen, A., Nohama, P. (2016). Adaptable virtual keyboard and mouse for people with special needs. 2016 Future Technologies Conference (FTC). doi:10.1109/ftc.2016.7821782.
- [10] Reddy, V. V., Dhyanchand, T., Krishna, G. V., Maheshwaram, S. (2020). Virtual Mouse Control Using Colored Fingerprint Tips and Hand Gesture Recognition. 2020 IEEE-HYDCON. doi:10.1109/hydccon48903.2020.9242677 .
- [11] Shetty, M., Daniel, C. A., Bhatkar, M. K., Lopes, O. P. (2020). Virtual Mouse Using Object Tracking. 2020 5th International Conference on Communication and Electronics Systems (IC-CES). doi:10.1109/icc48766.2020.9137854.
- [12] Xu, G., Wang, Y., Feng, X. (2009). A Robust Low Cost Virtual Mouse Based on Face Tracking. 2009 Chinese Conference on Pattern Recognition. doi:10.1109/ccpr.2009.5344072