e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

TWO NAIVE ALGORITHMS FOR HAND TRACKING AND GESTURE RECOGNITION

Hrushikesh P*1, Sri Vasthav Reddy P*2, Shravalika P*3, Sushmitha S*4, Md Shabbeer*5

*1,2,3,4Student, Department Of Computer Science And Engineering, B V Raju Institute Of Technology, Medak, Telangana State, India.

*5Assistant Professor, Department Of Computer Science And Engineering, B V Raju Institute Of Technology, Medak, Telangana State, India.

ABSTRACT

Software Technology is growing more and more every day. The percentage of people who use any form of computers like smartphones, laptops etc. and internet is growing significantly each year. In this growing world of computer users, the use of a physical controllers like mouse, keyboard for human computer interaction may hinder natural interface as there is a strong barrier between the user and computer. Natural interaction between user and computer can be achieved using natural user interfaces like hand gesture controls and voice controls. Natural user interfaces can lead to increased efficiency of human computer interaction and also improves the usability of the computer. Natural UIs can also increase the satisfaction of the user, thus improving the overall user experience. Natural UI technologies require very minimal learning to seamlessly control the system when compared to hardware.

Keywords: Opencv, Artificial Intelligence, Computer Vision, Hand Gesture Recognition.

I. INTRODUCTION

Human–machine interaction (HMI) refers to the communication and interaction between a human and a machine via a user interface. The technology behind human machine interaction is growing rapidly every day. Natural user interfaces like gestures and voice controls have gained a lot of popularity among the users nowadays. The main reason for the rise in popularity of natural interfaces is because of the ease of interaction they provide with the machines.

The users do not need to use basic interfaces like mouse and keyboard, which may require learning about them, to effectively interact with the machine. Good natural user interfaces will certainly improve the user experience with using the machine and do not require as much effort as when using a keyboard for typing.

Natural user interfaces may not always be feasible, but can drastically improve the user experience in the areas where it is feasible. The technology which we use and interact with today are mostly embedded into devices like computers, mobile phones etc. Technology in 3-Dimensional space is being worked on more and more these days because of the massive potential it holds for improvements in human-machine interaction in the future. This 3D Technology will be fueled by natural user interfaces like hand gestures and voice controls.

II. METHODOLOGY

Natural user interfaces are already being used in mass these days. Of these, voice control is the most used technology. Hand Gesture Recognition for computer control has been in use since late 1990s. Hand gesture recognition for computer controls started with the invention of glove-based control interfaces, which contain electrical sensors in each fingertip that allow users to interact with virtual environments in an immersive and natural way.

Researchers realized that gestures inspired by sign language can be used to offer simple commands for a computer interface. This technology became more and more feasible and effective with development and use of infrared sensors, depth cameras, improved accelerometers.

Hand gesture recognition first used specialized gloves for computer controls, then depth cameras came into light for hand recognition and tracking them for gestures without any specialized hardware. Day-to-day cameras like computer cameras and webcams can also be used, but reduce hand recognition efficiency.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

III. MODELING AND ANALYSIS

FIRST ALGORITHM: USING OPENCV

The algorithm described in this paper is effective for how basic and beginner friendly it is.

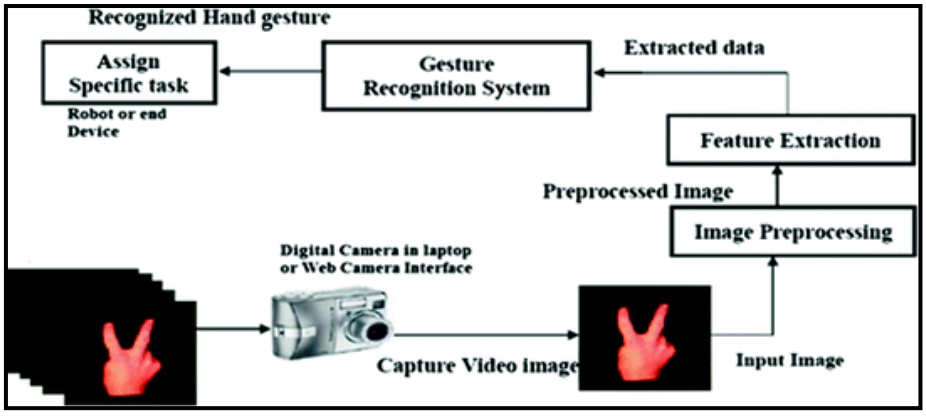

Figure 1: Hand Gesture Recognition Architecture

The steps involved in this algorithm are:

Step 1: **Capture background image and select ROI**:

Capture the image of the background in the frame of the camera without hand in the frame and select the Region of Interest (ROI) and average the background image for about 60 frames.

