Gesture Control Using Computer Vision

Kushagra Singh
Galgotias University
kushagra.8a@gmail.com

Anushka Singh
Galgotias University
anushkasingh13513@gmail.com

Ms. Lalita Verma
Galgotias University
lalita.verma@galgotiasuniversity.edu.in

Abstract: Humans for a long time have been communicating with devices like computers, smartphones, television, robots, drones, etc. Our technological developments have been increasing at a pace never seen before. New innovations are taking place on a regular basis. We have been switching our focus towards innovative and useful technology. One of the technologies which have been continuing to grow and is very useful and we can see its impact on our day-to-day life is wireless technology. We can clearly observe its impact as computers, phones, internet, headphones, microphones, even the chargers and many more technologies have benefited from wireless technology and made our life easier, but when it comes to interacting with these devices, we are still using traditional methods like keyboard and mouse. In this paper we aim to achieve a method for interacting with these devices using gestures with the help of computer vision. The program uses webcam to get a real-time video feed and then detects the hand landmarks to identify the gesture and completes the task associated with the gesture. Hence, we can interact with a device without the need to physically touch the device.

Keywords: Gesture, Computer Vision, OpenCV

I. INTRODUCTION

Technology has completely changed the day-to-day lives of people. Our way of interacting with the environment and each other has changed significantly. Machines and computers have been proven very helpful for our development. With the advancement in technology everything is being automated and machines and computers are becoming an essential part of our lives. Yet after all this development in technology we are using the same old traditional methods for communicating with machines and computers like keyboard and mouse.

Devices like keyboard and mouse decelerate the speed of communication with computers as we must provide very specific inputs for the desired outputs. With further advancements in technology the problem is being tackled. For example, with the advancements in wireless technology wireless devices like wireless mouse, trackpad, and keyboards are being developed and are already delivering on their promises but they are not the complete solution as they require separate charging, sometimes the communication and be disturbed due to some reasons etc. For actives that requires precise inputs such as gaming or programming devices like keyboard is still irreplaceable but for activities that doesn't require such precise input like navigation definitely needs some innovation in this field as it consumes most of our time. From simple activities like navigating in our pcs in our day to day live, or changing the channels on television, we are completely dependent on physically touching the buttons.

Gesture refers to the movement done by body parts specifically hand or forming certain patterns for example, waving or a thumbs up. Gestures are quite a normal way of communicating and are used by almost all of us in our day-to-day life. The same concept can also be applied to computers using computer Vision. We can use our webcam, which already comes equipped in most of the current devices to capture the video feed and identify simple gestures of our hand, accompanied by technologies like face recognition it can become more precise and avoid confusion.

In this paper we are going to implement this concept using Open CV and python, using our webcam to capture the feed and using open CV and python to recognize the hand landmarks and, successfully communicate with our device using gestures.

II. LITERATURE SURVEY

There are various other research going on in the field trying to come up with a solution for the above-mentioned problem. Neuralink is a company owned by Elon Musk which is working of brain implants that can give one ability to communicate to our computers just by thinking without

physically touching our machine, but this kind of activity sounds risky and quite inaccessible in current times.

Many companies are also working on voice assistants, which is yet another attempt to solve the above-mentioned problem. Many devices are already using this technology and getting some very good results.

Combining this technology with technologies like computer vision, which is already quite accessible as most of our devices come equipped with cameras, we can have quite a similar method to communicate with our devices.

III. OVERVIEW

The objective of this research is to implement this concept using Open CV and python, using our webcam to capture the feed and using open CV and python to recognize the hand landmarks and, successfully communicate with our device using gestures.

IV. ARCHITECTURE & EXPLANATION

1. Designing and Experimentation

1.1 Camera Setup

Our first objective is to receive feed from webcam. If we get a successful live video feed, we will further use the feed to identify different hand-landmarks as reflected in the image below. The device is able to recognize different hand-landmarks in the hand and also able to recognize whether it's a left hand or a right hand.

Figure 1: Hand-Landmarks

1.2 Hand Tracking Module

The ability to perceive the shape and motion of hands will play a vital role in detecting the hand and improve the overall user experience. We are going to use MediaPipe Hands which is a high-fidelity hand and finger tracking solution. MediaPipe uses machine learning to detect 21 landmarks of the hand from a single frame. We are going to create a hand tracking module using MediaPipe Hands to efficiently detect hands and fingers from each frame of the webcam feed. Following are the various steps involved in the making of hand tracking module:

- Get the Image: We will use the webcam feed to get the images from each frame of the video feed. If we successfully receive the images from the web cam feed, we will move on to the next step.
- Find the hand and its landmarks: The next step is to check if the hand is detected, if the hands are successfully detected then we will use it to get the landmarks of each hand. We are going to set the max hand limit to "2" for better performance. If the module successfully detects hands, it will return an array of different hands and its landmarks.