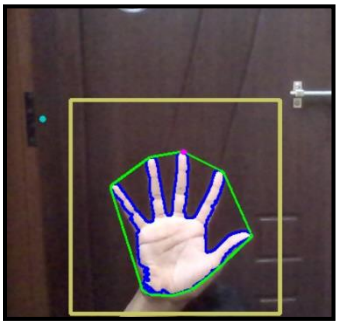

Figure 2: Image capture and ROI, Contours, Convex Hull

Step 2: **Image Preprocessing**:

Process the image captured with hand in the ROI by converting the image into greyscale for segmentation.

Step 3: **Hand Segmentation**:

Subtract the image with hand from the background without hand. This outputs the image with only hand in it represented in white.

Figure 3: Segment of the hand in the image

Step 4: **Finding Contours of the hand**:

Contours are defined as the line joining all the points along the boundary of an image that has the same intensity. Contours of the hand in the image is the line along the boundary of the hand after segmentation.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

Figure 4: Contours of the hand

Step 5: **Finding Convex hull for the contours**:

A convex hull is the smallest convex polygon that contains all points of a given set. Convexity means that a line segment between any two vertices of the polygon is completely inside the polygon.

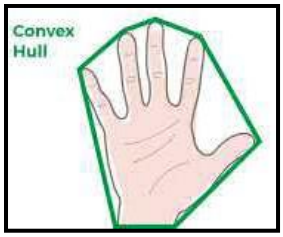

Figure 5: Convex Hull of hand

Step 6: **Construct circular ROI**:

A Circular ROI is a circle from the center of the convex hull and having radius 0.8 times the farthest point in the convex hull. This is used to identify the number of fingers up.

Figure 6: Circular ROI based on the center of the convex hull

Step 7: **Create a mask**:

A mask is a binary image consisting of zero and non-zero values. If a mask is applied to another binary or to a grayscale image of the same size, all pixels which are zero in the mask are set to zero in the output image. Create a mask of segmented image of the hand and the circular ROI image. This will show then output an image in which if the fingers are folded then there is only one segment i.e., of the wrist, else there are n+1 segments, where n is the number of fingers up.

Step 8: **Predict the Gesture**:

The gesture can be based on the number of fingers up in the image.

Figure 7: Mask of hand segment on circular ROI

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

IV. SECONG ALGORITHM: USING MEDIAPIPE WITH OPENCV

Mediapipe is a relatively new mobile-friendly AI framework which contains solutions for various Computer Vision related problems like Hand Recognition, Face Recognition, Iris Detection, Object Detection, Pose Detection and more. Mediapipe is developed by Google to "build world-class machine learning solutions" which can be deployed to almost any platform, from web assembly to Android to MacOS. Mediapipe is currently in alpha but it works almost better then already existing largely known frameworks for pose detection, hand recognition like OpenPose.

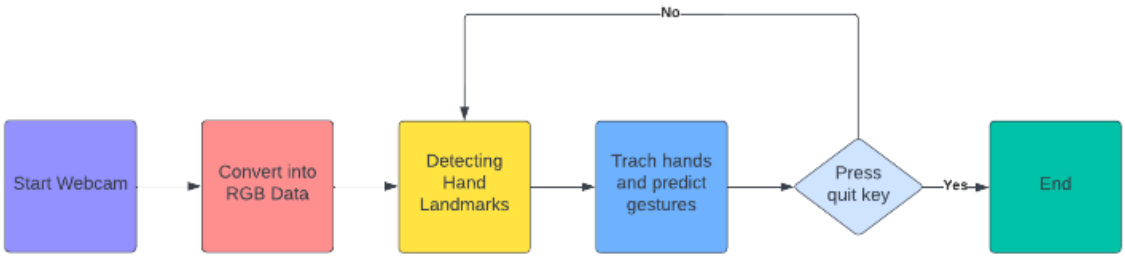

Figure 8: Architecture of the algorithm

The steps involved in this algorithm are:

Step 1: **Import necessary packages**:

Import Mediapipe framework and OpenCV for hand recognition and video capture as:

```
>import mediapie
>import cv2
```

Step 2: **Import the solutions for hand recognition**:

Import the classes hands, drawing_utils and drawing_styles from solutions in mediapipe as:

```
>Import mediapipe.solutions.hands
>Import mediapipe.solutions.drawing_utils
>Import mediapipe.solutions.drawing_styles
```

Step 3: **Start video capture**:

Start the video capture from opencv as:

```
>webcam = cv2.videoCapture(0)
```

Step 4: **Read the frames and convert them to RGB**:

The frames are grabbed from the video by using a "while" loop. These frames are initially in BGR color system. For hand recognition, the class "hands" requires an image with RGB system. We need to convert the frames grabbed into RGB as:

```
>success, frame = webcam.read()
>rgbframe=cv2.cvtColor(frame, opencv.COLOR_BGR2RGB)
```

Figure 9: Captured frame in RGB

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

Step 5: **Detect hand landmarks**:

Detect the hand landmarks in rgbframe using "hands" as:

>results = hands.process(rgbframe)

The class "results" contains the landmarks of hands detected in the frame as "multi_hand_landmarks" object. We can tweak the method "hands" in the class "hands" to increase/ decrease the model_complexity, min_detection_confidence, min_tracking_confidence and most importantly static_image_mode and max_num_hands with which we can limit the detection of no. of hands in the frame (default is 2).

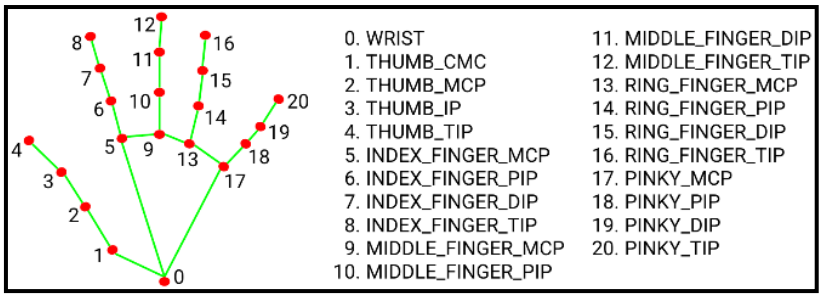

Figure 10: Hand landmarks representation

Step 6: **Draw the obtained hand landmarks on the frame**:

We draw the landmarks obtained from "multi_hand_landmarks" onto the frame, for visualization, using the function:

>mediapipe.solutions.drawing_utils.draw_landmarks()

and then show the frame using the opencv function:

>cv2.imshow("frame",frame)

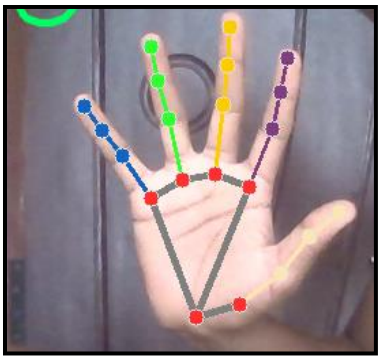

Figure 11: Hand landmarks drawn onto the frame

Step 7: **Predict gestures based on landmarks**:

Different gestures of the hand can be detected simply by the distance between the hand landmarks. As an example, we can detect the gesture "index finger up" based on simple math like the y-coordinate of landmark 7 is larger than the y-coordinate of landmark 8. The gesture "index and middle finger up" can be detected when both y-coordinates of 8,12 are less than 7,11 and the distance between x-coordinates of 8 and 12 is less than a threshold.

V. RESULTS AND DISCUSSION

FIRST ALGORITHM: USING OPENCV

This algorithm will track the hand and recognize simple gestures. The gestures that can be recognized with this algorithm are very simple, like the no. of fingers up in the frame. When the algorithm recognizes a gesture that is assigned to a particular computer function then the algorithm triggers that function. For example, let's say that the gesture "two fingers up" is assigned to the function left-click. When the algorithm recognizes the hand gesturing two fingers up, the function left-click is triggered.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

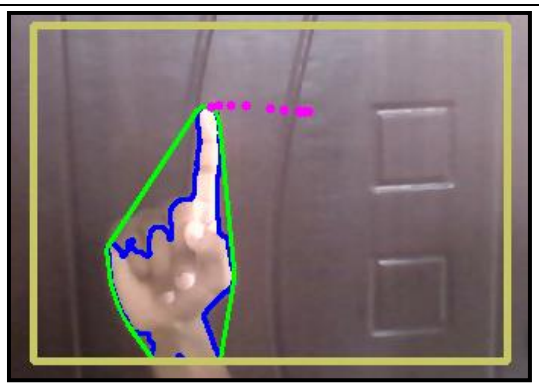

Figure 12: Algorithm tracking the index finger of the hand

In the figure above, the pink circles are the points where the index finger has been in the previous 20 frames (index finger points stored in a list of length 20).