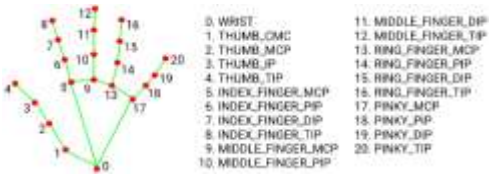

Figure 2: Hand Landmark Index

- Extract Data: We have created different functions to extract data from each frame of webcam. We are going to use the extracted data to detect gestures. The following functions are used to extract different data from each frame:
    - Find Hands: Firstly, we need to convert each image to RGB format as MediaPipe is compatible with RGB format only. Then we are going to return an array of hand landmarks.
    - Fingers Up: The second method that we need is to detect how many or which fingers are up, and which fingers are down. This function will return an array of 5 integers corresponding to each finger. For example, if index finger is out it will return [1,0,0,0,0].
    - Find Distance: This function is used to find distance between two points. It will help us to draw on screen.

We can program the device to detect certain kinds of gestures by setting conditions. For example, in the above image the as we can see all the five fingers are up so the program will return an array with each element corresponding to each finger. If the finger is up, it will return "1" or else return "0".

For the above image the following array would be returned: [1, 1, 1, 1, 1].

1.3 Defining different gestures

We are going to use computer vision to control presentation using gestures. We are going to define various gestures for different functionality. For example, we are going to use "thumb up" to go to previous slide. To achieve this, we are going to define a condition i.e., if the array returned equals to "[1, 0, 0, 0, 0]" then go to previous slide.

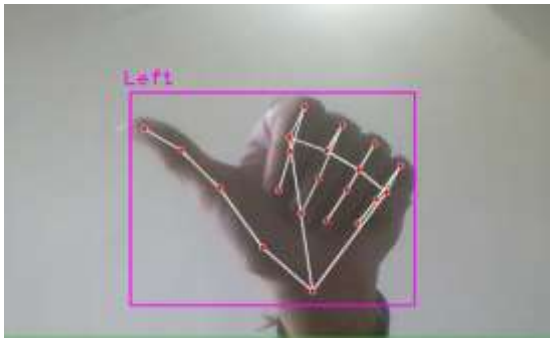

Figure 3: Previous Gesture

Similarly, we can use "pinky finger up" to go to the next slide by setting up similar conditions as above only difference we will target the pinky finger by using condition that the array returned is equal to "[0, 0, 0, 0, 1]".

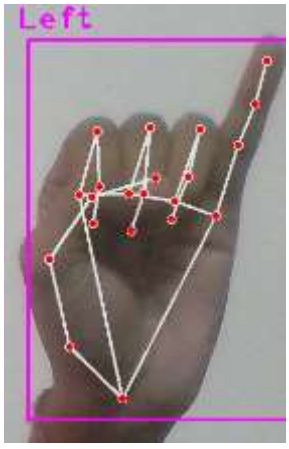

Figure 4: Next Gesture

We are going to set a threshold; the gestures will be fire action only if they are detected above the threshold to avoid confusion and avoid actions to fire amidst confusion. The green line as shown in the image below is the threshold.

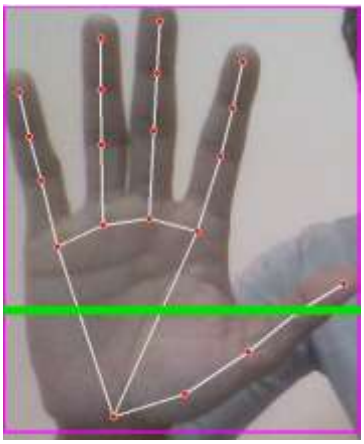

Figure 5: Threshold

Similarly, the third gesture that is used is two fingers up namely index and middle finger used to bring forth pointer. We can see the image below that a red pointer appears on the screen when the pointer gesture is detected, and this gesture follows the movement of the fingers.

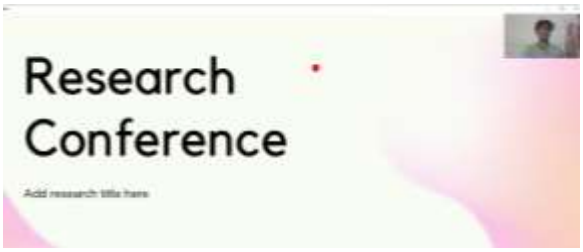

Figure 6: Pointer

Similarly, the next gesture is to draw on the screen using the pointer. We are going to use index finger up for this gesture. We will use the find distance function to extract the data between two points and use that to draw the line on the screen.

Figure 7: Draw

Finally, the final gesture is to undo or delete the line if anything goes wrong. We are going to use three fingersup for the following purpose.

V. Results and conclusions

I. Result

The above program performed exceptionally, and the proposed method can significantly help in communicating with the device with using gestures and without the need to physically press keys.

II. Discussion

- This concept can be implemented for many purposes including navigation, volume control, and other different controls such as playing and pausing a song or video from a distance.
- This concept is not only limited to laptops, but it can also be applied to any device that comes accompanied by a camera for example mobile devices or smart television etc.
- We can apply this for navigation through web or may even be implemented in web applications.
- This concept can also be used in augmented reality and to control objects in 3D without the need to press buttons giving a more realistic feeling.

III. Conclusion

After examining all these features, we came to this conclusion that this method would be able to solve the problem of communication with devices and can even be used in everyday activities with some proper research and experimentation.

IV. Future Scope

Since we are only relying on computer vision, in next versions voice assistant technology can be coupled with this technology to make the process more efficient.

Furthermore, technologies in the similar field can be proven useful to make the system more efficient. For example, model can be trained to recognize face of a presenter and only would react to the commands of the person. If the presenters are in a group, then the system can be programed to identify a gesture which allows to change the presenter. With all the above-mentioned technologies the above program can be turned into a more precise and efficient system and can provide a wonderful user experience.

V. Reference

1. MediaPipe, https://mediapipe.dev, 2020.
2. MediaPipe Hands, https://mediapipe.dev/hands, 2020.
3. MediaPipe Hands Model Card, https://mediapipe.page.link/handmc, 2020.
4. Google. Tensorflow.js Handpose. https://blog.tensorflow.org/2020/03/face-and-hand-tracking-in-browser-with-mediapipe-and-tensorflowjs.html.
5. Wikipedia https://en.wikipedia.org/wiki/Sign_language.
6. A. V. Dehankar, S. Jain, & V. M. Thakare (2017, December). “Performance analysis of RTEPI method for real time hand gesture recognition”. In 2017 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 1031-1036). IEEE.
7. Z. H. Chen, J. T. Kim, J. Liang, J. Zhang, & Y. B. Yuan (2014). “Real-time hand gesture recognition using finger segmentation”. The Scientific World Journal, 2014.
8. A. Choudhury, A. K. Talukdar, & K. K. Sarma (2014, February). “A novel hand segmentation method for multiple-hand gesture recognition system under complex background”. In Signal Processing and Integrated Networks (SPIN), 2014 International Conference on (pp. 136-140). IEEE.
9. ong yao,Xin Zang,Xiang Bai, Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Technique. IEEE.
10. Bobo Zeng, Guijin Wang, Xinggang Lin A Hand Gesture Based Interactive

Presentation System Utilizing Heterogeneous Cameras.

11. T.-D. Tan and Z.-M. Guo, “Research of hand positioning and gesture recognition based on binocular vision,” in Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI '11), pp. 311–315, March 2011.

12. Siddharth S Rautaray and Anupam Agrawal. Vision basedhand gesture recognition for human computer interaction: asurvey. Artificial intelligence review, 43(1):1–54, 2015.

13. P. Sharma, R. Joshi, R. A. Boby, S. Saha, and T. Matsumaru, “Projectable interactive surface using microsoft kinect v2: Recovering information from coarse data to detect touch,” in 2015 IEEE/SICE International Symposium on System Integration (SII). IEEE, 2015, pp. 795–800.

14. https://ijrpr.com/uploads/V2ISSUE5/IJRPR462.pdf Real-time vernacular sign language recognition using mediapipe and machine learning.

15. [PDF] Applying Hand Gesture Recognition for User Guide Application Using MediaPipe.

16. Vikram Sharma M, Virtual Talk for deaf, mute, blind and normal humans, Texas instruments India Educator’s conference, 2013.

17. Prakash B Gaikwad, Dr. V.K.Bairagi,” Hand Gesture Recognition for Dumb People using Indian Sign Language”, International Journal of Advanced Research in computer Science and Software Engineering, pp:193-194, 2014.

18. Obtaining hand gesture parameters using Image Processing by Alisha Pradhan andB.B.V.L. Deepak ,2015 International Conference on Smart Technology and Management (ICSTM).

19. Ge, Liuhao, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai and Junsong Yuan. “3D Hand Shape and Pose Estimation From a Single RGB Image.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 10825-10834.

20. Chengde Wan, Thomas Probst, Luc Van Gool, and Angela Yao. Self-supervised 3d hand pose estimation through training by fitting. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 10853–10862, 2019.