The efficiency of hand tracking and hand gesture recognition by this algorithm is relatively low when compared to the second algorithm in this paper.

The following steps can be taken to improve the efficiency of this algorithm:

•Make sure that there is ample lighting on the hand for better recognition.

•The background should be significantly differentiable than the hand color for better recognition.

•Using bright colored tapes for different fingers which are differentiable from the background can massively increase the recognition of fingers and gestures.

•We can also use convexity defects instead of circular ROI mask to identify the number of fingers that are up, but this can be unstable if there is bad lighting in the image.

SECOND ALGORITHM: USING OPENCV AND MEDIAPIPE

The second algorithm discussed in this paper is far more reliable and efficient than the first one. The reliability of this design comes from the work put into training the model to detect hands and the landmarks by Google Inc.

This algorithm uses a deep learning model for image processing called as “single shot detector model” which reads the image, processes the image in a single shot and detects the hands in the image.

The identification of the locations of individual landmarks on the hand in the image can be really necessary for recognizing complex gestures like those in sign language.

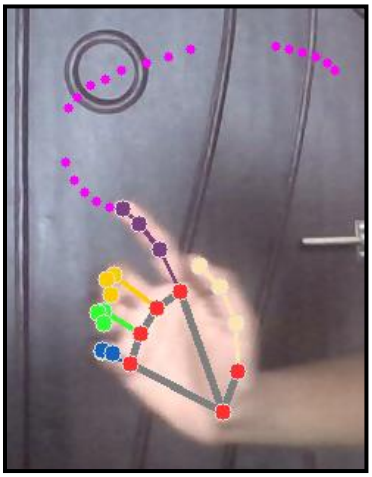

Figure 13: Algorithm tracking the index finger of the hand using Mediapipe

In the figure above, the pink circles are the points where the index finger has been in the previous 20 frames (index finger points stored in a list of length 20).

The Mediapipe algorithm detects the hand even in harsh lighting conditions and palm-colored backgrounds. This is because the algorithm searches for the hand rather than searching contours as in the first algorithm. The

results and discussion may be combined into a common section or obtainable separately. They may also be broken into subsets with short, revealing captions. An easy way to comply with the conference paper formatting requirements is to use this document as a template and simply type your text into it. This section should be typed in character size 10pt Times New Roman.

VI. CONCLUSION

This paper describes how simple algorithms can be used to effectively and efficiently recognize and track hands and recognize hand gestures which can then be used to control computers by assigning different simple commands to different gestures. This paper discusses the different stages an input images goes through to identify hand gestures. This paper also discusses how to improve the efficiency of gesture recognition using simple methods and when to use those methods.

Hand Gesture Recognition holds a lot of potential in fields like 3D technology like Metaverse, Virtual Reality etc. and Robotics for much effective and satisfying user experience. Both these technologies are extensively being worked on and are very near to being of use for public. Virtual Reality and Metaverse are almost out to the public and they hold a lot of potential for public entertainment and also productivity of work.

All these technologies are fueled by natural user interfaces i.e., hand gestures and voice controls, which make using the very fun and very efficient.

VII. REFERENCES

[1] J. Zeng, Y. Sun, and F. Wang, ―A natural hand gesture system for intelligent human-computer interaction and medical assistance, || in Proceedings of the 3rd Global Congress on Intelligent Systems (GCIS '12), pp. 382–385, November2012.

[2] P.Suganya, R.Sathya, K.Vijayalakshmi, International Journal of Pure and Applied Mathematics, 2018.

[3] Noreen, Uzma & Jamil, Mutiullah & Ahmad, Nazir. (2016). Hand Detection Using HSV Model. Advances in Computer Science and Engineering

[4] Sung, G., Sokal, K., Uboweja, E., Bazarevsky, V., Baccash, J., Bazavan, E.G., Chang, C.L. and Grundmann, M., 2021. On-device Real-time Hand Gesture Recognition. arXiv preprint arXiv:2111.00038.

[5] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C.L. and Grundmann, M., 2020. Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214.

[6] MediaPipe Hands. Retrieved April 22, 2022, from https://google.github.io/mediapipe/solutions/hands